QUT Digital Repository:
http://eprints.qut.edu.au/

This is the author version published as:

Wong, K.K. and Ho, T.K. (2004) ***Coast control for mass rapid transit railways with searching methods.*** IET Electric Power Applications, 151(3). pp. 365-376.

Copyright 2004 The Institution of Engineering and Technology

Coast Control for Mass Rapid Transit Railways with Searching Methods

K.K. Wong T.K. Ho

Abstract: With the daily commercial and social activities in the metropolitan cities, regulation of train service in Mass Rapid Transit Railways is necessary to maintain the service and fulfil the demand of passenger flow. Dwell-time adjustment at stations is one commonly used approach to regulate the train service but its control space is very limited. Coasting control is a viable means to meet the specific run-time in an inter-station run. The current practice is to start coasting at a fixed distance from the departed station, hence, it is only optimal with respect to a nominal operational condition of train schedule but not the current service demand. The advantage of coasting can only be fully taken when coasting points are determined in real-time. However, identifying the necessary starting point(s) for coasting under the constraints of current service conditions is no simple task because train movement is governed by a large number of factors. This paper studies the feasibility and performance of classical and heuristic searching measures to locate coasting point(s), with the aid of a single train simulator, according to specified inter-station run times.

List of symbols

X, Ψ, Ω sets on the solution space

V_c minimum coasting speed

V_{rm} re-motoring speed

F objective function

x_i optimum coasting point

d_i search direction

$\frac{z_1}{z_2}$ Golden ratio

α reflection factor

γ expansion factor

β contraction factor

1 Introduction

With the rising population and growth of economy in most developing, or even developed, countries, a reliable and efficient means of transportation is necessary to meet the expanding commercial, industrial and social activities. Mass Rapid Transit Railway systems (or metro) are currently one of the major means of mass transportation in most commercial and industrial cities around the world. Since the trains run on exclusive tracks, the service is usually free of congestion and hence a reliable and regular service has become an expectation in life. Indeed, any delays or interruptions on railway service may bring a city to a standstill, which may also carry a significant economic loss and substantially affect daily lives.

In order to maximise the capacity of the rail line and provide a reliable service for passengers throughout the day, regulation of train service to maintain steady service headway is essential. Service headway is often referred as one of the indicators for the quality of train service. Small service headway implies a high train frequency for passengers.

Adjustment to dwell times at stations is one of possible practices to maintain the headway regularity in metro systems because of its simplicity, regardless of other system constraints and parameters, such as traction equipment characteristics and signalling constraints. However, dwell times cannot be extended or shortened without limitation as the quality of train service may worsen as a result. From the viewpoint of passengers, a longer run-time is more preferable than a longer station waiting time if ever a trip has to be lengthened, particularly at off-peak hours. In addition, train-doors have to be kept open as long as a train is still at station. Energy loss from the air-conditioning, either in form of cooling or heating, through the open train-doors at stations accounts for substantial proportion of the electricity bill.

Coast control [1-2] is another possible approach to regulate headway by allowing the train to accomplish an inter-station run within a specific run-time through turning off the traction motors at a certain point (i.e. coasting point). Energy reduction comes as a bonus for electrified lines while the train spends less time on motoring. Another advantage of coasting control on the reduction of maintenance cost of braking equipment was exploited in another study [3]. Since the coasting point(s) is/are usually pre-determined prior to the daily train operation [4], it is only optimal with respect to a nominal operational condition of train schedule but not the current service demand which varies throughout the day. Thereby, the extent of time regulation is somewhat limited and the advantage of coasting can only be fully taken when coasting points are determined in real-time, taking into account the imminent needs of the train service. In other words, identifying the coasting point(s) to achieve a run-time as close to the specific one as possible with the minimum computational demand is essential to coast control.

A number of advanced techniques have been proposed recently to improve train operation in terms of service quality and energy consumption. A fuzzy logic control system [5] has been adopted to determine safe and accurate train speed control. Simulation results reveal that the number of changes of notches to regulate the train speed is reduced and hence the riding comfort is improved in comparison with PID control. A GA-based method [6] was proposed to synthesise a coasting lookup table in an inter-station run. The lookup table provides the locations where coasting, motoring or braking should commence. This study is primarily to minimise energy consumption in an inter-station run. With this coast control, a better train operation can be achieved when comparing with the conventional fixed-point coast control. The result also shows that the energy consumption obtained by coast control is even better than that achieved by fuzzy control. However, the suitability for real-time control and hence robustness to changing service demand were not discussed in details.

Train movement is governed by a large number of factors, such as track geometry, signalling, traction equipment characteristics, power supply and speed restrictions [7-9]. Some of them are position-dependent whilst the others are speed-dependent. As the coasting control is to alter the speed profile of the train at a particular position, formulation of an analytical model to connect the coasting points and their corresponding run-time and energy consumption and then applying appropriate optimisation techniques is very much impractical, if not entirely impossible. Further consideration of uncertainties, like human behaviour and equipment delay, only makes matters more complicated. Having ruled out an analytical approach, searching methods are the potential candidates to attain the optimal coasting points according to the real-time operational conditions. Despite numerous successful applications of heuristic methods, classical search methods are not to be neglected without extensive investigation in this study.

This paper presents the applications of both direct and heuristic search methods on locating single, or multiple if necessary, coasting point for an inter-station run with specified run-time and thus the comparisons of their performance. Such coasting control allows individual train control by requesting each train to meet its inter-station run-times while the run-times are derived from a central controller. The objective of this study is to explore possible ways to trade-off between computation time and the quality of the searched outcomes for the purpose of real-time control. Further, this study also discusses the appropriate algorithms in accordance with the track layout characteristics and inter-station distances, as well as their respective limitations. As the coast control is applied to each train independently, a single train simulator is a sufficient tool for the evaluation of the searching methods.

2 Coast control

Train movement in an inter-station run includes three operation modes, motoring, coasting and braking. Run-time and energy consumption required for this simple run depends on the proportion of these modes of operation. Fig. 1 illustrates a typical inter-station run and it consists of 4 phases: (1) acceleration from a complete stop to maximum permissible speed; (2) maintaining the speed as close to the maximum permissible speed as possible; (3) coasting, where there is slight deceleration due to frictional resistance to the motion; and (4) deceleration to a complete stop by the application of brake. Phases (1) and (4) are essential and hence compulsory in an inter-station run. As well as depending on the train's traction equipment characteristics, phases (1) and (4) are constrained by available adhesion and passenger comfort. In a short inter-station run, phase (2) may not exist and coasting starts once the train attains a certain speed. On the other hand, coasting may bring the train speed down to such an extent that re-motoring is necessary to take the train to the next station (i.e. repeated phases (1) and (3)). The location where coasting commences may thus drastically change the speed profile of the inter-station run, leading to wide range of possible run-times and energy consumption.

Fig. 2 shows the speed profile of a flat-out run between two stations, as well as those with three possible coasting point assignments. It is evident that different coasting points alter the speed profile significantly. One of the speed profiles even indicates re-motoring. As the motoring time is shortened because of coasting, energy saving is possible at the expense of run-time. Fig. 3 illustrates the possible time differences for three coasting points. Generally, the run-time can be extended when coasting is allowed to start sooner. The range of run-time resulting from coast control provides headway regulation and possible energy reduction for the operators.

Depending on the traction drive system, an energy saving of 30% can be attained with only a 5% increase in run-time [10].

2.1 Solution space

Theoretically, any point between the two stations is a possible solution for coasting. Fig. 4 shows that the solution set X may contain all points between A and D. Even if a certain distance resolution is imposed so that X is a finite set, the solution space is still large for any searching method. A number of subsets of X can be excluded from the space to make the searching process more feasible.

If a train starts coasting at a low speed, it is very likely that it will re-motor before it reaches the next station. The unnecessary turn-on/off of the traction drives is not desirable in practice because it will hasten the tear-and-wear of both the electrical and mechanical components. Re-motoring should thus be kept to minimal as much as possible. In addition, energy reduction is not guaranteed with too much re-motoring. To ensure the train has sustained sufficient momentum prior to coasting, a train is allowed to start coasting only when it reaches a minimum coasting speed V_c. In other words, coasting is prohibited from points A and B, as indicated in Fig. 4, and the set of the possible points between A and B, denoted as Ψ, can be excluded from X. The size of Ψ is thus determined by V_c, which is a pre-defined system parameter. When the inter-station distance is so short that only one coasting point is needed, run-time decreases and energy consumption increases monotonically if the coasting point shifts from the starting station to the next. The necessary coasting point to trade off run-time and energy consumption can be attained by simple optimisation techniques, except for some extreme track-related conditions, such as drastic changes of speed restrictions or steep slopes along the line, because there are no local optima obscuring the global one.

On the other end of the speed profile, after a train has entered the braking region, it cannot start coasting or it will overshoot the station and miss the stopping mark completely unless excessive braking is applied, which may over-strain the braking system and, more importantly, cause passenger discomfort. Therefore, at the approach of a station, there is a distance, between points C and D, on which the set of points Ω can also be discounted from X. However, point C is not a fixed parameter and it largely depends on how the train trajectory has gone before the train starts braking, in which the coasting point plays a decisive part. During the search, point C may be defined as a specific point on the braking profile for simplicity. Alternatively, it can be made a variable, estimated by the most updated location of the coasting point in the search.

With a longer inter-station distance, multiple coasting points may be required, which inevitably turns the solution space multi-dimensional. However, there are no specific rules on the number of coasting points. From Fig. 4, V_{rm} is an operation parameter for multiple coasting point identification. When the train speed falls below V_{rm} from coasting, it is allowed to re-motor to ensure sufficient momentum to go on. It should be noted the train spends more time in motoring mode and hence consumes more energy when multiple coasting points are allowed. V_{rm} is not used to eliminate the solution space in the searching process and it just ensures the next possible coasting point is located. Moreover, the location of the first coasting point inevitably affects that of the second and so on. Further, the solution space for the next coasting point varies with the location of the previous coasting point.

2.2 Objective function

In order to evaluate the possible solution, an objective function is necessary. In this study, the objective function is to determine how close the chosen coasting point is to lead to the desired run-time and it is quantified in the equation (1). F is a non-negative quantity and a smaller value implies a fitter solution. Other definitions for F are equally valid if other consideration is taken into account.

$$F = \left| \frac{T_g - T_D}{T_D} \right| \qquad (1)$$

where T_D is the desired run-time (sec) and T_g is the run-time achieved by the updated solution (sec). Since the run time may be either above or below the desired values in a particular run, the absolute sign is in place to nullify the polarity effect.

3 Searching methods

To locate the necessary coasting point for real-time control under a specified run time constraint, classical searching methods and heuristic approaches are adopted and the searching problem can be divided into single and multiple dimensional searches. In this section, three classical methods and a heuristic one are introduced as the tools to identify the coasting point for train service regulation. In general, there are two major approaches, direct and indirect searches.

With the direct search or numerical methods, the coasting solution is obtained in a step-wise manner, and the value of the cost function *F* is improved at each step. The direct search methods do not require an explicit evaluation of any partial derivatives of the function, but instead rely on values of the cost function *F*, gained from previous iterations. These methods basically use the cost function values to obtain numerical approximations to the derivatives of the cost function. Dichotomous search, Golden section search, Fibonacci search and Simplex method are examples of direct search.

Indirect or analytic search method, on the other hand, attempts to reach the necessary coasting point by calculation, without test or guess. It is based on the analysis of the special properties of the cost function F at the position of the extremum. In the simplest case, the tangent plane at the optimum is horizontal, that is the first partial derivatives of the cost function exists [11] and it can be defined as follows:

$$\left.\frac{\partial F}{\partial x}\right|_{x=x_i} = 0$$

where x_i is the optimum coasting point under the specified operational constraints on coasting control for service regulation. Gradient method is one of the examples of the indirect search, in which it selects the search direction d_i using the polarity of the value of partial derivatives of the cost function F with respect to the independent variables x, and the information gained from previous iterations.

Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS) and Simulated Evolution (SE) are well-known heuristic approaches [12]. They solve problems by trial-and-error with certain rules-of-thumb or guidelines and they often have an intuitive justification. One of the main differences between classical and heuristic methods is that the mathematical model is essential in the former, whilst it is not necessary for the latter. Moreover, classical methods always provide the best solution but heuristic approaches obtain a good, rather than the best, solution satisfying the defined constraints. Nonetheless, the common feature of these two approaches is the iterative nature.

3.1 One-dimensional search

Since the distance between stations is rather short in metro system (i.e. the solution space is small), a single coasting point is usually adequate for service regulation. In general, the run time and energy demand monotonically decreases and increases respectively when the coasting point move away from the starting station. Hence, this single-variable problem can be simply accomplished by classical optimisation methods. With one-dimensional search on coasting control, the Bi-section (Golden and Fibonacci) and Gradient based searching methods are deemed to be appropriate. With the Gradient-based methods, the cost function F has one variable (i.e. coasting point) only and the inter-station run time is a function of the coasting point. However, the continuity and hence differentiability of the function cannot be guaranteed. A point-by-point evaluation of F on the solution space is thus required. Besides, the applicability of Bi-section method is limited to functions which are unimodal (i.e. the function has only one global optimum point).

3.1.1 Golden section search

With the application of this algorithm on coasting control, the fitness of two initial coasting points are first determined and then used for further search of new coasting point. These two coasting points are obtained from either end on the solution space with the spacing of a golden ratio. The basic idea of the Golden section method [13-14] is that the solution space is divided into two unequal parts, whereas the ratio of the larger of the two segments to the total length of the interval should be the same as the ratio of the smaller to the larger segment.

Assume the solution space consists of a length z which composes of two segments z_1 and z_2, as shown in Fig. 5, the Golden section requires that

$$\frac{z_1}{z} = \frac{z_2}{z_1} \tag{2}$$

$$z = z_1 + z_2 \tag{3}$$

Equation (2) gives

$$z_1^{\ 2} = z \times z_2 \tag{4}$$

By substituting for z from equation (3) into (4) and normalizing with z_2^2, the following equation is obtained.

$$\left(\frac{z_1}{z_2}\right)^2 + \left(\frac{z_1}{z_2}\right) = 1 \tag{5}$$

This quadratic equation can be solved for the ratio z_1/z_2. The positive root is

$$\left(\frac{z_1}{z_2}\right) = 0.618033989 \tag{6}$$

Now assume the fitter coasting point is located between points *a* and *b*, the searching process with the Golden section method is listed as follows: -

1. Two initial coasting points, *x* and *y*, are placed with the "golden ratio" spacing (i.e. 0.618) from either end on the solution space between *a* and *b*, as shown in Fig. 5, the solution space *z* will then be reduced to a fraction of 0.618.
2. Assume *F*(*x*) is smaller than *F*(*y*), *y* replaces *b* and the new solution space z_1 becomes (*a*, *y*).
3. The process is repeated and the new solution space z_1 is further reduced by the golden ratio until the obtained coasting point satisfies the expected run time requirement of train operation.

To achieve a smaller solution space, the last estimate should be placed symmetrically at the middle of the solution space. The final guess of coasting point then provides an additional reduction on the solution space.

3.1.2 Fibonacci search

The concept of Fibonacci search [14-15] is very similar to that of the Golden section search. The main difference is that the reduction ratio on the solution space at each iteration is predetermined and optimised according to the given number of iterations with Fibonacci search. The arrangement of the search point (i.e. coasting point) within the new search interval is shown in Fig. 6.

To help illustrate how the reduction ratio at each iteration is devised, the iterations are shown in reverse order whereas; x_n and x_{n-1} are the last pair of estimates. Point "y" is one end of the solution space and the successive new coasting points attained (i.e. x_{n-2}, x_{n-1} and x_n) at iterations are assumed to be a fitter solution for the sake of simplicity. Hence, the new solution space becomes (x_{n-3}, y), (x_{n-2}, y) and (x_{n-1}, y) through the iterations. The searching process repeats until the obtained coasting point satisfies the expected run time requirement of train operation.

From Fig. 6, the interval of uncertainty L_{n-1} is (x_{n-2}, y) and the final search interval is defined as,

$$L_n = \frac{L_{n-1} + \varepsilon}{2} \tag{7}$$

L_n is the length of the interval of uncertainty after the nth iteration and ε represents the smallest distance by which two evaluations may be separated and still be distinguished from one another. The symmetry requirement for the search interval is

$$L_{n-2} = L_{n-1} + L_n \tag{8}$$

Combining equations (7) and (8),

$$L_{n-2} = 3L_n - \varepsilon \tag{9}$$

It is possible to work backward to determine the required size for any intermediate interval of uncertainty.

$$L_{n-3} = 5L_n - 2\varepsilon$$

$$L_{n-4} = 8L_n - 3\varepsilon$$

It can then be generalised as

$$L_{n-k} = F_{k+1}L_n - F_{k-1}\varepsilon \tag{10}$$

The coefficients F_{k+1} and F_{k-1} can be obtained by

$$F_{k+1} = F_k + F_{k-1} \qquad k = 1,2,3,\ldots n \tag{11}$$

and $F_0 = F_1 = 1$

Fibonacci method provides a specific reduction ratio on the solution space at each iteration, and the interval of uncertainty can be used to plan the evaluation spacing if the maximum number of iterations is determined in advance. In general, Fibonacci search method retains one of the two ends of interval from the previous iterations and therefore requires only one new estimate of coasting point in a new iteration.

3.1.3 Gradient based search

With the Gradient-based method [16], a certain derivative must be adopted to enable the search. In this application, the derivative should relate the control variable (i.e. coasting point) to the consequence (i.e. run time). Hence, the gradient required is given in equation (12). The two initial coasting points are chosen randomly and the gradient is obtained with them. The search direction of the updated coasting point depends on the polarity and the magnitude of the gradient,

$$Gradient = \frac{\Delta\ Run\ Time}{\Delta\ Coasting\ location} \tag{12}$$

The length of the next step can then be calculated by,

$$Step\ length = \left\{Gradient^{-1} \times \left[Run\ time_{Flat-out} - Run\ time_{Expected}\right]\right\} \tag{13}$$

The new coasting point can be obtained by the following equation

$$New\ coasting\ location = Old\ coasting\ location + Step\ length \tag{14}$$

In general, the step size varies with the difference between the run time attained by the latest iteration and the expected one. Therefore, the Gradient method is likely to require a smaller number of iterations to achieve the same level of convergence when compared with other methods. However, the drawback of this algorithm is that the step length and search direction cannot be defined in the searching process when there is no change on the run-time attained from two successive iterations (i.e. the gradient is zero) and the searching process will be terminated. It can be shown in Fig. 7 that the train is forced to operate in coasting between points *a* and *b* even in a flat-out run because train speed is confined by the lower speed restriction. Even if the coasting point is chosen at any location between points *a* and *b*, the speed profile of train behind point *b* is likely to be the same and so is the run time,

3.2 Multi-dimensional search

3.2.1 Nelder and Mead's method

It is an extension of the simplex method for the purpose of multi-dimensional search. A set of (n+1) mutually equidistant points in n-dimensional space is known as a regular simplex. Thus, in two-dimensional problem, the simplex is an equilateral triangle and in a 3-dimensional space, it is a regular simplex tetrahedron. The idea is to compare the return value of the fitness function at the (n+1) vertices of the simplex and move the simplex towards the optimum point during the iterative process. The vertices of the simplex represent the multiple coasting points on coasting control and each vertex is in two-dimensional form (i.e. a pair of coasting point) in this application for the sake of simplicity. The original simplex method maintains a regular simplex at each stage. Nelder and Mead [17] proposed several modifications to the method in which it allows the simplices to become non-regular. The result is a very robust and direct search method and it is extremely powerful provided that the number of variables does not exceed 5 or 6.

In order to locate multi-coasting points with this algorithm, three basic operations, reflection (α), expansion (γ) and contraction (β), are applied to reshape and resize the simplex. The simplex takes on a new shape and/or size when a vertex is replaced by a better one with respect to the three factors α, γ and β corresponding to the three operations. The searching process will repeat until the new vertex satisfies the expected run-time requirement in an inter-station run. There are no specific rules to assign the factors of expansion, reflection and contraction. Nevertheless, these three factors cannot be too small because a fast convergence may not be attained, nor can it be too large because the generated solution (i.e. a pair of coasting point) may be out of the boundary of the solution space. Details of the setting of α, γ and β are given in the Appendix.

3.3 Heuristic method

3.3.1 Genetic algorithm

Genetic algorithm (GA) [18-19] is an evolutionary algorithm that resembles biological processes to optimise a cost function. It is applicable in solving one and multi-dimensional searching problems. It allows a population composed of many individuals (solution of problem) to evolve to a state that optimise the fitness. There are two basic steps to have the genes evolved through successive generations, selection and replacement. The former is to decide which genes in the generation are deemed to be fit to produce off-springs whilst the latter is to allow the genes with the worst fitness to vanish in order to make room for the better off-springs to compete and survive. A new generation thus consists of the surviving and the reproduced genes of the previous generation. Through natural evolution process by mutation, crossover or other possible evolution methods, an individual with better fitness may be generated.

Using GA, every possible solution (i.e. gene) in a problem is considered as an individual. Fitness function is the selection criteria to determine the fittest genes for further evolution. The other important components of GA include initial population, fitness evaluation and generation evolution.

3.3.2 Coasting point identification by GA

When applying GA in coasting control, gene can be designed in any format but it should be well defined to the problem. As coasting point(s) is/are searched for train service regulation in an inter-station run, locations of coasting points should be integrated into the gene for evolution. In this study, gene is encoded in binary format and it represents the relative position to start coasting between stations. Attention on the assignment of the length of gene representation is quite important since gene generation error may arise if it is achieved by recombination of two genes of unequal length. (i.e. new genes generated may be out of the boundaries of the solution space).

4 Application of searching methods

4.1 Simulation setup

A number of tests have been carried out to comprehensively study the performance of the searching methods on coasting point identification. Computation time required and quality of the search outcomes are the key performance indicators.

In order to evaluate the train movement upon the chosen coasting points, a single train simulator, developed in Visual Basic, has been adopted. The train simulator calculates the details of train movement throughout the inter-station run, taking into account all factors affecting train movement, such as track geometry and traction equipment characteristics. The simulator, providing full set of user interfaces, is integrated into the searching methods. The input interfaces allow the definition on track layout, train information, traction equipment characteristics and selection of inter-station run. One of them is illustrated in Fig. 8, whilst the input and output interfaces for coast control are incorporated and shown in Fig. 9. The allowable range of run time with the flat-out runs, as well as the operation with minimum coasting speed, is determined as the reference for service regulation prior to the definition of run-time setting. The simulation is run on IBM-compatible PC with PIII CPU. A time-step of 0.2 sec is employed in the simulator as a balance between reasonable computation time and sufficient resolution of the solution.

4.2 Tests

Working with the train simulator, short and long inter-stations run with two different track layouts are chosen for investigation in a number of studies. These two inter-station track conditions are given in Tables 1 and 2.

To simply investigate the basic function of coast control in a preliminary study, the track geometry effect on the train movement calculation is neglected and the result reveals the application of a single coasting point control with different searching methods to achieve a specific run time requirement, in term of a certain run time extension with respect to the flat-out runs. Train is not allowed to operate in coasting mode until its speed exceeds the minimum coasting speed which is 45kph.

Single-coasting-point searches are undertaken with the Golden section, Fibonacci, Gradient method and GA. With GA, resolution on the coasting-position representation depends on the number of binary bits used, which is directly related to the distance between stations. The resolution of one metre is set.

The next study is to examine the relationship between the coast control and track layout. The two inter-station conditions are the same as in the previous study and their corresponding track layout characteristics are shown in Tables 3 and 4. Their related flat-out operation and desired run-time requirement are listed in Tables 5 and 6. Again, the four search methods with single coasting point identification are applied.

The third study is to explore the performance of multi-coasting-point control. The inter-station conditions and track layouts remain. Two multi-dimensional search methods, Nelder and Mead method and GA, are applied. The advantages and limitations of multi-coasting command application are also examined in the study. A fitness value, as defined in equation (1), of 0.01, and a maximum number of iterations of 20 is required in all tests.

5 Results and discussions

5.1 Results

a) Preliminary test

The run-time requirements for a short and long inter-station run, as well as the desired fitness value is 0.01 are achieved. The results are summarised in Table 7. As each iteration requires a simulation run with the single train simulator and each simulation run takes similar CPU time, number of iterations is therefore a convenient time measurement unit for performance indicator. The average number of iterations is only applicable for Gradient method and GA since a new solution depends on the gradient of the cost function and genetic information of the previous solution respectively.

From the simulation results, the classical and heuristic methods provide an acceptable solution with a reasonable average number of iterations. In general, the classical searching methods offer a smaller average number of iterations. The possibility of GA to achieve a fitter solution is reduced when the search is approaching to the desired solution. Since the coasting solution with GA is in binary representation of distance within the genes and the bits carry binary-weighted significance on the distance according to their locations, some of the useful genetic characteristics may be discarded when the mutation (reversion) of a bit is assigned to the most significant ones when the search is approaching the required solution. The randomly assigned initial population within GA is also accounted for the total number of iterations.

Simulation results also reveal that the average number of iterations is the lowest with the Gradient method. The step length and search direction of a new coasting solution with Gradient method depend on the difference between the current and previous run-times with their corresponding coasting solutions, but not the solution space. In other words, the step size between the current and new coasting solution becomes larger when the difference between run time of the current coasting location and the expected one increases. Nevertheless, the solution space with the Golden section and Fibonacci search is reduced with a specific ratio. The number of iterations in all tests with these two methods are always the same.

b) Track geometry

The simulation results with extreme track geometry are shown in Tables 8 and 9. Again, both the classical and heuristic methods can provide the desired solution within a reasonable average number of iterations. The average number of iterations with classical method is also smaller than that with the heuristic method. From Tables 7, 8 and 9, it is obvious that the average number of iterations with the Golden section, Fibonacci and Gradient method increase when the track layout is taken into account because the train movement is affected with the track layout and hence the run time is not necessarily monotonically related with the corresponding coasting location. As a result, the required fitness may not be achieved even with the maximum number of iterations in GA.

Further, it is worth noting that the average number of iterations with the track layout of positive slopes to achieve the same fitness is smaller than that with the negative one. The gradient force with the track layout of positive slope and the other train resistances are all opposite to the train movement and hence slows down the train speed. Run-time thus decreases monotonically if the coasting point is shifted from the starting station to the next and hence the search is more significantly uni-directional. Nevertheless, the gradient force with the track layout of negative slopes is against the other train resistances and it thus provides lower energy consumption with the same operation constraints in an inter-station run. In addition, a direct relationship between the current and a new coasting solution cannot be easily obtained and thus more iterations are needed.

c) Multi-coasting point

From the results summarised in Tables 10 and 11, GA provides a lower average number of iterations and a fitter solution than the Nelder and Mead method in general. The searching performance with the Nelder and Mead method is limited with the three operation factors: reflection, expansion and contraction, when resizing and reshaping the simplex during the iterative process. A new solution is more likely to be trapped out from the desired solution with Nelder and Mead methods when it is getting closer to the optimum, if the three operation factors are set to larger values. Nevertheless, the possibility of finding the optimum solution is higher with GA as it only depends on the resolution of solution.

In a short inter-station run, the algorithms usually produce a single coasting solution even they are designed to search for multiple coasting-points (i.e. the second coasting point is located within the first coasting) since there is not enough space to accommodate more coasting points. Further, it is reasonable that the location of single coasting point is more or less far away from the starting station to maintain the train speed, when compared with multiple coasting point control.

Even though these two methods obtain the solution within an acceptable number of iterations, the energy consumption of the corresponding train movement is roughly 20% higher than that with the single coasting point for the same run-time requirement in a long inter-station run. The Golden and Nelder and Mead methods are applied for the comparison between the single and multiple coasting point control. The results are summarised in Table 12 and Figures 10 and 11. The train spends more time at high speed with single coasting point, whilst it has to accelerate more at low speed with multiple coasting-points and the power consumption is thus higher. Thereby, a faster coast solution with lower energy consumption can be accomplished with single coasting point control. In addition, the second or further coasting points are only necessary when a train operates in motoring again to recover the momentum of the train movement from a low speed level because of the track geometry.

Even though the application of multiple coasting point control is not the most desirable in term of energy consumption, it is still one of the possible measures to provide a bigger solution space. To further explore the relationship between the inter-station distance and number of coasting points, a very long inter-station run is carried out to identify the necessary coasting solution with Golden and Nelder and Mead methods to achieve the operation condition as listed in Table 13. The results are illustrated in Fig. 12 and 13. Simulation result shows the energy consumption with multiple coasting point control is 5 ~ 6% less than single coasting point control and it is very encouraging. Hence, multiple coasting point control is more preferable in a very long inter-station distance run. Although there are no specific rules to identify the number of coasting points in an inter-station run, it has been shown that an inter-station distance and the track geometry are the two key factors in the application of coast control for train service regulation.

5.2 Discussions

Both classical and heuristic methods can provide an acceptable solution with reasonable computation time in all cases. The results show that a smaller average number of iterations can be attained with the classical searching methods in single coasting point control. Gradient method provides the solution with the least number of iterations on average. However, one of the main drawbacks of this algorithm is the step length and search direction cannot be determined if the slope of the cost function is not available and hence the searching process is terminated. In other words, a solution is not possible if there is no change on the run time with the current and previous coasting solutions. Therefore, the Gradient method is not applicable for an inter-station run with the extreme track geometry and speed restrictions along the line since the current and previous guess of coasting locations may be chosen from the same coasting region as described in Section 3.1.3.

Even though Golden section and Fibonacci methods are not the fastest means to obtain the solution when compared with the Gradient method, they are more applicable for practical implementation because they are not limited by the track geometry along the line and they only depend on the distance in an inter-station run. Hence, they are more robust and reliable than the other methods. Moreover, they obtain the solution within 10 iterations in all cases, which is sufficient for the operators to find the coasting solution when a train stops at stations. Therefore, the two Bi-section methods are the best to produce a single coasting solution for the regulation of train schedule, especially in a metro railway system where the inter-station distances are usually short.

Simulation results also reveal that multiple coasting point control is also useful for train service regulation. A long inter-station distance is needed to provide enough space for the multiple coasting solutions. The second or further coasting points are only necessary when the train speed slows down to a certain speed level from the first coasting and a train operates in motoring again. Since the location of first coasting point inevitably affects that of the second and so on, the solution space of the optimal coasting control becomes complicated. A lower average number of iterations cannot be achieved with Nelder and Mead method because of the operation factors of the simplex, when compared with GA. Further, a lower energy consumption of train movement cannot be achieved with multiple coast control when an inter-station distance is not long enough to release more available solution space.

5.3 Practical implementation

In practice, this "coast control" system can be an independent supervisory tool for individual train service regulation, implemented at train-level, likely on train-borne computers, and integrated in the Automatic Train Operation (ATO) with direct interface with the Automatic Train Supervision (ATS) control centre. It is not safety critical and its recommended actions are safeguarded by the other systems like Automatic Train Protection (ATP). The allowed run time between stations is forwarded to the train-borne "coasting control" system from the central control centre when the train stops at stations and the "coasting control" system then determines an appropriate speed profile accordingly. Two sets of input are required for coast control, static and dynamic. The former consists of track layout and traction equipment characteristics which are loaded onto the train-borne computer in advance; whilst the latter contains traffic conditions and operational requirements which are attained from the central control centre.

As repeated train movement calculation is needed to identify the coasting command within the course of the search, a fast microprocessor platform is necessary to quicken up the searching process for on-line implementation as there are only 30 seconds or less to find the coasting solution when train stops at stations. Selection of searching algorithms is of course a key concern from the viewpoint of software development. Their advantages and limitations of both classical and heuristic methods have been stated. An advanced high-level language such as C, C++, and Pascal can be used for software development. Further, a duplicate and hot standby hardware and software are expected to enhance the system availability and reliability.

6 Conclusions

We have presented classical and heuristic approaches to identify the necessary coasting points for service regulation in a railway metro system and revealed the feasibility of their application on coasting control. Based on the studies with different searching methods, inter-station distance and characteristics of track layout are the two important factors on the choice of searching methods for coasting control. The study shows that the average number of iterations is smaller with the Gradient based method. The main drawback is that the searching process terminates with Gradient based method if the slope of the cost function is not available in a search. Golden and Fibonacci methods, however, are more robust than the Gradient based methods as they only depend on the size of the solution space. The additional advantage of Fibonacci method is the reduction scale on the solution space in each iteration is maximised with the given number of iterations in a search.

On the other hand, the results also deduce that the heuristic approach, GA, offers a lower average number of iterations and a fitter solution with multiple coasting point control when compared with the classical one, Nelder and Mead method. Nevertheless, the distance in an inter-station run is the prime concern because there is not enough room to accommodate multi-coasting commands if the distance between stations is short. Moreover, the energy demand in a long inter-station run is slightly lower with multi-coasting points.

Railway system consists of numerous sub-systems which require stringent real-time monitoring and control because of the demanding safety standards. The results show that the searching methods are capable of providing reasonably good and fast coasting solution(s) for flexible online train scheduling control with the aid of a train simulator according to the operation requirements in all cases of the tests. In practice, dynamic coasting control has not yet been commonly applied in service regulation. It can be integrated in the on-board Automatic Train Operation (ATO) system and the coasting control command for the next inter-station run can be obtained when a train stops at a station. Even though it may not be the solution to all problems of the same nature, it certainly offers an alternative to the operators. In addition, dynamic coasting control is more flexible and efficient in the regulation of train schedule as it adapts to the current train service demand and its additional advantage is that energy saving can be achieved. From the application viewpoint, the search of coasting points within a multiple inter-station run under a specific overall run-time is a challenging proposition. The search problem becomes multi-dimensional and the solution space is huge.

7 Acknowledgements

Financial support from the Hong Kong Polytechnic University for carrying out this work is gratefully acknowledged.

8 References

[1] Mellitt B., Sujitjorn S., Goodman C.J. and Rambukwella N.B.: 'Energy Minimisation Using an Expert System for Dynamic Coast Control in Rapid Transit Trains', Conference on Railway Engineering, 1987; 48-52.

[2] Wong K.K.: 'Optimisation of Run Time and Energy Consumption of Train Movement', MSc dissertation, Hong Kong Polytechnic University, 2001.

[3] Chui A., Li K.K. and Lau P.K.: 'Traction Energy Management in KCR', IEE 2nd International Conference on APSCOM, 1993; 202-208.

[4] de Cuadra F., Fernandez A., de Juan J. and Herrero M.A.: 'Energy-Saving Automatic Optimisation of Train Speed Commands Using Direct Search Techniques', Computer in

Railway V, 1996; vol. 1, 337-346.

[5] Yasunobu S., Miyamoto S. and Ihara H.: 'Fuzzy Control for Automatic Train Operation System', IFAC Control in Transportation Systems Baden-Baden, Federal Republic of Germany 1983; 33-39.

[6] Chang C.S. and Sim S.S.: 'Optimising Train Movements Through Coast Control using Genetic Algorithms', IEE Proceedings - Electrical Power Applications, 1997; 144(1): 65-73.

[7] Wojtas B.J.: 'Development on British Railways Traction and Rolling Stock', Power Engineering Journal, March 1989; 95-102.

[8] Hill R.J.: 'Electric Railway Traction – Part 3 Traction power supplies', Power Engineering Journal, December 1994; 275-286.

[9] Hill R.J.: 'Electric Railway Traction – Part 4 Signalling and Interlocking', Power Engineering Journal, August 1995; 201-206.

[10] Mellitt B, Goodman C.J., Rambukwella N.B.: 'Optimisation of Chopper Equipment for Minimising Energy Consumption in Rapid Transit Systems', IEE Conference of Railways in Electronic Age, 1987; 34-41.

[11] Box M.J., Davies D. and Swann W.H.: 'Non-Linear Optimisation Techniques', Oliver and Boyd, 1969.

[12] Winston W.L.: 'Operations Research: Applications and Algorithms', International Thomson, 1994.

[13] Reklaitis G.V., Ravindran A. and Ragsdell K.M.: 'Engineering Optimisation – Methods and Applications', John Wiley and Sons, 1983.

[14] Shoup T. and Mistree F.: 'Optimisation Methods with Applications for Personal Computers', Prentice Hall, 1987.

[15] Douglass J.: 'Optimum Seeking Methods', Prentice Hall, 1964.

[16] Gottfried S. and Weisman J.: 'Introduction to Optimisation Theory', Prentice Hall, 1973.

[17] Bunday B. D.: 'Basic Optimisation Methods', Edward Arnold, 1984.

[18] Haupt, R.L. and Haupt, S.E.: 'Practical Genetic Algorithm', Wiley, 1998.

[19] Goldberg D.E.: 'Genetic Algorithms in Search, Optimisation and Machine Learning', Addison Wesley, 1989.

Appendix

Operation factors of the Nelder and Mead method

The effects of the three operation factors of the Nelder & Mead method on the searching performance, in terms the average number of iterations and fitness (cost), are investigated here. There is no specific rule on assigning these three operation factors and their specific range are listed as follows: -

1. Reflection factor, $\alpha > 0$;
2. Expansion factor, $\gamma > 1$;
3. Contraction factor, $0 < \beta < 1$

In order to define an appropriate range of value of the operation factors for coasting points search, 4 sets of operation factors have been chosen at regular intervals and 5 tests have been carried out to reach the specific cost value in each case under the same track layout conditions as in Tables 1 and 2.

	Nelder and Mead											
	α	γ	β	α	γ	β	α	γ	β	α	γ	β
	0.2	1.2	0.2	0.4	1.4	0.4	0.6	1.6	0.6	0.8	1.8	0.8
Average number of iterations	10.3			11.3			8.7			12*		
Fitness	0.0098			0.016			0.02			0.018		

Table A Average number of iterations with different sets of operation factors

* A specific fitness value cannot be attained in some tests up to the maximum number of iterations.

The result shows that a lower fitness and reasonable average number of iterations can be attained with smaller values on operation factors. Though a smaller average number of iterations can be reached with larger values of the operation factors, the possibility of finding the optimal solution is limited as the generated solution is more likely to be trapped out from the expected solution when the updated solution is close to the optimal point. Furthermore, the generated solution may be out of the boundary of the solution space if the three operation factors are set to a larger value. Hence, smaller values of operation factors are necessary to trade off between the number of iterations and the fitness. As a result, the three operation factors α, γ and β are set at 0.2, 1.2 and 0.2 respectively.

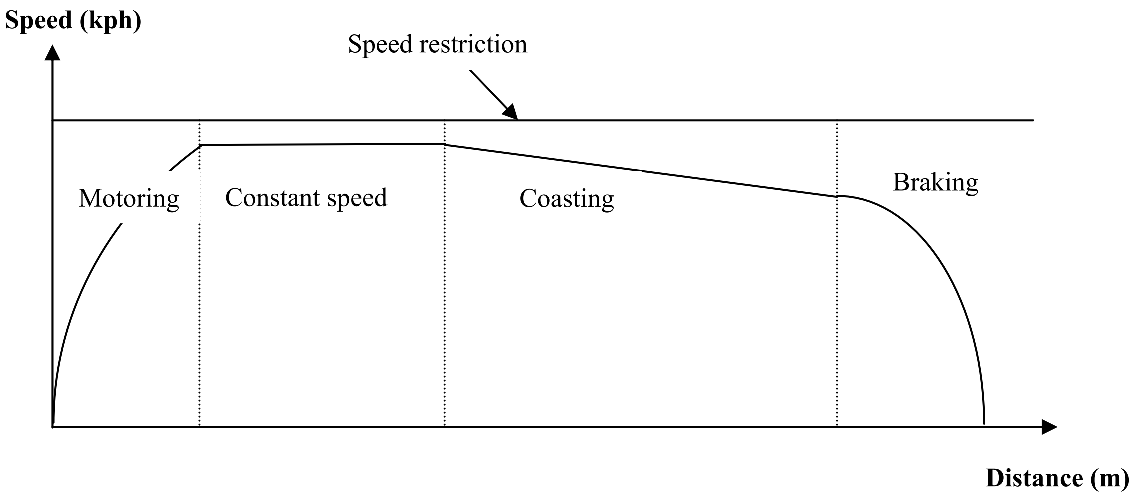

Fig. 1 Speed profile of a simple inter-station run

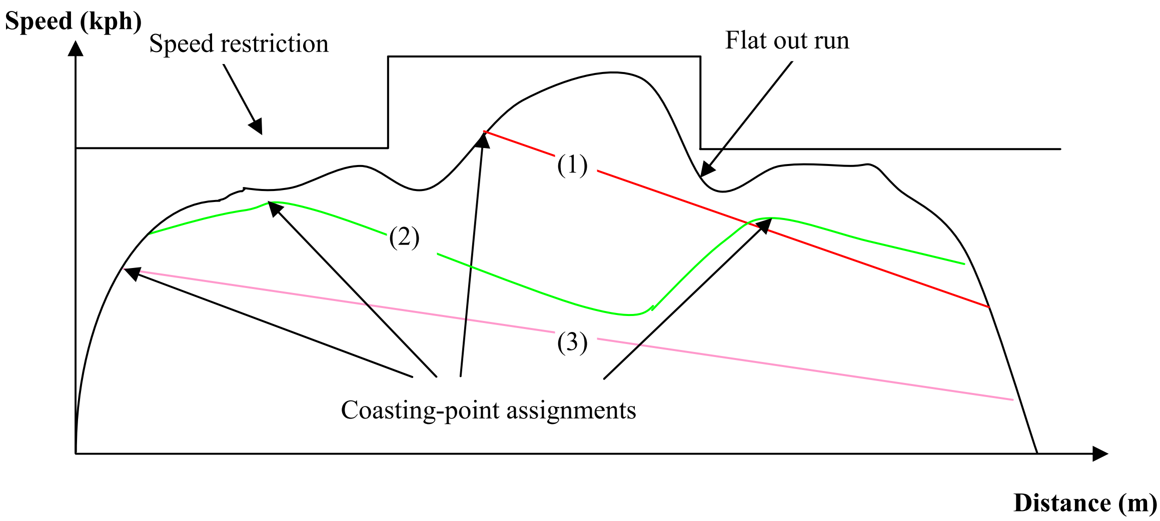

Fig. 2 Speed profiles of flat-out run and some possible coasting-points

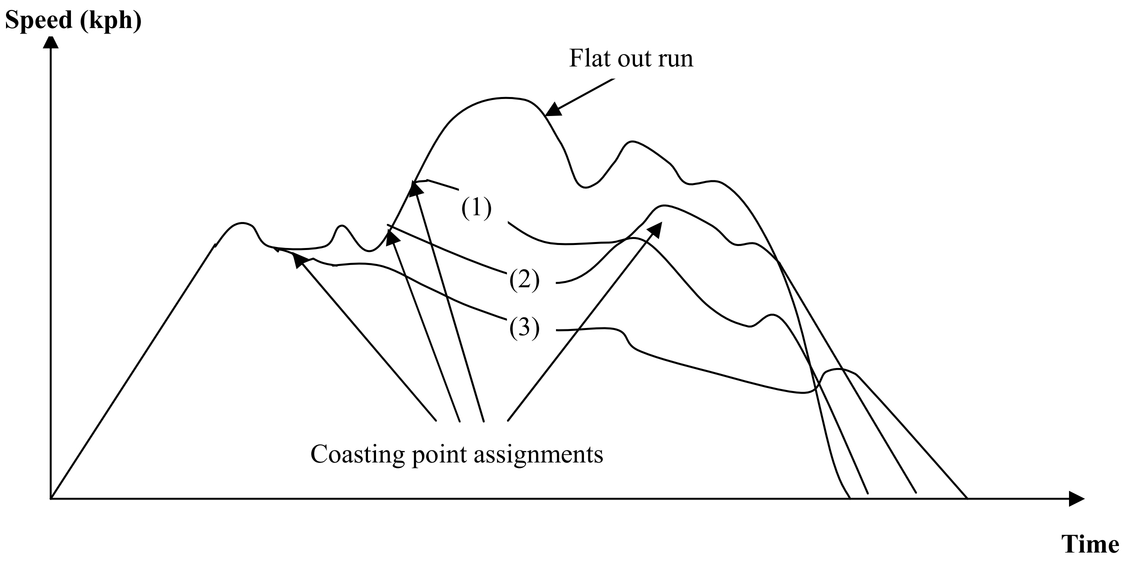

Fig. 3 Run-time extensions with some possible coasting points

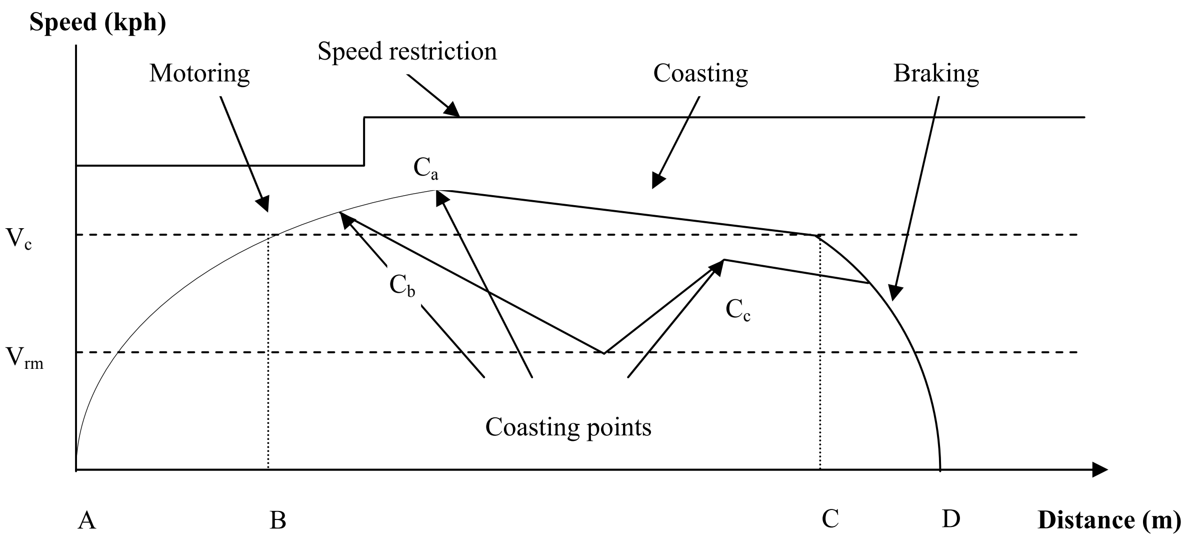

Fig. 4 The range of possible coasting point

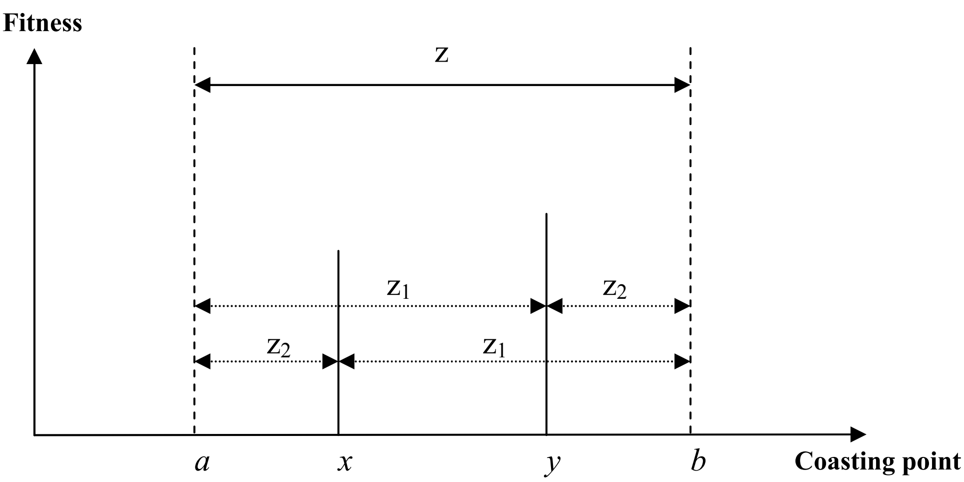

Fig. 5 Golden section search

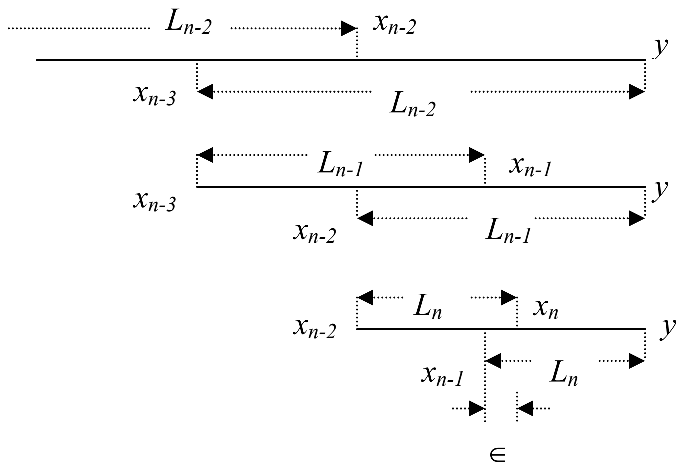

Fig. 6 Sequence of uncertainty intervals in a Fibonacci search

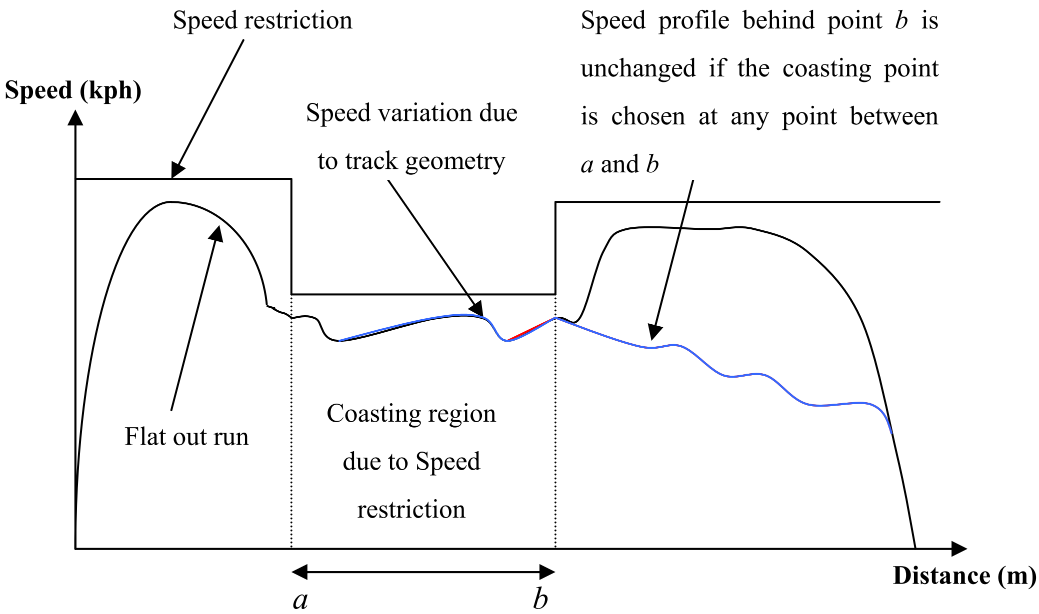

Fig. 7 Coasting assignments without adjustment on run time

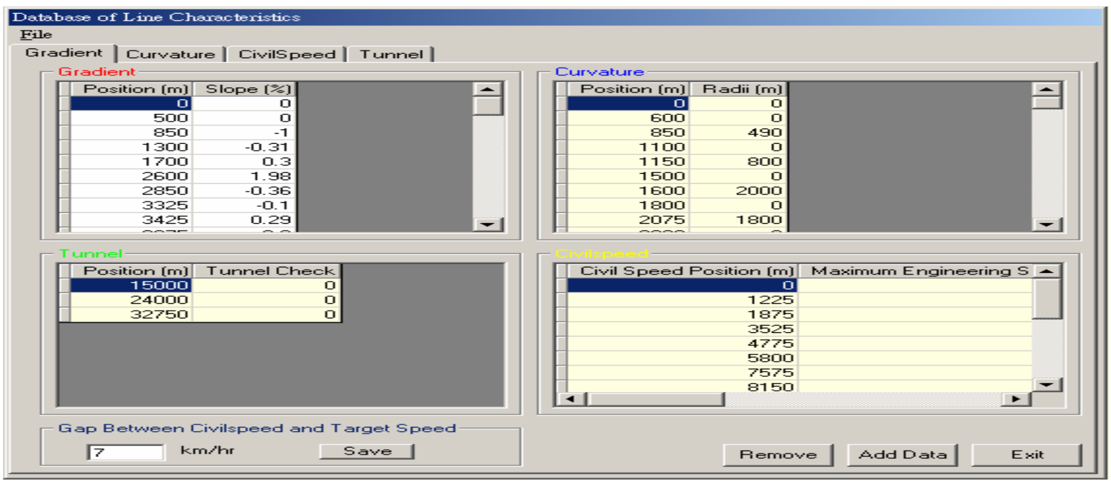

Fig. 8 Input interface for track layout characteristic

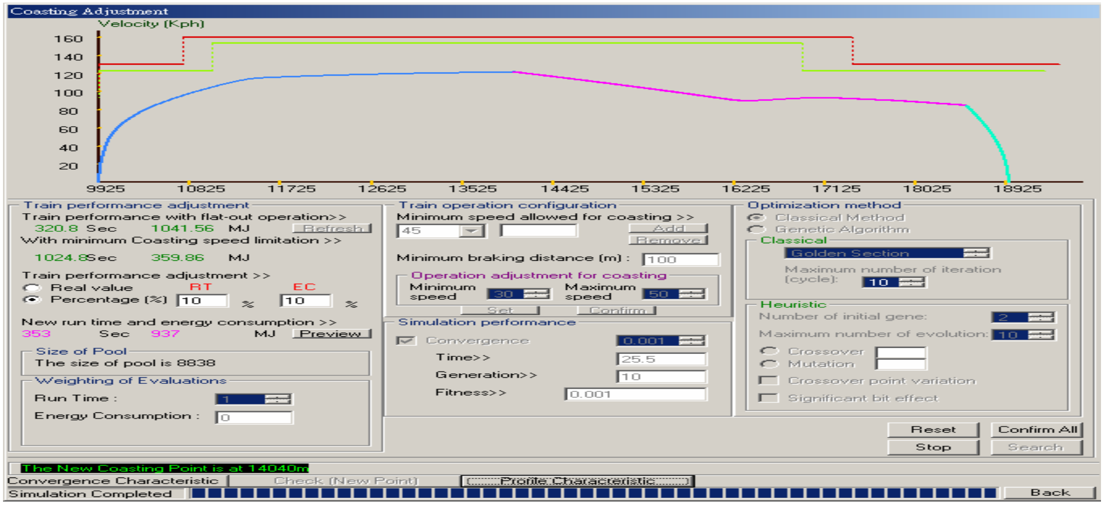

Fig. 9 Output interface with coast control performance

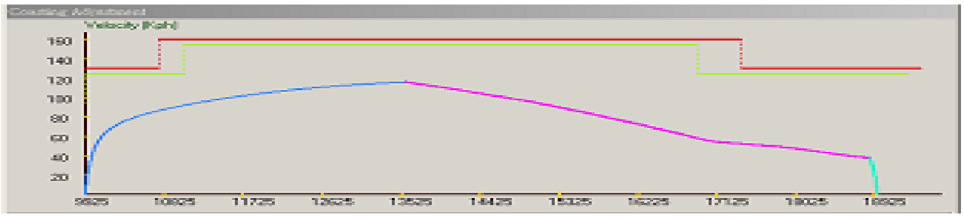

Fig. 10 Speed profile of train with single coast control

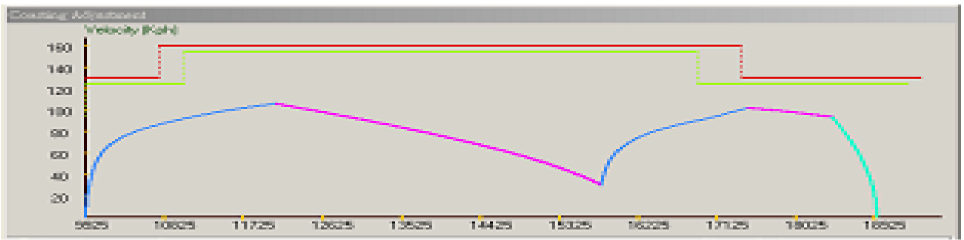

Fig. 11 Speed profile of train with multiple coast control

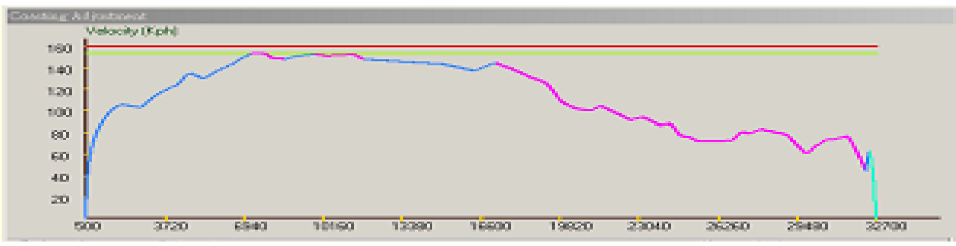

Fig. 12 Speed profile of train with single coast control in a very long inter-station run

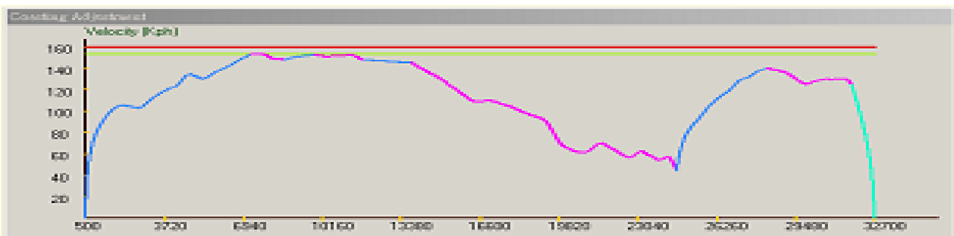

Fig. 13 Speed profile of train with multiple coast control in a very long inter-station run

Short inter-station run	
Inter-station distance:	1.1 km
Max. line speed:	80 kph
Min. coasting speed:	50 kph
Slope effect	✕
Flat-out operation	81.8 sec and 181.37 MJ
Run time extension	8% (i.e. 88 sec)

Table 1 Short inter-station operation conditions

Long inter-station run	
Inter-station distance:	9 km
Max. line speed:	130 kph
Min. coasting speed:	50 kph
Slope effect	✕
Flat-out operation	309 sec and 1004.6 MJ
Run time extension	12% (i.e. 346sec)

Table 2 Long inter-station operation conditions

Section (m)	Slope (%)
0 ~ 150	0.3
150 ~ 600	1.08
600 ~ 1100	0.3

Table 3 Track layout of a short inter-station run

Section (m)	Slope (%)
0 ~ 125	0
125 ~ 1325	0.76
1325 ~ 6225	0.8
6225 ~ 6975	0.76
6975 ~ 7775	0
7775 ~ 8475	0.2
8475 ~ 8675	0.11
8675 ~ 9025	0

Table 4 Track layout of a long inter-station run

Short inter-station run	
Slope effect	Table 3 with positive value
Flat-out operation	82.4 sec and 233.75 MJ
Run time extension	7% (i.e. 88 sec)
Long inter-station run	
Slope effect	Table 4 with positive value
Flat-out operation	334.6 sec and 1198.23 MJ
Run time extension	3.5% (i.e. 346 sec)

Table 5 Short and long inter-station condition with positive slope effect

Short inter-station run	
Slope effect	Table 3 with negative value
Flat-out operation	80 sec and 169.9 MJ
Run time extension	10% (i.e. 88 sec)
Long inter-station run	
Slope effect	Table 4 with negative value
Flat-out operation	292.6 sec and 668.1 MJ
Run time extension	18.2% (i.e. 346 sec)

Table 6 Short and long inter-station condition with negative slope effect

Searching methods	Average number of iterations	Run time with the corresponding coasting solution (sec)
Short inter-station run		
Golden*	4	88.4
Fibonacci*	4	88.4
Gradient	2.8	87.8 ~ 88.2
GA	5.6	89
Long inter-station run		
Golden*	4	344.6
Fibonacci*	4	344.6
Gradient	3.4	345.6 ~ 347.4
GA	11.8	346 ~ 349

Table 7 Average number of iterations with no track effect

*Average is not applicable as the number of iterations is the same of all tests

Searching methods	Average number of iterations	Run time with the corresponding coasting solution (sec)
Short inter-station run with positive slope (i.e. uphill)		
Golden*	6	88.4
Fibonacci*	6	88.6
Gradient	3.2	87.8 ~ 88.4
GA	10.3	86 ~ 89
Long inter-station run with positive slope (i.e. uphill)		
Golden*	6	345.8
Fibonacci*	6	345.6
Gradient	4.2	346.2 ~ 347.4
GA	6.9	347

Table 8 Average number of iterations with positive track layout

*Average is not applicable as the number of iterations is the same of all tests.

Searching methods	Average number of iterations	Run time with the corresponding coasting solution (sec)
Short inter-station run with negative slope (i.e. downhill)		
Golden*	6	87.8
Fibonacci*	6	88
Gradient	3.5	87.8 ~ 88.2
GA**	5	87 ~ 88
Long inter-station run with negative slope (i.e. downhill)		
Golden*	9	346.2
Fibonacci*	10	346.8
Gradient	6.3	344.8 ~ 347
GA**	6.3	346 ~ 350

Table 9 Average number of iterations with negative track layout

*Average is not applicable as the number of iterations is the same of all tests.

**The required fitness of 0.01 cannot be achieved at the maximum number of iterations.

Searching methods	Average number of iterations	Run time with the corresponding coasting solution (sec)
Short inter-station run with no slope		
Nelder & Mead	13.67	86.2 ~ 87.6
GA	6.3	88 ~ 90
Short inter-station run with positive slope (i.e. uphill)		
Nelder & Mead	7.17	87.2 ~ 88
GA	5.56	88 ~ 89
Short inter-station run with negative slope (i.e. downhill)		
Nelder & Mead	9.8	86.6 ~ 87.2
GA	6	87 ~ 88

Table 10 Average number of iterations with multi coasting point control in a short inter-station run

Searching methods	Average number of iterations	Run time with the corresponding coasting solution (sec)
Long inter-station run with no slope		
Nelder & Mead	9.6	344.2 ~ 348.4
GA	8.2	346 ~ 348
Long inter-station run with positive slope (i.e. uphill)		
Nelder & Mead	12.75	344.6 ~ 346.8
GA	12.1	347 ~ 348
Long inter-station run with negative slope (i.e. downhill)		
Nelder & Mead	8.5	343 ~ 348.4
GA	9.8	346 ~ 350

Table 11 Average number of iterations with multi coasting point control in a long inter-station run

Remark: The required fitness of 0.01 may not be attained at the maximum number of iterations in both methods.

Inter-station distance:	9 km	
Max. line speed:	160 kph	
Run time extension	35% run time of the flat out run	
Energy consumption of train movement with the corresponding coasting solution (MJ)	Single point	Multi point
	1089.9 (see Figure 10)	1306.7 (see Figure 11)
Searching method	Golden	Nelder and Mead

Table 12 Comparison of the train movement performance with single and multi coast control to achieve the same run time requirement

Inter-station distance:	30 km	
Max. line speed:	160 kph	
Run time extension	30% run time of the flat out run	
Energy consumption of train movement with the corresponding coasting solution (MJ)	Single point	Multi point
	2675.8 (see Figure 12)	2535 (see Figure 13)
Searching method	Golden search	Nelder and Mead

Table 13 Comparison of the train movement performance with single and multi coast control in a very long inter-station run