An Energy-Efficient Train Control Framework for Smart Railway Transportation

Jin Huang, Yangdong Deng, *Member, IEEE*, Qinwen Yang, and Jiaguang Sun

Abstract—Railway transportation systems are the backbone of smart cities. With the rapid increasing of railway mileage, the energy consumption of train becomes a major concern. The uniqueness of train operations is that the geographic characteristics of each route is known a priori. On the other hand, the parameters (e.g., loads) of a train varies from trip to trip. Such a specialty determines that an energy-optimal driving profile for each train operation has to be pursued by considering both the geographic information and the inherent train conditions. The solution of the optimization problem, however, is hard due to its high dimension, nonlinearity, complex constraints and time-varying characteristics of a control sequence. As a result, an energy-saving solution to the train control optimization problem has to address the dilemma of optimization quality and computing time. This work proposes an energy-efficient train control framework by integrating both offline and onboard optimization techniques. The offline processing builds a decision tree based sketchy solution through a complete flow of sequence mining, optimization and machine learning. The onboard system feeds the train parameters into the decision tree to derive an optimized control sequence. A key innovation of this work is the identification of optimal patterns of control sequence by data mining the driving behaviors of the experienced train drivers and then apply the patterns to online trip planning. The proposed framework efficiently find an optimized driving solution by leveraging the training results derived with a compute-intensive offline learning flow. The framework was already testified in a smart freight train system. It was demonstrated an average of 9.84 percent energy-saving can be achieved.

Index Terms—smart railway transportation, trip planning, train control, energy efficient, framework

1 Introduction

Due to their capability of enabling massive transportation in an efficient and convenient manner, advanced railway systems have been and will continue to be the backbone of any smart cities [1], [2]. At the present time, energy efficiency is one of the most essential topics for building the infrastructure of smart cities. With the fast-growing railway mileage, however, the energy consumption of trains have become a major concern. For instance, the annual energy (converted into electrical energy) used by the trains of Chinese railway systems amounts to 142 billion kilowatt-hour [3], which is around 0.4 percent of the total energy consumption of China. In fact, the energy consumption of the whole railway system is even higher than that of major metropolitan cities like Shanghai. On the other hand, the uniqueness of train operations offers significant opportunities for energy optimization. In fact, given a railway route, the geographic characteristics is known a priori, and only the parameters (e.g., loads) of a train vary in each run. As a result, an energy-optimal driving profile for each train operation can be pursued by considering the geographic and inherent conditions. In practice, the optimization problem is extremely hard to solve due to its high dimension, nonlinearity, complex constraints and time-varying characteristics of control sequences.

In the past decades, significant research and engineering efforts have been dedicated to deriving the optimal driving sequence for railway transportation systems. Many advanced numerical and heuristic techniques have been proposed to calculate the optimal trajectory of the train operation, *i.e.*, train trip profile. Han et al. [4] and Li and Hou [5] used genetic algorithm to construct the optimal reference trajectory that optimizes train control for energy efficiency. Wang et al. [6] proposed two approaches to solve this optimal control problem under various constraints like the fixed arrival time treating the energy consumption and the riding comfort as a trade-off in the objective function. A commercial system for operating a train equipped one or more locomotives were developed by Kumar et al. [7]. Many researchers extended the optimization problem to treat the railway systems as a whole. Caprara et al. [8] described the main optimization problems that are faced in the planning of a passenger railway system. The problems range from the definition of the routes and frequencies of the trains in the railway network to the construction of the duties and the schedules of drivers and conductors. The minimization of the total energy consumption of railway systems was discussed by Miyatake and Ko by taking advantage of numerical methods such as dynamic programming (DP), gradient method, and sequential quadratic programming (SQP) [9]. An evolutionary algorithm based Pareto optimization approach for speed-tuning in a railway system was presented in [10] by optimizing both travel duration and energy saving. The solution provides a set of diversified non-dominated solutions to the decision-maker.

- J. Huang, Y. Deng, and J. Sun are with the School of Software, TNLIST, KLISS, Tsinghua University, Beijing 100084, China. E-mail: {huangjin, dengyd, sunjg}@tsinghua.edu.cn.
- Q. Yang is with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha, Hunan 410082, China. E-mail: yangqw@hnu.edu.cn.

Manuscript received 15 Feb. 2015; revised 6 Oct. 2015; accepted 5 Nov. 2015. Date of publication 12 Nov. 2015; date of current version 13 Apr. 2016. Recommended for acceptance by S. Hu, G. Betis, R. Ranjan, and L. Wang. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TC.2015.2500565

Su et al. [11] developed the concept of optimized integrated timetable, which includes both the timetable and the speed profiles. The above approaches offers superior solutions but tend to be compute intensive. As a result, they are more aligned to offline processing and too time-consuming for onboard real-time control.

Since each train operation has varying system parameters (e.g., loads), onboard optimization of the train trip profile is essential to improve energy efficiency. Online real-time control techniques for train operations were also widely studied. Salmasi [12] classifies and extensively overviews the state-of-the-art control strategies for hybrid electric vehicles. Ding [13] proposed an optimal driving model for energy-efficient operation of trains under fixed block and mobile block conditions, and designed a corresponding heuristic optimization algorithm. Ke et al. [14] presented a heuristic method to optimize the train-speed trajectory and control sequence by considering track gradient, average speed, restriction of train speed, acceleration and jerk. The problem of Automatic Train Operation (ATO) system with multiple working conditions was investigated by Wang et al. [15]. Gao et al. [16] presented a neuro-adaptive robust control method for automatic train operation which is subject to unknown systematic time-varying dynamics. The above real-time techniques attained the online processing capability, but suffered from the lack of guarantee for a sufficiently optimized solution.

From the above analysis, it can be seen that an energy-saving solution to the train control optimization problem has to address the dilemma of optimization quality and real-time control. With the fast growth of smart railway transportation technology [17], [18], [19], it is critical to develop energy efficient driving solutions for trains. This work proposes an energy-efficient train control framework for smart railway transportation by integrating both offline and onboard optimization techniques. A key innovation of this work is to identify optimal patterns of control sequence by data mining the driving behaviors of the experienced train drivers and then apply the patterns to online trip planning. The proposed framework efficiently find an optimized driving solution by leveraging the training results derived with a compute-intensive offline learning flow. The offline processing performs data mining on the driving behavior of experienced drivers and builds a decision tree to encode sketchy solutions through a complete flow of sequence mining, optimization computation, sequence modification and machine learning. The onboard system feeds the train parameters into the decision tree to derive an optimized control sequence. The proposed approach was testified in a real smart freight train system. It was demonstrated a 9.84 percent energy-saving can be achieved. This works offers a working example for solving compute-intensive optimization and control problems, and has great potential for other smart city system applications.

The paper is structured as follows. Section 2 formulates the train trip optimization problem and introduces the engineering requirements. Section 3 presents the design methodology of the proposed trip planning framework by properly combining both offline and online processing modules. Sections 4 and 5 elaborates the details of the online process and offline processing flows, respectively. Then in Section 6, we give a practical application to demonstrate the effectiveness of the proposed approach. Section 7 summarizes the contents of this work.

2 Problem Formulation

In this work, we focus on optimize the energy consumption of trains. It can be formulated a trip planning problem based on the motion dynamics of a train as well as the route information. The optimization objective is to minimize the energy consumption as well as the time deviation under various constraints. The solution is a control sequence consisting a series of discrete or continuous levels of the control throttle that set the traction or braking force.

2.1 Train Motion Dynamics

The mass-point model is usually employed to capture the dynamics of a train. It is formulated as follows [20]:

$$\begin{aligned} m\rho\frac{\mathrm{d}v}{\mathrm{d}t} &= f(s) - R_b(v) - R_l(s), \\ \frac{\mathrm{d}s}{\mathrm{d}t} &= v, \end{aligned} \tag{1}$$

where m is the mass of the train, ρ is a factor accounting for the rotating mass, v is the velocity of the train, s is the position (i.e., displacement) of the train, $f(s)$ is the traction or braking force along the position, which is bounded by the maximum traction force f_{tmax} ($f_{tmax} > 0$) and the maximum braking force $f_{bmax}(f_{bmax} > 0)$, $-f_{bmax} < f(s) < f_{tmax}$, $R_b(v)$ is the basic resistance including both roll resistance and air resistance, whose empirical equation can be defined as [21]

$$R_b(v) = m(a_1 + a_2 v + a_3 v^2), \tag{2}$$

where the coefficients a_1, a_2 and a_3 depend on the train characteristics and can be obtained through certain experiments. $R_l(s)$ is the line resistance caused by track slope, curves and tunnels, whose empirical equation is given by [21]

$$R_l(s) = m * g * \sin\alpha(s) + f_c(r(s)) + f_t(l_t(s), v), \tag{3}$$

where g is the gravitational acceleration, $\alpha(s)$, $r(s)$ and $l_t(s)$ are the slope, the radius of the curve, and the length of the tunnel along the track, respectively. When running in a tunnel, compared to normal running status, the train experiences a higher air resistance that depends on the tunnel shape, the smoothness of tunnel walls, the exterior surface of the train, and so on. The curve resistance $f_c(\cdot)$ and the tunnel resistance $f_t(\cdot)$ can be given by empirical equations [20]

$$\begin{aligned} &f_c(r(s)) = 6.3/(r(s) - 55)m \quad \text{for} \quad r(s) \geq 300m, \\ &f_c(r(s)) = 4.91/(r(s) - 30)m \quad \text{for} \quad r(s) < 300m, \\ &f_t(r(s)) = \frac{l_t v(s)^2}{10^7} m. \end{aligned} \tag{4}$$

It should be noted that different trains may exhibit different resistances reflected by different coefficients in (4).

For numerical calculation, it is more convenient to choose the position s as an independent variable rather than the time t. Such a treatment simplifies the consideration of track-related data, such as line resistance and speed limits

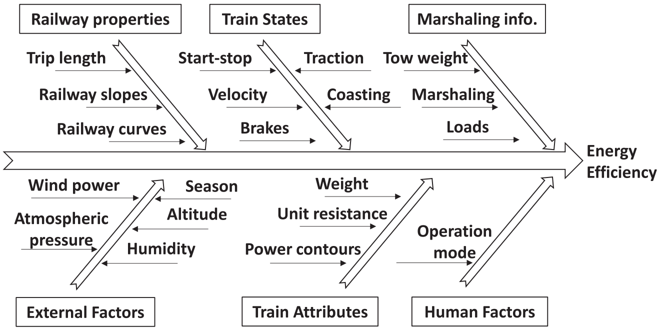

Fig. 1. The effect factors for energy efficiency in smart railway transportation.

[22]. In addition, the solution complexity of the optimization problem can be considerably lowered. The choice of kinetic energy instead of speed v facilitates the solution of the optimal control problem by eliminating some of the model nonlinearities. Therefore, we select kinetic energy per mass unit as $K = 0.5v^2$. The train motion can then be captured as the following continuous-space model

$$\begin{aligned} m\rho \frac{\mathrm{d}K}{\mathrm{d}s} &= f(s) - R_b\big(\sqrt{2K}\big) - R_l(s), \\ \frac{\mathrm{d}t}{\mathrm{d}s} &= \frac{1}{\sqrt{2K}}, \end{aligned} \tag{5}$$

with all the terms defined as previously.

2.2 Train Trip Optimization Problem

The optimal energy consumption for a train on a given trip is hard to derive due to its high dimension, inherent nonlinearity, various constraints and the potential variation of the sequence elements. As listed in Fig. 1, many factors including railway properties, train states, train attributes, marshaling information, human factors, as well as other external factors, all play a role on the energy efficiency.

This work addresses the train trip optimization problem that is aimed to find an energy optimal problem by considering train dynamics, loads and route. The problem can be formulated as an general optimization problem [22]. The traction or braking force $f(s)$ is the control input, which is determined by the discrete or continuous level of the throttle for most of the railway locomotives. The state variables are the train position s and speed v. The objective functions to be minimized can be the trip time deviation from the schedule and the energy consumption for a given trip time, or both time deviation and energy consumption. In this paper, we consider both the energy consumption and time deviation in the objective functions, all other factors such as safety issue are taken as constraints.

By employing the train motion dynamics model (5), the optimization objectives can then be stated in the position depended form as:

$$\begin{aligned} J_E &= \int_{s_{start}}^{s_{end}} \phi(f)\left(f(s) + \lambda\left|\frac{\mathrm{d}f(s)}{3\mathrm{d}s}\right|\right), \\ J_T &= \big|T - \bar{T}\big|, \end{aligned} \tag{6}$$

subject to the following constraints

$$\begin{aligned} &f_{min} \le f(s) \le f_{max}, \\ &0 \le T(s) \le T_{max}(s), \\ &v(s) \le v_{limit}(s), \end{aligned} \tag{7}$$

as well as the following boundary conditions

$$\begin{aligned} s(0) &= s_{start}, \quad v(0) = v_{start}, \\ s(T) &= s_{end}, \quad v(T) = v_{end}. \end{aligned} \tag{8}$$

Here, J_E and J_T represent the optimization objective on energy consumption and the time deviation, respectively. $\phi(f)$ stands for the throttle depended coefficient. $\bar{T}$ is the scheduled time for a train trip and T as the real time cost for the trip. The maximum allowable velocity $v_{limit}(s)$ depends on the train characteristics and the line conditions, and thus it is usually a piecewise constant function of the coordinate s; s_{start} and v_{start} are the position and the velocity at the beginning of the route; s_{end} and v_{end} are the position and the velocity at the end of the route. The duration of the trip $\bar{T}$ is usually given by the timetable.

Here we assume that the unit kinetic energy $\triangle E(s) > 0$, which means the speed of the train is always strictly larger than zero and the train travels in a non-stop manner on the given trip. Here, we should stress that there is a possible trade-off between the energy consumption and the time deviation. The objective is to minimize the objective functions including both J_E and J_T under proper weights.

3 A Systematic Solution for Trip Optimization

The train trip optimization is a typical multi-constrained and nonlinear optimization problem. Generally a railway line is decomposed into multiple sections that categorized as steep uphill, uphill, flat, downhill, or steep downhill. Besides the geographical characteristics, each section has a specific speed limit. The whole trip time is also subject to pre-defined limit. The output of the trip optimization problem is a sequence of control throttle settings. Although existing methods such as genetic algorithms and neural networks can eventually obtain an optimized control sequence, the shortcomings of these approaches are obvious. They cannot guarantee the consistence of the returned results and the compute time required by onboard real-time control systems. On the other hand, heuristic and rule-based real-time control approaches enable a relatively high computing efficiency, but their empirical nature can hardly guarantee the optimality.

In this work, we propose an integrated approach consisting of both offline and online techniques as illustrated in Fig. 2. The objective is to satisfy the requirements for both optimization quality and computing efficiency. The offline and online processing is built around the concept of the parameter decision tree, which encodes pre-computed optimization results for various conditions and will be elaborated in Section 4. The offline processing consists of a complete flow of sequence mining, optimization and machine learning so as to identify the empirically optimal

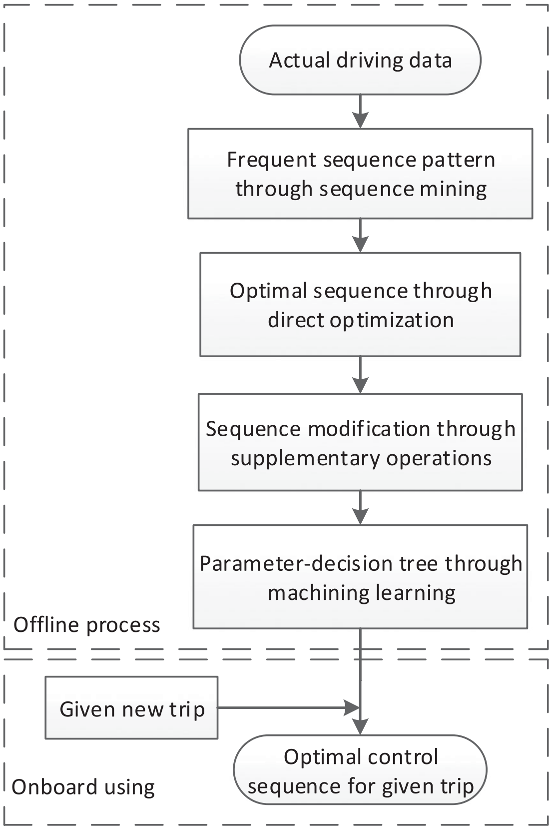

Fig. 2. Systematic solution for trip planning integrating both offline and online processing.

human driving behaviors and organize them into a decision tree, while the onboard system feed train- and route-specific information to the tree to derive an optimized control sequence.

The offline processing consists of three steps. Firstly, sequence mining techniques are employed for mining frequent patterns from the recorded human driving data. Secondly, after getting the frequent sequence patterns, optimization techniques are utilized for searching optimal sequence throttle selection and proportion, supplementary operations may still be needed as in searching a relatively

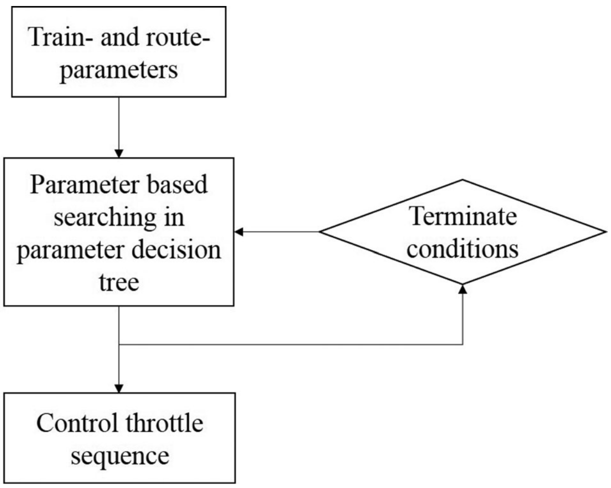

Fig. 3. The application process of the parameter decision tree.

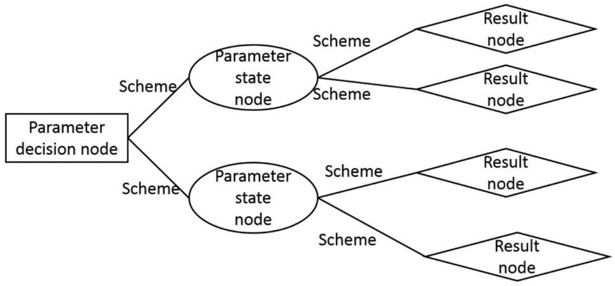

Fig. 4. The general structure of a parameter decision tree.

optimal sequence. Thirdly, machine learning techniques are used to build the index between the parameters of the trip and the driving strategies.

The parameter decision tree built through the offline processing is coded into the onboard system for online driving decision making. Given locomotive and route parameters, the onboard optimization is reduced to a search problem. A fast traversal of the parameter decision tree will return optimal throttle control, as shown in Fig. 3. The search algorithm is designed to guarantee that a solution can be found under all conditions. A reference condition for terminating the parameter searching process will guarantee a reasonable short time for calculation, which may only allow several seconds.

The proposed framework combined the best of the two worlds. The offline module focuses on compute-intensive knowledge discovery, while the onboard system takes advantage of the pre-computed solutions for efficient online decision making.

4 Parameter Decision Tree Based Onboard Trip Optimization

In our trip optimization system, the onboard processing flow is built around the parameter decision tree, in which the human driving knowledge is approximated as a discrete-valued target function represented by a decision tree [23]. The parameter decision tree is formally defined as a tree-shaped diagram that can be used to determine a course of actions. Each branch of the decision tree represents a possible decision or occurrence. The tree structure defines how a sequence of parameters lead to a given decision, while different paths in the tree imply mutually exclusive situations[24]. A general parameter decision tree structure is shown in Fig. 4. In this paper, the parameter decision trees classifies the learned sequence instances according to the combination of parameters. Fig. 5 shows an instance of parameter decision tree for organizing the optimization rules used in this research. Each node in the tree specifies a test of a given parameter and dictates the child node, i.e., the next parameter, to evaluate. A traversal of the tree finds a sequence of satisfied conditions and leads to a pre-computed trip optimization decision under such conditions.

For a given railway line, the parameter decision tree is built offline and we will elaborate the construction in Section 5. The online system performs the trip optimization under two scenarios. The first scenario corresponds to the trip initialization stage when the load, speed limit for each section and other parameters are known. The online processing

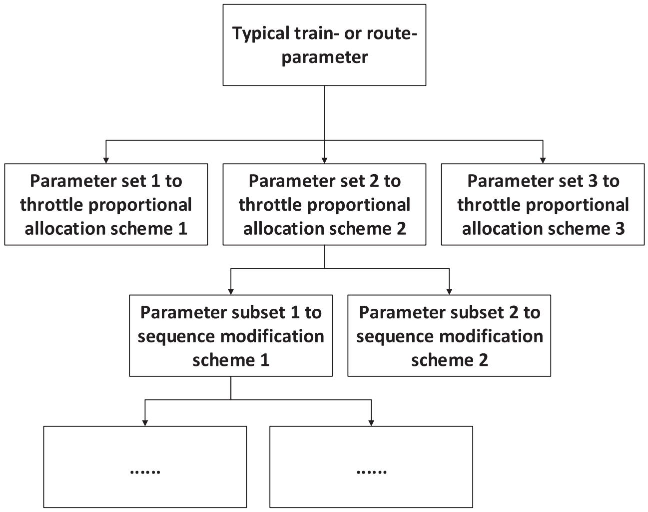

Fig. 5. An instance of parameter decision tree for organizing the optimized sequence rules.

module feed the trip specific information to the parameter decision tree and perform a traversal for each section of the current line to compute the optimized throttle allocation. The sections from the starting station to the destination are processed sequentially because the decision for the current section depends on the speed of the previous section. In other words, we consider the joint velocity between any two adjacent sections. The optimization process terminates when a complete throttle sequence is constructed for all sections. The second scenario targets the dynamic change of driving conditions such as a temporary change of speed limit. Now the online processing module dynamically re-compute the driving profile for those sections subject to change.

The parameter decision tree serves as a bridge between the driving knowledge discovery and the online trip optimization processes. Its concise structure enables efficient online derivation of high-quality driving decisions.

5 The Off-Line Processing

The offline processing consists of three major steps, the sequence mining, optimization and machine learning. Different from the online processing module, these off-line steps are designed to guarantee the solution quality of energy optimization. The remaining of this section elaborates the above steps.

5.1 Frequent Pattern Mining

An exact solution of the energy-optimal train trip profile is expensive, if not completely infeasible. On the other hand, experienced human drivers are able to achieve superior energy saving. Such an observation motivates us to develop effective data mining techniques for the discovery of energy-efficient human driving behaviors. Empirical data suggest that there exist patterns in human drivers' responses to typical combinations of line and train conditions. As a result, we perform frequent pattern mining on a large collection of human driving data samples.

In this work, we used the train-borne monitoring devices to collect the manipulations of throttle under five typical geographical features, steep uphill, uphill, flat, downhill, and steep downhill, of railway lines. We classified the sample data from human drivers by the range of the load for a set of lines. We chose driving behaviors leading to the best 50 percent of energy consumption. The driving data are preprocessed with two steps, noisy data filtering and data generalization.

In this work, we employ the Apriori-all algorithm [25] for frequent pattern mining. It proceeds by identifying the frequent individual items in the database and extending them to larger itemsets as long as they appear sufficiently often in the database. The frequent itemsets can be used to determine association rules that reflect general trends in the database.

Algorithm 1. Optimized Apriori-All Algorithm

Input: Original time series data of the drivers' driving records (maximal length n)
Output: Maximal frequent operation pattern P_{max}

$iteration\ counter \leftarrow 1$;
while $true$ **do**
  if $i == 1$ **then**
    Generate all single operation;
  else
    Generate $i\mathrm{th}$ candidate frequent pattern set from $(i-1)\mathrm{th}$ frequent pattern;
  for *candidate* $i\mathrm{th}$ *frequent pattern item* s **do**
    Set $sum = 0$;
    Generate two successive sub-pattern s_1 and s_2 split in the middle of p;
    for $j = 1; j \leq n; j++$ **do**
      $sum += Support_{s_{1j}} + Support_{s_{2j\rightarrow n}}$;
    if $sum \geq Support_{threshold}$ **then**
      Scan s from the original series data, count and judge;
      if $Support_s \geq Support_{threshold}$ **then**
        Add s into $i\mathrm{th}$ frequent pattern set;
  if $i\mathrm{th}\ frequent\ pattern\ set\ is\ 0$ **then**
    break;
  $i = i + 1$;
return$(i-1)\mathrm{th}$ frequent pattern set;

Pseudocode for employing the Apriori-all for frequent data mining from driver's records is given in Algorithm 1. For the Apriori-all algorithm, the length of a sequence is the number of itemsets in the sequence, a sequence of length k is called a k-sequence, each itemset in a large sequence must have minimum support. An itemset with minimum support is called a large (frequent) itemset or litemset. The process is given as follows: At the beginning, the entire data sources are scanned to determine whether each item has reached the minimum support level and finally select first frequent itemset. For each iteration, taking the $(K+1)$th iteration for example, the candidate frequent itemsets are generated according to the property of Apriori-all from the Kth frequent itemset. Then the database is scanned to determine whether it is larger than the minimum support level. Finally the $(K+1)$th frequent itemset is obtained. The whole iterative process ends when the set is empty. At this time the frequent itemset from the previous iteration is the maximum frequent itemset in data sources, which contains the sequence with the longest sequence length. The outputs are five sets of frequent sequence patterns, corresponding to the five types of railway conditions.

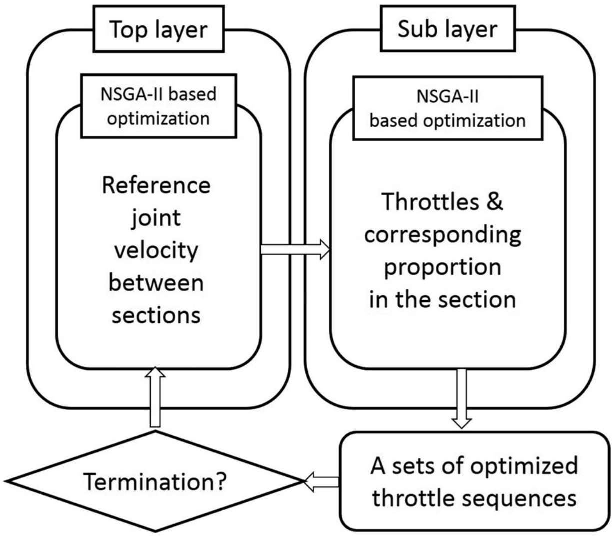

Fig. 6. The process of the NSGA-II algorithm for control sequence optimization.

5.2 Control Sequence Optimization

After getting the frequent sequence patterns, a series of optimization operations are subsequently carried out. For a multi-objective optimization problem, Pareto optimal solutions are those that cannot be improved in any of the objectives without degrading at least one of the other objectives. The set of Pareto optimal solutions is often called the Pareto front [26]. In getting the Pareto front for optimal throttle sequence selection, we chose a two-layer optimization method by using NSGA-II as the optimization engine. NSGA-II is a classic multi-objective optimization algorithm that has been applied in many fields of science and engineering domains where trade-offs between two or more conflicting objectives need to be considered [27]. The process of the two-layer optimization is illustrated in Fig. 6.

We designed a two-layer optimization flow for better convergence behaviors of the global optimization. The top layer is deployed to search the optimal reference joint velocity of different sections along a railway line. The results can be considered as the basis of sub-layer optimization. The sub-layer aims at the throttle selection and the allocation of traction (or braking efforts) among sections. Within each layer, we use NSFGA-II to derive the Pareto front consisting of multiple irreducible optimal solutions.

In the optimization process, a simulation platform is built to derive the dynamics according to the system motion Equation (5). With the simulation platform, the sub-layer optimization evaluates the time and oil consumption of a given solution of the throttle selection determined by the top layer. The top layer gradually adjusts the reference joint velocity and dispatches it to the sub-layer to generate a sets of optimized control sequences. When the number of iteration reaches a maximum number or the selection reach a certain threshold, the optimization process terminates and then returns the best N optimized control sequences. A weighted objective function is defined by considering both the trip time and energy consumption to evaluate the performances of different control sequences.

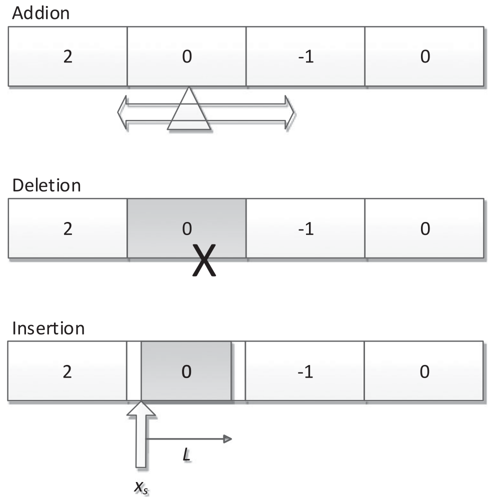

Fig. 7. Sequence modification operations during optimizing searching.

$$J(n) = w_1 J_E(n) + w_2 J_T(n), \tag{9}$$

where w_1 and w_2 are two weighting factors reflecting the relative importance of the trip time and the energy consumption.

The above direct searching process may not satisfy the requirements of onboard system control due to the high dimension of results given by the frequent pattern mining process. Especially, the derived patterns may not have sufficient diversity for onboard optimization. Therefore, we designed a set of operators, namely addition, deletion, insertion and expansion, which can be chosen by NSGA-II to modify an intermediate solution for better results. Fig. 7 illustrates the operations.

- *Addition*
  The Addition operator is to add operations between two adjacent operations in one specific sequence. Generally speaking, the adding point of the added operation can move from the end point to the starting point of the existed operations. Taking one specific throttle sequence, $[2,\ 0,\ -1,\ 0]$ for instance, the additional operation can be added between 0 and -1, as shown in Fig. 7. The position of the added operation can move from the end point to the starting point of the existed operations, the information of the added operation can be expressed as $[p,\ \rho_l,\ \rho_r,\ g]$, where p is the position for addition operator, ρ_l is the length ratio of the last operation which is occupied by the added operation, ρ_r is length ratio of the next operation which is occupied by the added operation, g is the throttle selection of the added operation.
- *Deletion*
  Opposite to the Addition operator, the Deletion operator is to delete one operation from one specific sequence. After deleting one operation, the adjacent operations can be extended to compensate the length where the operation is deleted. Only two variable p and ρ_l is needed to express a deletion. Here p is the position for deletion operator, and ρ_l represents the length ratio of the deleted operation occupied by the

TABLE 1
Extension of Throttle Selection Scope

Motion mode	Original range	Extended range
Passive Deceleration	[1,4]	[0,6]
Keeping constant velocity	[5,6]	[3,7]
Passive Deceleration	[7,8]	[5,8]

extension of previous operation. On the contrary, $1-\rho_l$ means the length of the deleted operation occupied by the extension of next operation.

- *Insertion*
  Similar to the Addition operator, insertion also puts one more operation. However, different from adding one operation between two adjacent operations, inserting puts an existed operation sequence with a length L at a given position. So the action of the insertion operator can be expressed as $[p,\ x_s,\ L]$, where p is the position for insertion operator, x_s is the starting point of inserted operation in the original operation where the inserting operation takes place. The end point of the inserted operation will not exceed the end point of the original operation.
- *Expansion*
  Different from the previous three operators, expansion does not change the sequence structure but magnify the scope of throttle levels as exemplified in Table 1. The motion mode indicates the motion status that the train runs at. The expansion process may violates certain constrains.

The above four modification operators can further be arbitrarily grouped into compound modifications. Experimental results proved that the introduction of these operators effectively improves the solution quality.

5.3 Machine Learning Based Parameter Decision Tree

As explained in the previous two sections, we first extract frequent sequence patterns from the behaviors of experienced drivers and deploy a two-layer optimization engine to refine the pattern. The results are a set of throttle control sequences for various load conditions and geographical features. The number of such sequences can be huge and we need an efficient way to extract an optimized solution for online processing. In other words, we still need a fast solution to organize the discovered knowledge. We used machine learning techniques to organize the optimization results so that a fast online lookup can identify a proper solution. Machine learning algorithms work by building a model with a set of inputs and training the parameters of the model. Then the model can be used to make predictions or decisions according to the values of the inputs [28], [29]. In this work, we adopted the well-known BP neural network [30] as the machine learning engine since the algorithm is good at processing supervised learning as well as collaborating with optimizations.

In this work, we propose to use the parameter decision tree as the knowledge "container". The tree is first constructed by assigning one layer to a single condition (i.e., an evaluation of an input parameter). More nodes then take into branches if the evaluation leads to two or more possibilities.

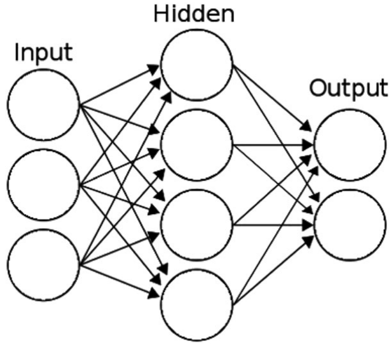

Fig. 8. The structure of the one hidden layer neural network.

Algorithm 2. Back Propagation Algorithm for One Hidden Layer Network

Input: Locomotive parameters, route parameters and the optimized trip profiles
Output: Operations and control throttle sequences
Initialize network weights (often small random values);
while *true* **do**
  for each *training example* ex **do**
    $prediction$ = neural_net_output($network$, ex);
    $actual$= teacher_output(ex);
    compute error ($prediction$ - $actual$) at the output units;
    compute Δw_h for all weights from hidden layer to output layer;
    compute Δw_i for all weights from input layer to hidden layer;
    update network weights;
  if *all train trip pattern examples classified correctly or the network output precision satisfied* **then**
    break;
return *the network*

On such a decision tree, we superimpose a BP neural network to each node. General neural network model like the one hidden layer neural network can be adopted in applications, whose structure is showed as in Fig. 8. Algorithm 2 further shows the pseudocode of the algorithm. It can be seen that there may be more than one inputs/outputs variables of the model. In this application, the input variables of the BP neural network include the locomotive traction and braking characteristics, the geographical profile of the next section, the current speed, the speed constraint, and the joint velocity of adjacent sections, etc. as shown in Fig. 9. As in Fig. 10, the output variables include control throttle allocation with, possibly, throttle selections and their proportions for each sequence, etc. We use the previously derived optimized patterns of throttle sequences as the training dataset for the BP neural network. Since each node in the parameter decision tree implies a series of evaluations, the BP network for this node can be trained with the datasets satisfying the same conditions. After training, the BP network on a node will be able to output an optimized throttle sequence under both the satisfied conditions and all other "free" variables. The introduction of machine learning

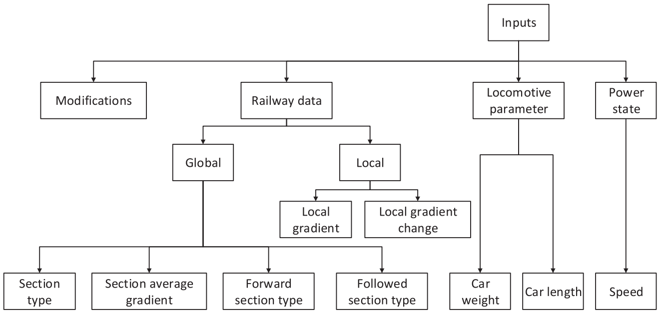

Fig. 9. The system inputs for machine learning.

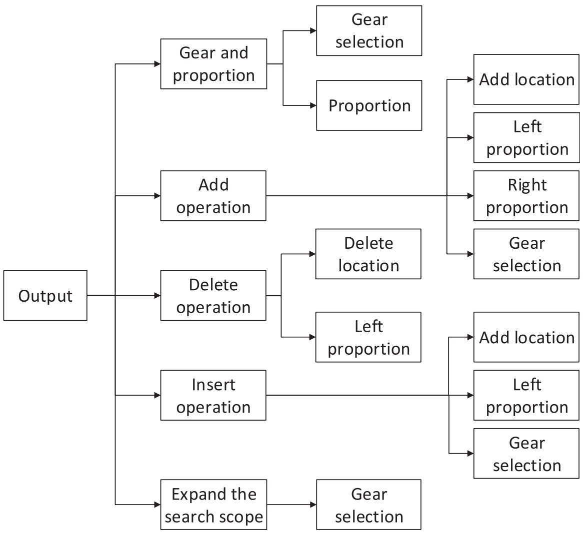

Fig. 10. The neural network output.

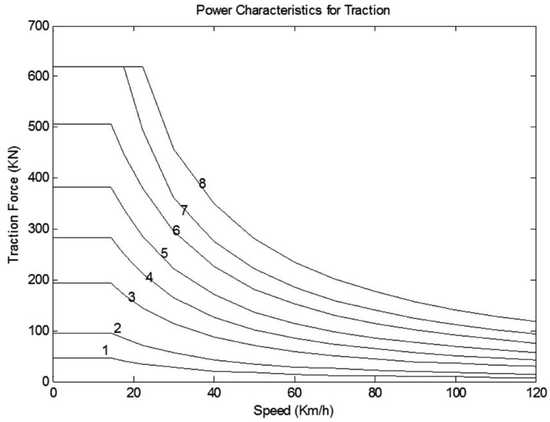

Fig. 11. Power characteristics of the selected locomotive for traction.

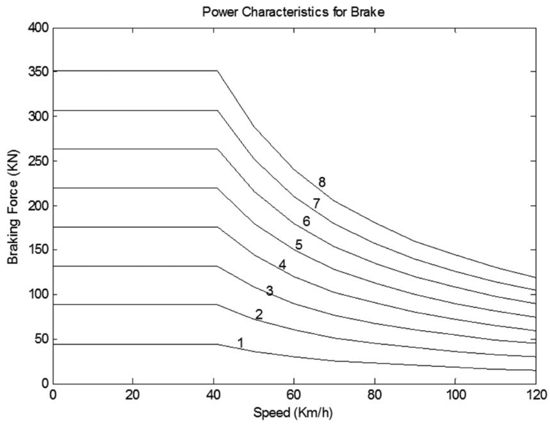

Fig. 12. Power characteristics of the selected locomotive for brake.

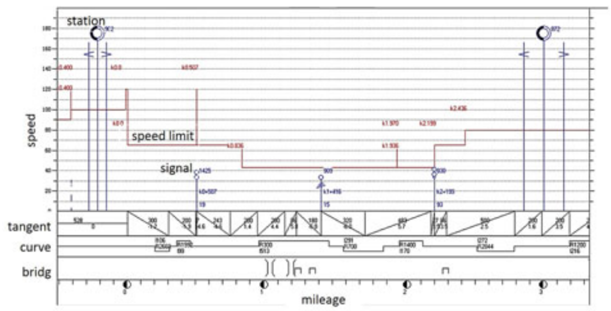

Fig. 13. Illustration of part of the railway route using in the experiments.

techniques to the decision tree is essential for this work, since the learned patterns can never cover the huge space of potential train and line conditions. The BP neural networks enables the online module to handle unexpected variations of various conditions.

To achieve superior solution quality, the offline processing is time-consuming. The complexity of the Apriori-all algorithm used in our sequence mining is $O(P * D)$, where P is total length of the potential frequent itemsets, and D is the size of the database under mining. The two-layer NSGA-II control sequence optimization algorithm has a complexity of $O(G^2H^2)$, which is the most expensive in our framework. Here, G is the number of generations and H is the number of individuals in each generation. The BP neural network based machine learning process has to be performed for each node. Its complexity at a given node is related to the number of parameters of the target parameter decision tree.

6 Case Study and Analysis

In this paper, the experiments and simulations are carried out based on a specific model of diesel locomotive. The throttle has 17 levels, with eight traction levels (1 to 8), a neutral level (0), and eight braking levels (-1 to -8). A higher absolute value suggests a stronger traction or braking force, which can be translated into a higher level of energy consumption. Since the locomotive maintains a constant power output, the energy consumption can be considered as only related to the selection of throttle. For the locomotive that used in the experiments, the power characteristics for traction and braking is shown in Figs. 11 and 12, respectively. We used the data of a commercial railway line between Mudanjiang and Harbin, in the Heilongjiang Province of China. Fig. 13 illustrates the complex geographical features of the railway line. For this case study, it takes about two weeks to mine the driver data and construct the optimized parameter decision tree for onboard processing.

We applied the proposed solution to the process. Experiments were carried out on the Hardware-in-Loop test platform shown in Fig. 14. The platform is constructed with the onboard trip optimization&control hardware device developed in this work (⑧ in Fig. 14), and other devices for a real

Fig. 14. The photo of the Hardware-in-Loop test platform of the system with: ①-Working condition generator, ②-LKJ2000 train running monitor and record device, ③-Displayer of LKJ2000, ④-Supplementary communication device from LKJ2000, ⑤-Master controller, ⑥-Throttle signal converting device, ⑦-Power supplier, ⑧-Onboard trip optimization&control device, ⑨-Train motion simulation platform.

freight train operated in China. The Hardware-in-Loop test platform serve as simulation and measurement platform of a freight train to test the performances of the onboard trip optimization&control device. While in automatic driving mode, the train running monitor and record related devices (②-④ in Fig. 14) provide the driving state to the onboard trip optimization&control device, who will generate the optimized throttle sequence to drive the simulated train model in the train motion simulation platform (⑨ in Fig. 14). The outputs of the train model then serve as the feedbacks to the train running monitor and the optimization&control devices. Evaluations on the consumption of time and energy can be done by the train motion model in the simulation platform. Due to the constant power characteristics of the locomotive at each throttle level, it is possible to calculate the energy consumption based on the rate and the running time of each throttle. The applied realistic evaluation method and the carefully learned parameters from actual driving data for the train motion simulation model guarantees the accuracy of the evaluation. Experiments showed that the evaluation error is less then 0.5 percent for most valid trips, and approaching 0 with the increase of the data. The loads inputs and the trip constraints in the simulations are given according to different actual driving data.

We choose 20 groups of actual driving data from experienced drivers with different loads conditions for the round trip along the route. The data is rather disciplinary as the trains are full-loaded when go to Harbin, and basically unloaded on the trip back. We then compare the average actual driving data by experienced drivers and simulation results by employing the proposed approach in terms of the energy efficiency. Results from experiments and simulations are shown in Table 2. It can be seen that the average energy consumption from the proposed approach is about 9.84 percent lower than the actual driving data counting both full-loaded and unloaded trips, while the average time deviation from the train schedule is merely several seconds (3 seconds as in the table). Experiments showed the capability to different running conditions of the proposed algorithm. With respect to the computation time, the proposed approach can successfully be embedded in the onboard system with limiting calculation capability for real-time control. Experiments on the Hardware-in-Loop test platform with Cortex-A8 processor of maximum 600 MHz shows the computation time of the optimization is less than 20 seconds for a pre-departure trip planning, and less than 1 second for optimizing trip for temporary speed limit. The planned trip profile can then guarantee the safety as long as the computation starts several seconds earlier than reaching the temporary speed limit.

Besides, it is interesting if one takes look into a single trip. the comparisons between optimization results based on parameter decision tree and the actual driving data on the sections along a specific trip are shown in Fig. 15, where the vertical axis stands for the performance ratios of the

TABLE 2
Comparisons on the Trip Time and Energy Consumptions between Experienced Human Drivers and the Proposed Approach

No.	loads(tons)	Human Driver - Energy Consumption(kg)	Human Driver - Trip Time(s)	Proposed - Energy Consumption(kg)	Proposed - Trip Time(s)	Trip Time Deviation(s)	Energy Saving(kg)	Energy Saving by Percentage
1	5,051	1,464.60	11,051	1,361.66	11,434	−383	102.94	7.03%
2	5,050	1,640.66	11,408	1,526.90	11,296	112	113.76	6.93%
3	5,045	1,609.07	11,140	1,487.73	11,315	−175	121.34	7.54%
4	5,028	1,537.78	11,579	1,438.36	11,328	251	99.42	6.47%
5	5,022	1,494.83	11,264	1,376.11	11,356	−92	118.72	7.94%
6	5,006	1,543.81	11,420	1,429.66	11,302	118	114.15	7.39%
7	5,005	1,453.61	11,135	1,328.08	11,414	−279	125.53	8.64%
8	5,004	1,642.12	11,244	1,558.59	11,400	−156	83.53	5.09%
9	5,002	1,504.33	11,216	1,389.21	11,337	−121	115.12	7.65%
10	5,001	1,564.47	12,012	1,440.61	11,450	562	123.86	7.92%
11	1,527	664.70	10,979	544.99	11,149	−170	119.71	18.01%
12	1,506	689.35	10,644	564.95	10,790	−146	124.40	18.05%
13	1,483	639.45	10,664	543.53	10,773	−109	95.92	15.00%
14	1,480	628.07	10,535	538.86	10,419	116	89.21	14.20%
15	1,473	665.90	10,540	556.01	10,382	158	109.89	16.50%
16	1,430	647.32	11,212	523.22	10,936	276	124.10	19.17%
17	1,405	654.15	10,808	547.02	10,659	149	107.13	16.38%
18	1,397	619.15	10,616	526.28	10,601	15	92.87	15.00%
19	1,396	604.11	10,451	515.49	10,436	15	88.62	14.67%
20	1,365	620.54	10,820	537.46	10,894	−74	83.08	13.39%
Avg	3,233.80	1,094.40	11,036.90	986.74	11,033.55	3.35	107.67	9.84%

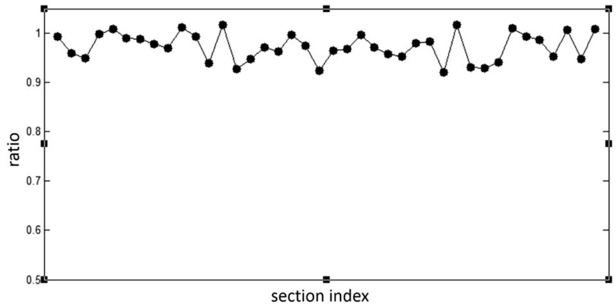

Fig. 15. Comparisons between the proposed approach and the actual driving data in ratios on the sections of along a single trip.

energy consumptions by the proposed approach over the one by the actual driving data, while the horizontal axis stands for section index under comparison. It shows that although the overall evaluation on the energy consumption performance by the proposed approach is better than the average one by human drivers, it fails the superiority on some of the sections. This is mainly because the optimization is focusing on the global performance which may adjust time distribution between different sections.

7 CONCLUSION

This paper presents a energy-efficient train control framework for smart railway transportation. By integrating online and off-line processing techniques, the propsoed framework generates throttle sequences that lead to energy saving under the constraints of trip time and computation time. This work leverages the fast-growing machine learning techniques to extract the optimized driving behaviors of human drivers and encode the learned knowledge into a parameter decision tree for fast online optimization. A case study on a given locomotive proved the effectiveness of the proposed framework and an energy saving of 9.84 percent on different running conditions can be achieved. The paradigm and techniques developed in this work are of significant potential for other smart city system applications.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China (No. 11302074) and the National Key Technology Support Program (No.2013BAG24B03).

REFERENCES

[1] B. Bin, K. Cao and X. Cai, "Framework of security for the railway intelligent transportation system," *Bridges*, vol. 10, pp. 700–705, 2013.

[2] D. Guck, J. P. Katoen, M. I. A. Stoelinga, T. Luiten and J. Romijn, "Smart railroad maintenance engineering with stochastic model checking," in *Proc. Second Int. Conf. Railway Technol.: Res.*, Development and Maintenance, Railways, Ajaccio, Corsica, France, Apr. 2014, vol. 299.

[3] China Railway Corporation, *Annual Railway Statistics Report*, China Railway Corporation, Beijing, China, 2013.

[4] S. H. Han, Y. S. Byen, J. H. Baek, T. K. An, S. G. Lee and H. J. Park, "An optimal automatic train operation (ATO) control using genetic algorithms (GA)," in *Proc. IEEE Region 10 Conf.*, Aug. 1999, vol. 1, pp. 360–362.

[5] Y. S. Li and Z. S. Hou, "Study on energy-saving control for train based on genetic algorithm," *J. Syst. Simulation*, vol. 19, no. 2, pp. 384–387, 2007.

[6] Y. Wang, B. De Schutter, T. J. Van den Boom and B. Ning, "Optimal trajectory planning for trains—A pseudospectral method and a mixed integer linear programming approach," *Transp. Res. Part C: Emerging Tech.*, vol. 29, pp. 97–114, 2013.

[7] A. K. Kumar, W. Daum, G. R. Shaffer, P. K. Houpt, J. F. Noffsinger and D. L. McKay, "General Electric Company, Trip optimizer method, system and computer software code for operating a railroad train to minimize wheel and track wear," U.S. Patent 9,233,696, 2016.

[8] A. Caprara, L. Kroon and P. Toth, "Optimization problems in passenger railway systems," in *Wiley Encyclopedia of Operations Research and Management Science*. Hoboken, NJ, USA: Wiley, 2011.

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

[10] R. Chevrier, "An evolutionary multi-objective approach for speed tuning optimization with energy saving in railway management," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 279–284.

[11] S. Su, X. Li, T. Tang and Z. Gao, "A subway train timetable optimization approach based on energy-efficient operation strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 883–893, Jun. 2013.

[12] F. R. Salmasi, "Control strategies for hybrid electric vehicles: Evolution, classification, comparison, and future trends," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 2393-2404, Sep. 2007.

[13] Y. Ding, "Study on train movement calculation and operation simulation system," Ph.D. dissertation, Beijing Jiaotong Univ., Beijing, China, 2004.

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

[15] D. Wang, X. Lin and Y. Zhang, "Fuzzy logic control for a parallel hybrid hydraulic excavator using genetic algorithm," *Autom. Construction*, vol. 20, no. 5, pp. 581-587, 2011.

[16] S. Gao, H. Dong, Y. Chen, B. Ning, G. Chen and Q. Liang, "Neuro-adaptive and robust automatic train control subject to unknown dynamics," *Proc. 32nd Chinese Control Conf.*, 2013, pp. 8130–8134.

[17] S. Ghosh, T. Lee and T.S. Lee, *Intelligent Transportation Systems: New Principles and Architectures*. Boca Raton, FL, USA: CRC Press, 2002.

[18] Y. Yang, Y. Zhu, Q. Dai and T. R. Li, "Smart rail transportation—an implementation of deeper intelligence," *J. Comput. Appl.*, vol. 32, no. 5, pp. 1205–1207+1216, 2012.

[19] P. Li, L. M. Jia and A. X. Nie., "Study on railway intelligent transportation system architecture," in *Proc. Intell. Transp. Syst.*, 2003, vol. 2, pp. 1478–1481.

[20] V. Xuan, "Analysis of necessary conditions for the optimal control of a train," Ph.D. dissertation, University of South Australia, St Adelaide, SA, Australia, 2006.

[21] B. Mao, *The Calculation and Design of Train Operation*. Beijing, China: People Transport Press, 2008.

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

[23] T. M. Mitchell, *Machine Learning*. Singapore: McGraw-Hill, 1997.

[24] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *Data Structures and Algorithms*. Reading, MA, USA: Addison-Wesley, 1983.

[25] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. 11th Int. Conf. Data Eng.*, 1995, pp. 3–14.

[26] K. Miettinen, *Nonlinear Multiobjective Optimization*. New York, NY, USA: Springer, 2012.

[27] C. L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making, Methods and Applications: A State-of-the-Art Survey*. New York, NY, USA: Springer-Verlag, 29 May 2012.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[29] M. Mohri, A. Rostamizadeh, A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.

[30] D. W. Patterson, *Artificial Neural Networks: Theory and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.

Jin Huang received the BE and PhD degrees from the College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China, in 2012 and 2006, respectively. He is also received a joint PhD degree from the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, during 2009-2011. He has been the postdoc and asistant researcher at the School of Software, Tsinghua University, Beijing, China, since 2013. His research interests include artificial intelligence in Intelligent Transportation Systems, Cyber-Physical systems, automation and mechatronics, fuzzy engineering, etc.

Yangdong Deng (M-'06) received the BE and ME degrees from the Electrical and Electronics Department, Tsinghua University, Beijing, China, in 1998 and 1995, respectively, and the PhD degree in electrical and computer engineering from the Carnegie Mellon University, Pittsburgh, PA, in 2006. He is currently an associate professor at the School of Software, Tsinghua University, Beijing, China. His current research interests include parallel electronic design automation algorithms, parallel program optimization, and general purpose computing on graphics processing hardware, etc. He is a member of the IEEE.

Qinwen Yang received the BE and ME degrees from the College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China, in 2009 and 2006, respectively, and the PhD degree from the Department of Mechanical and Manufacturing Engineering, University of Calgary, AB, Canada, in 2014. She has been an assistant professor at the College of Mechanical and Vehicle Engineering, Hunan University, since 2014. Her current research interests include sequential sampling, global optimization, and multi-level collaborative modeling, etc.

Jiaguang Sun received the BS degree in automation science from the Tsinghua University, in 1970. He is a professor, member at the Chinese Academy of Engineering, dean at the School of Information Science & Technology (IST), Tsinghua University, director at the Ministry of Education Key Laboratory of Information Security, director at the National Engineering Research Center of CAD Supporting Software, director at the Tsinghua National Laboratory for Information Science & Technology(IST), and president at the Executive Council of China Engineering Graphics Society. His research areas include computer graphics, computer-aided design, software engineering, and system architecture, etc.