APPLYING MONTE CARLO TREE SEARCH TO THE OPTIMIZATION OF TRAIN SPEED CURVE FOR ENERGY-SAVING IN A MASS RAPID TRANSIT SYSTEM

[1]BWO-REN KE, [2]CHENG-WEI CHOU, [3]GUAN-WEI HUANG

[1]Department of Electrical Engineering, National Penghu University of Science and Technology, Makung, Penghu, 880 Taiwan R.O.C.,
[2] National Chung-Shan Institute of Science and Technology, Longtan, Taoyuan, 325 Taiwan R.O.C.,
[3] Department of Electrical Engineering, National Penghu University of Science and Technology, Makung, Penghu, 880 Taiwan R.O.C.
E-mail: [1]brke@npu.edu.tw, [2]kapakapa@gmail.com, [3]ship73212@gmail.com

Abstract - This study applied combinatorial optimization techniques to Monte Carlo tree search (MCTS) and incorporated the calculation of the optimal speed curve for energy-saving mass rapid train operation. The goal was to achieve online speed command planning based on this derived curve when the trains operate between successive stations. MCTS, an algorithm originally designed for two-player games, was subsequently adjusted and transformed into an adequate optimization method. In addition, energy-saving factors, coasting and regenerative braking were considered in the experiments, the results of which confirmed that MCTS can indeed solve the problem of energy-saving train operation.

Index Terms - MCTS, MRTS, Train Speed, Energy-Saving

I. INTRODUCTION

With increasing urban scale and population density, developing public transportation systems has become a necessary trend. Such systems can improve the convenience of traveling as well as reduce traffic congestion and compensate for the lack of parking space in urban areas. Many countries have planned relevant policies for promoting public transportation systems; areas with vast land and a low population density tend to develop air transportation systems, whereas densely populated areas mostly develop rail transportation systems. Taiwan has a dense population due to its geographic features; therefore, special attention must be paid to developing rail transportation systems.

Rail transportation is an effective tool for solving traffic congestion, increasing transportation capacity, alleviating air pollution, and reducing energy consumption. It plays a crucial role among various public transportation systems. Rail systems have evolved from intercity railways and urban rapid mass transit to high speed rail between metropolitan areas. Advanced rail transportation systems have become a symbol of advanced countries. Rapid mass transit systems are mainly developed in urban areas and have the advantages of comfort, safety, zero exhaust emissions, high transportation capacity, fast speed, and convenience. Furthermore, operating them on an exclusive right-of-way helps reduce surface road traffic.

Presently, most electric railway trains are equipped with regenerative braking capability, an energy recovery mechanism that enables a traction motor to be used as a power generator that feeds the energy generated by the train back into the DC traction power grid as electricity. This recycling of renewable energy achieves the goal of improving electrical energy efficiency. By employing the automatic operation system of a high-capacity mass transit railway train as the research framework, this study applied combinatorial optimization techniques to Monte Carlo tree search (MCTS) when running a control algorithm for energy-saving railway operation. In addition, the calculation of the optimum speed curve for energy-saving operation was incorporated into the search method to improve power utilization efficiency.

II. METHODOLOGY

MCTS is often used to solve problems in the field of computer games [1, 2], where the core concept is to use it as a method for evaluating the current status. Because it utilizes tree search, it is particularly suitable for solving combinatorial optimization problems. The following passage elaborates on how MCTS was employed to solve the energy-saving operation problem proposed in the present study.

In a search tree of MCTS, each node represents a state, and a branch of a node represents the action that can be implemented under this state. To map the energy-saving operation problem to the tree search framework, the problem must first be discretized to determine the states and actions to be selected. In general, the distance between two stops can be divided into several sections with equal length. For each section, a change in train speed represents an action (i.e., speed command). Thus, the state of each node signifies the current section and speed of the train, and the branch of the node is the speed command

corresponding to its current status.

The concept can be used to map the energy-saving operation problem to the tree search framework. The procedures of MCTS are explained as follows:

A. Simulation

MCTS is a method that derives an approximate value from a large amount of simulation results. In the field of computer games, a simulation refers to obtaining a victory or defeat that involves using computer calculations and chessboard simulations to finish a game with a designated method. In the proposed energy-saving operation problem for railway trains, a simulation refers to the process in which a simulated train continually moves to the next section from its current one by selecting speed commands until it runs through all sections and obtains speed and energy consumption data.

In this study, speed commands were randomly selected during the simulation. However, if random simulations were implemented without any constraints, numerous invalid simulations would occur because they could not meet the railway train's arrival time. Such invalid simulations do not generate useful data and can substantially increase the time required for optimization. Therefore, the speed command data in all sections were preprocessed to exclude the possibility of invalid simulation outcomes. An invalid simulation outcome means that under a certain state, implementing a speed command would cause the railway train to arrive too early (late) at the destination even if the speed at all subsequent sections is switched to the lowest (highest) level. Such speed command was considered unreasonable in this study. After preprocessing, the speed and accuracy of the algorithm increased.

B. Selection

In MCTS, the growth direction of the search tree was selected before the simulation. The upper confidence bound (UCB) algorithm is typically used to determine the growth direction of the search tree; it has achieved great success in the field of Go [1]. In the present study, the UCB algorithm was adopted to determine the search direction.

In the field of computer games, the term "past performance" generally refers to the odds ratio of a simulated chess game; this ratio ranges from 0 to 1. In the proposed energy-saving problem, "energy consumption" was the criterion by which the simulation result was judged. However, the level of energy consumption differs greatly between railway trains. Thus, a normalization method was required to achieve data consistency, allowing the simulation method to be applicable to various train stations and reduce the need to greatly adjust parameters to suit each one. In the present study, multiple methods of energy consumption normalization were adopted. The first involved using a formula, (1−average energy consumption)/M, where M is set to 500 to ensure that the normalized outcome would approximate 0.5. However, the problem of "case relevance" remained. The second method involved using a UCB algorithm formula, where the adjusted value for balancing exploitation and exploration was multiplied by the current level of average energy consumption. This method can avoid "case-related" problems.

MCTS was originally designed for two-player games. Thus, applying it to a "single-player game" such as train operation for energy-saving inevitably generates a problem: For branches frequently selected during early stages, they tend to have lower average energy consumption levels than less frequently selected branches do because of higher search depth. As a result, frequently selected branches remained easily selected whereas scarcely selected branches stayed unsearched, generating a vicious cycle. By contrast, a search tree used for two-player games is counterbalanced and produces fewer of the aforementioned problems because it considers the goals of both players (each selecting a search direction to their advantage in respective tree levels). This counterbalancing effect is absent in single-player games, which easily results in a highly unequal distribution of resources. A temporary solution is to adjust the parameters used in the formula to lean toward exploration during the searching process. However, this can reduce the search depth and transform the usual depth-first search into a breadth-first search. Once the number of simulations exceeds a certain limit, the same problems will reoccur.

To eliminate this drawback, this study referred to a relevant study on the application of single-player MCTS [3], which adopted the following formula for branch selection:

$$UCB_i = X_i + bias\sqrt{\frac{\ln N}{T_i}} + \sqrt{\frac{\sum r^2 - T_i \times X_i^2 + D}{T_i}} \quad (1)$$

The first half of the formula is identical to that of the UCB algorithm. In the last term of this equation, $\sum r^2$ denotes the sum of the square of the past performance of all branches emanating from a node (including the performance of the node itself); T_i represents the number of visits to node i; X_i denotes past performance; and D represents an extremely large constant.

The core concept of Equation (1) lies in the fact that nodes with a higher number of visits tend to have superior performance. If the performance of such nodes meets expectations, they possess the value of being continuously selected; otherwise, they should be replaced by other nodes. The experiment results revealed that the middle term of the equation did not exert a large effect on the overall performance, and that the effect of D-value was nonsignificant. Thus, the

equation was adjusted as follows and adopted in subsequent simulations:

$$UCB_i = X_i + bias\sqrt{\frac{\sum r^2 - T_i X_i^2}{T_i}} \quad (2)$$

C. Backpropagation

In this procedure, data concerning the energy consumption required in each simulation was returned; thus, the average energy consumption of and number of visits to all nodes in the best sequence (including root and leaf nodes) were updated in the following selection stages.

D. Expansion

Upon discovery of any leaf node that had been visited but had no children yet, all legal branches were expanded.

In a given case, MCTS was used to run a designated number of simulations, after which the branch with the highest number of visits was selected from the first level of the search tree and used as the result that represented section 1. Subsequently, the steps performed for section 1 were repeated on the remaining sections until they all generated adequate results. During this process, the optimal simulation result was retained and used as the final result upon completing all steps.

III. SIMULATION RESULTS

The following paragraphs present and discuss the experimental results of applying MCTS to solve the optimization problem of the speed curve of energy-saving railway operation. The simulation data obtained under various coasting and regenerative braking modes are presented in diagrams. Furthermore, the various selection methods are compared. Table 1 lists all parameters related to the route type and operation of the train.

Table 1. Operating, speed limit, and gradient parameters in the studied case

Operating parameters		
Station-to-station distance (m)		3057.7
Average speed (km/h)		46
Operating time between stations (s)		239
Error rate for on-time arrival (%)		5
Speed limit		
Starting mileage (m)	Ending mileage (m)	Speed limit (km/h)
0	3057.7	80
Gradient		
Starting mileage (m)	Ending mileage (m)	Gradient (%)
0	78	0
78	298	-0.5
298	678	-2.8
678	1037	-2.293
1037	2420	0.3
2420	2898	2.999
2898	3057.7	0

The following experiments were conducted under various simulation numbers, ranging from 1,000 to 100,000 ($bias$: 0–1.25). Each set of parameters underwent 50 experiments to derive the average and optimum values of energy consumption. The selection procedure of the MCTS was performed using Equation (2). Fig. 1 displays the results of optimization when no coasting mode was adopted. In this figure, "avg" represents the average level of energy consumption and "best" denotes the minimum level of energy consumption during the simulation. The results of the experiment indicated several trends. First, with a higher number of simulations, either the average or the optimum values improved significantly. Second, the level of energy consumption was the lowest at a $bias$ value of approximately 0.75, suggesting that both excessively large and small $bias$ values led to problems. With an excessively large $bias$, the tree-based search became too sparse and was conducted at a lower search depth; thus, the results of numerous insignificant branches were mixed with those of the crucial ones. Moreover, with an excessively small $bias$, the search became overly concentrated and generated unsatisfying results because it did not consider any other branches.

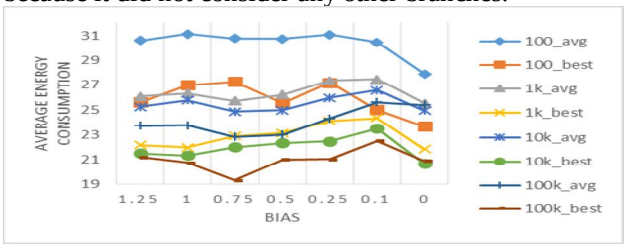


Fig. 1 Minimum energy consumption derived from various numbers of simulation and $bias$ values without adopting the coasting mode

Fig. 2 demonstrates the results of energy-saving optimization under the coasting mode. The parameters remained the same as in the preceding experiment. The coasting mode significantly reduced the overall energy consumption. In addition, a higher number of simulations resulted in better performance in energy consumption. Either excessively large or small $bias$ values did not increase the energy consumption performance.

Under the regenerative braking mode, the energy recovery rate was set to 0.3. Figs. 3 and 4 illustrate the results of energy-saving optimization under the coasting mode and without, respectively. The experiment results indicated that energy consumption decreased when the number of simulations increased. Although using regenerative braking enhanced the train's performance of energy consumption, compared to not using regenerative braking, it attained a slower rate of improvement in terms of optimum value reduction throughout increased simulations.

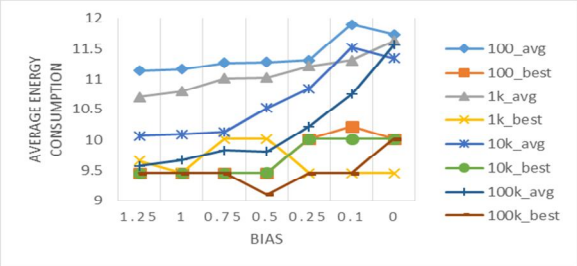


Fig. 2 Minimum energy consumption derived from various numbers of simulation and $bias$ values under the coasting mode

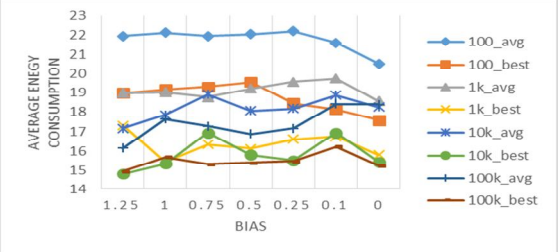


Fig. 3 Minimum energy consumption derived from various numbers of simulation and $bias$ values under regenerative braking without adopting the coasting mode

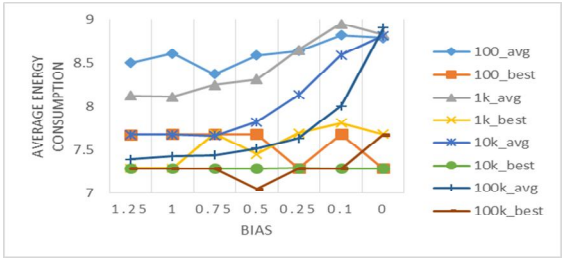


Fig. 4 Minimum energy consumption derived from various numbers of simulation and $bias$ values under regenerative braking and the coasting mode

CONCLUSION

In this study, a discretization approach was applied to map the proposed problem of energy-saving railway operation to a tree search framework. MCTS, an algorithm that has been confirmed in the field of computer games as suitable for solving highly ambiguous problems, was adopted to solve the proposed problem. This study adjusted and transformed this algorithm, which was originally designed for two-player games, into an adequate optimization method. In addition, the two energy-saving approaches of coasting and regenerative braking were incorporated into the simulation. Overall, the experiment results confirmed that MCTS can indeed solve the problem of energy-saving railway operation.

ACKNOWLEDGEMENTS

This research was sponsored by the Ministry of Science and Technology, Taiwan, under Grants MOST 103-2221-E-346 -005 and MOST 106-2221-E-346-006.

REFERENCES

[1] L. Kocsis and C. Szepesvari, "Bandit Based Monte-Carlo Planning," European Conference on Machine Learning, pp. 282-293, 2006.

[2] B. B. Browne, et al., "A Survey of Monte Carlo Tree Search Methods," IEEE Transactions on Computational Intelligence and AI in Games, Vol. 4, No. 1, pp. 1-43, 2012.

[3] M. P. D. Maarten, et al. "Single-Player Monte-Carlo Tree Search for SameGame," Knowledge-Based Systems, Vol. 34, pp. 3-11, 2012.

★★★