A Novel Dynamic Programming Approach for Optimizing Driving Strategy of Subway Trains

Hao Gao[1], Yadong Zhang[1,a] and Jin Guo[1]

[1] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China*

Abstract. The reduction of operation energy consumption without decreasing service quality has become a great challenge in subways daily operation. A novel DP based approach is proposed for optimizing the train driving strategy. The optimal driving problem is first considered as a multi-objective problem with five optimal targets (i.e., energy saving, punctual arriving, less switching, safe driving and accurate stopping). The optimization problem is remodelled as a multistage decision problem by discretizing the continuous train movement in space. The process of dynamic programming is carried out in the velocity-space status space. Due to the discretizing rules of searching space, the optimal goals of safe driving and accurate stopping can be satisfied during the searching process. The rest of multiple goals are spilt into cost functions and constrains for each stage. Due to the multiple cost functions, a set of pareto optimal solutions can be achieved at each vertex during the process of dynamic programming. To further improve the efficiency of algorithm, two evaluation criterions are introduced to maintain the capacity of the pareto set at each vertex. A case study of Yizhuang urban rail line in Beijing is conducted to verify the effectiveness and the efficiency of DP based algorithms.

1 Introduction

As a fast and green transportation, urban rail transit can efficiently reduce congestion and pollution in our daily life. The fast growth of urban rail transit has brought huge energy consumption in recent years. According to the incomplete statistics, the total energy consumption of urban rail transit in China was 15 billion kilowatt-hours in 2018. The traction energy consumption accounted for 45.3 percent of the total energy consumption. Therefore, reducing traction energy consumption has become a hot research topic recently.

The traction energy consumption can be efficiently reduced by optimizing the train driving strategy. The fundamental work of driving optimization was established a few decades ago. Most of those work is based on optimal control theory, and on Pontryagin Maximum Principle particularly. The derivation shows that the optimum driving strategy should contain four regimes (i.e., maximum traction, cruising, coasting, and maximum braking) [1, 2, 3, 4]. Based on the derivation, some researchers presented heuristic algorithms to solve the optimal driving problem by determining the position of multiple coasting switching points [5, 6]. Due to the randomness of searching process, the convergence speed of heuristic algorithms is slow. Furthermore, the global optimum solution is not always guaranteed to be found. To obtain the global optimum solution, some researchers presented exact algorithm to solve the problem by considering the target problem as a multistage decision problem [7, 8, 9, 10]. The multistage optimization is carried out in a space-time-velocity three-dimensional searching space. However, the efficiency of the exact algorithms is limited by the 'curse of dimensionality'.

To overcome the shortage of heuristic algorithms and traditional exact algorithm. A dynamic programming (DP) based exact algorithm is proposed to obtain the global optimum solution of the multi-objective problem in an efficient way. Furthermore, a more realistic train driving model is considered with multiple optimal goals (i.e., energy saving, punctual arriving, less switching, safe driving and accurate stopping).

2 Problem definition

The optimal driving problem can be described as follows. For a given running time, there exists enormous train driving strategies which can ensure train arriving punctually, accurately and safely. Among those feasible driving strategies, the strategy which costs minimum energy consumption is regarded as the optimum driving strategy. Overlooked by other researchers commonly, the frequent switching of the driving regimes may cause passengers uncomfortable and increase tear and wear of vehicle equipment. Thus, reducing the regimes changes will be regarded as another optimal object in this paper.

For simplicity, we assume that the train equip with Automatic Train Control (ATO) system to realize the cruising control at any line position. We also assume that the train load is a constant value at any time. The

[a] Corresponding author: ydzhang@home.swjtu.edu.cn

regenerative energy between accelerating train and braking train is not considered in this paper.

2 Model and algorithm

In our approach, the space range [0, X] of interstation is spilt into K stages. Note, the speed restriction and track slope of each stage is constant. At kth stage, the train takes unique regime $u_k \in$ {maximum traction, cruising, coasting, maximum braking} as the driving mode. All the K regimes constitute the driving strategy U, $U = \{(u_1, x_1), (u_2, x_2), \ldots, (u_K, x_K)\}$, where x_k is the travelling distance at kth stage. The speed range [0, $V_{\max}(x_k)$] at position x_k is discretized into a set of vertices $s_{k,i} = (x_k, v_i)$ by velocity interval Δv. The $V_{\max}(x_k)$ is the maximum train velocity at position x_k under the flat-out driving strategy. Figure 1 shows an example of the discretization, where the solid line is the flat-out train trajectory and the hollow dots are the vertices. The vertex transition equation $s_{k+1,j} = F(s_{k,i}, u_k)$ at kth stage can be described by train dynamic equations in details which based on single-particle model:

$$M \cdot (1+\alpha) \cdot v \cdot dv = \left[p(u_k) - q(u_k) - r \right] \cdot dx \quad (1)$$

$$r = M\left(a + bv + cv^2\right) + Mg\sin(\theta) \quad (2)$$

where x is the position of the train, v is the velocity of the train, M is the mass of the train, α is a factor to consider the rotating mass, $p(u_k)$ is the train traction at regime u_k, $q(u_k)$ is the braking force at regime u_k, r is the running resistance which includes basic resistance and line resistance. The basic resistance can be formulated as Davis equation where a, b and c are train-specific constants. The line resistance is the force caused by the the track gradient θ. The g is the gravitational constant.

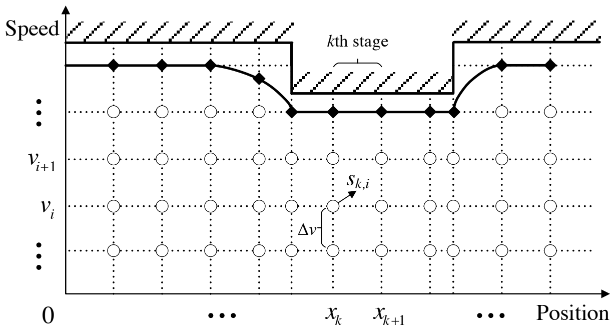

Figure 1. Discretization

The traction and braking force of train at regime u_k are described in Table 1, where the P is the maximum traction and Q is the maximu braking force, P and Q are constant values. The goal of enery saving can be fomulated as：

$$\min E(U) = \int_0^X p(u_k)\, dx \quad (3)$$

where X is the interstation distance, $E(U)$ is energy consumption of strategy U. The goal of punctual arriving can be fomulated as：

$$T(U) = \int_0^X \frac{1}{v} dx = T_{\text{set}} \quad (4)$$

where $T(U)$ is interstation running time of strategy U and T_{set} is the running time specified by timetable. The goal of less switching can be formulated as:

$$\min D(U) \quad (5)$$

where the $D(U)$ is the regime changes of strategy U. The goal of safe driving can be formulated as:

$$v < V_{\text{limit}} \quad (6)$$

where V_{limit} is the speed restriction. The goal of accurate stopping can be formulated as:

$$\sum_{k=1}^{K} x_k = X \quad (7)$$

According to the discretization rules above, the optimal goals of safe driving and accurate stopping can be satisfied during the searching process. Other three goals (i.e., energy saving, punctual arriving, less switching) should be optimized during the dynamic programming approach.

Table 1. Traction and braking force of each regime

Regime	Output force
Maximum traction	$p(u_k) = P$, $q(u_k) = 0$
Cruising	$p(u_k) = r$, $q(u_k) = 0$
Coasting	$p(u_k) = 0$, $q(u_k) = 0$
Maximum braking	$p(u_k) = 0$, $q(u_k) = Q$

The dynamic programming is executed backward from destination to origin. We define the sub-strategy $U(s_{k,i})$ as the sequence of regimes from vertex $s_{k,i}$ to destination. There are three cost funcions involved to evaluate the optimality of $U(s_{k,i})$ at kth stage :

$$E\left(U^*\left(s_{k,i}\right)\right) = \min_{u_k}\left[e\left(s_{k,i}, u_k\right) + E\left(U^*\left(F\left(s_{k,i}, u_k\right)\right)\right)\right] \quad (8)$$

$$T\left(U^*\left(s_{k,i}\right)\right) = \min_{u_k}\left[t\left(s_{k,i}, u_k\right) + T\left(U^*\left(F\left(s_{k,i}, u_k\right)\right)\right)\right] \quad (9)$$

$$D\left(U^*\left(s_{k,i}\right)\right) = \begin{cases} \min\limits_{u_k} D\left(U^*\left(F\left(s_{k,i}, u_k\right)\right)\right) + 1, & \text{if } u_k \neq u_{k+1} \\ \min\limits_{u_k} D\left(U^*\left(F\left(s_{k,i}, u_k\right)\right)\right), & \text{if } u_k = u_{k+1} \end{cases} \quad (10)$$

The $e(s_{k,i}, u_k)$ and $t(s_{k,i}, u_k)$ are the equation of the immediate cost of energy and time in kth stage at vertex $s_{k,i}$ using regime u_k.

$$e\left(s_{k,i}, u_k\right) = \int_{x=x_k}^{x_{k+1}} p\left(u_k\right) dx \quad (11)$$

$$t\left(s_{k,i},u_k\right)=\int_{x=x_k}^{x_{k+1}}\frac{1}{v}dx \tag{12}$$

To satisfy the punctual arriving goal, the driving strategy $U(s_{k,i})$ should be constrained as:

$$T\left(U\left(s_{k,i}\right)\right)\geq T_{\text{des}}\left(s_{k,i}\right) \tag{13}$$

$$T\left(U\left(s_{k,i}\right)\right)\leq T_{\text{set}}-T_{\text{org}}\left(s_{k,i}\right) \tag{14}$$

Where $T_{\text{des}}(s_{k,i})$ is the minimum running time from $s_{k,i}$ to destination, $T_{\text{org}}(s_{k,i})$ is the minimum running time from origin to $s_{k,i}$. Specifically, the constrains at destination vertex $s_{K+1,0}$ and origin vertex $s_{1,0}$ are formulated as:

$$T\left(U\left(s_{1,0}\right)\right)=T_{\text{set}} \tag{15}$$

$$T\left(U\left(s_{K+1,0}\right)\right)=0 \tag{16}$$

The calculation method of $T_{\text{des}}(s_{k,i})$ and $T_{\text{org}}(s_{k,i})$ are described in Figure 2, where the blue line is the flat-out train trajectory from origin to $s_{k,i}$, and the red line is the flat-out train trajectory from $s_{k,i}$ to destination.

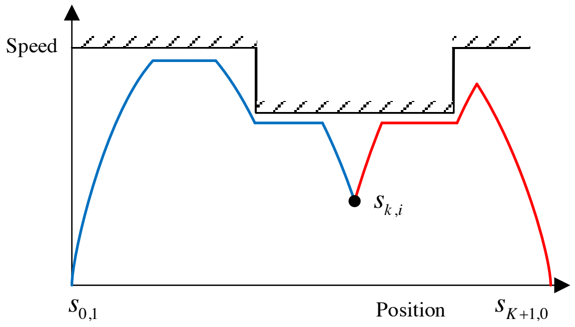

Figure 2. Flat-out time from $s_{k,i}$ to destination and origin

Due to the conflict between energy consumption and running time cost functions, each vertex $s_{k,i}$ exists a set of pareto-optimal solutions which satisfy the cost functions listed above. A strategy U^* is said to be pareto optimal if only there is no other strategy U such that $E(U)\leq E(U^*)$ and $T(U)\leq T(U^*)$ can be established at the same time. We define $U^*_w(s_{k,i})$ as the wth pareto solution in the pareto set of vertex $s_{k,i}$ and $W(s_{k,i})$ as the total capacity of pareto set at vertex $s_{k,i}$. To improve the efficiency of the algorithms, an upper limit W_{limit} for the capacity of pareto set at each vertex is introduced. The congestion degree and regimes changes of each pareto solution are taken as the evaluation criterions to maintain the capacity of the pareto set. The reserve operation will be executed only when the capacity of pareto set is bigger than the W_{limit}.

According to the less switching goal, the pareto solutions which have less regime changes will be reserved during the DP process. For the pareto solutions which have same number of regimes changes, the congestion degree is introduced to evaluate the importance of those pareto solutions. The congestion degree is defined as the searching space size of the each pareto solution for the conflict goals of energy consumption and running time, as described in Figure 3. The congestion degree of each pareto solution can be calculated as:

$$\begin{aligned}C\left(U^*_w\left(s_{k,i}\right)\right)=&\left|T\left(U^*_{w+1}\left(s_{k,i}\right)\right)-T\left(U^*_{w-1}\left(s_{k,i}\right)\right)\right|\\&+\left|E\left(U^*_{w+1}\left(s_{k,i}\right)\right)-E\left(U^*_{w-1}\left(s_{k,i}\right)\right)\right|\end{aligned} \tag{17}$$

where the $C(U^*_{\text{w}}(s_{k,i}))$ is the congestion degree of the wth pareto solution in the set of $s_{k,i}$. The pareto solutions which have bigger congestion degree will be reserved during the DP process. The introduction of the congestion degree can improve the efficiency of DP process and retain the diversity of the solutions generated at each stage.

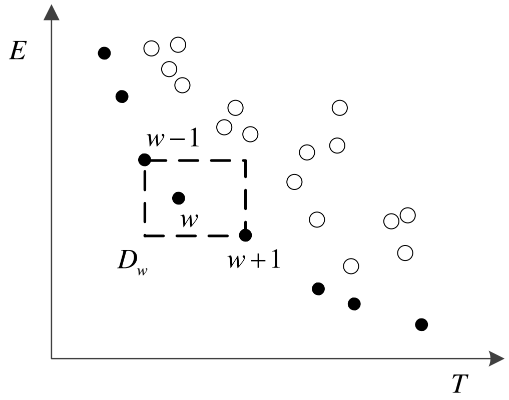

Figure 3. Schematic diagram of congestion degree

The vertex's pareto solutions which have more regimes changes and less congestion degree will be deleted only when the capacity of pareto set is bigger than the W_{limit}. Note, the evaluation criterion of switching number has the higher priority than congestion degree in this paper.

The process of DP is executed from destination to origin. The optimum solution is selected as the minimum energy consumption solution from the final vertex's pareto set. The process of the backward DP is discribed as follows :

Step1 Import the data of lines and trains. Set the model parameter (i.e., the number of stages K, the velocity interval Δv and the capacity of pareto set W_{limit}). Initialize the model, let $W(s_{K+1,0})=1$, $E(U^*_1(s_{K+1,0}))=0$, $T(U^*_1(s_{K+1,0}))=0$, $D(U^*_1(s_{K+1,0}))=1$, $k=K$.

Step2 Traversal all the vertices at kth stage. For each vertex $s_{k,i}$ and $u_k\in$ {maximum traction, cruising, coasting, maximum braking}, backtrack the last vertex though dynamic equations $F(s_{k,i}, u_k)$. Add the decision variable $u_{\underline{k}}$ to each pareto solution in the set of the last vertex $\{U^*_{\text{w}}(F(s_{k,i}, u_k))\}$ and generate new strategies. Check the feasibility of each new strategy with constrains (13 - 14). Add the feasible solution to the solution set $\{U(s_{k,})\}$ at vertex $s_{k,i}$.

Step3 According to the cost functions (8 - 9), execute the pareto dominance operation for the solution sets of all vertices at kth stage, and get the pareto solution set $\{U^*_{\text{w}}(s_{k,i})\}$ for each vertex $s_{k,i}$. If the capacity of pareto set $W(s_{k,i})>W_{\text{limit}}$, calculate the congestion degree of the solutions and rank the solutions from large to small. Based on the rank sequence of congestion degree, rank the optimal solutions by the switching number from lest to most. Remove the redundant solution until $W(s_{k,i}) = W_{\text{limit}}$. Note, the two-setp rank operation introduced above can keep the priority of regimes changes higher than congestion degree. If k -1 ≠ 0, let $k = k - 1$, and jump to **step2**, else output the optimum solution from the the final vertex's pareto set.

3 Simulation and experiment

A C++ based procedure is developed to verify the effectiveness and efficiency of the Multi-Objective Dynamic Programming (MODP) approach as mentioned above. The procedure is performed using a laptop with Intel (R) Core TM i5-2530M CPU and 4 gigabytes memory. The train parameters and line data of Beijing Yizhuang subway is selected according to the literature [11]. The searching space is discretized with the rules that stages number $K = 50$ and velocity interval $\Delta v = 1$ km/h. The upper limit capacity of pareto is set as $W_{\text{limit}} = 50$. As a comparison, a Weight-based Dynamic Programming (WDP) approach according to the literature [10] is performed with the same model setting. The WDP introduces weight factor to balance the importance of multiple objects. Table 2 shows the results.

There are 13 interstation listed with distance, trip time and flat-out energy consumption in our cases. Arriving deviation, energy-saving comparing to flat-out, switching number and calculate time are taken as the evaluating criterions of two optimization approaches. As the results showed in table 2, MODP obtains the better solutions using less calculation time. In average, the arriving deviation, energy saving and regimes changes of WDP is 3.5 second, 52.8% and 11.5 times. The arriving deviation, energy saving and regimes changes of MODP is 0.7 second, 54.5% and 7 times. Furthermore, the average calculation time of MOPD is 2.9 seconds, which is much less than the calculation time of WDP.

Figure 4 shows optimal results of case 3. The red line is the train trajectory of WODP and the blue line is the train trajectory of WDP. The regimes changes of WODP optimal solution is 7 and the regimes changes of WDP optimal solution is 12.

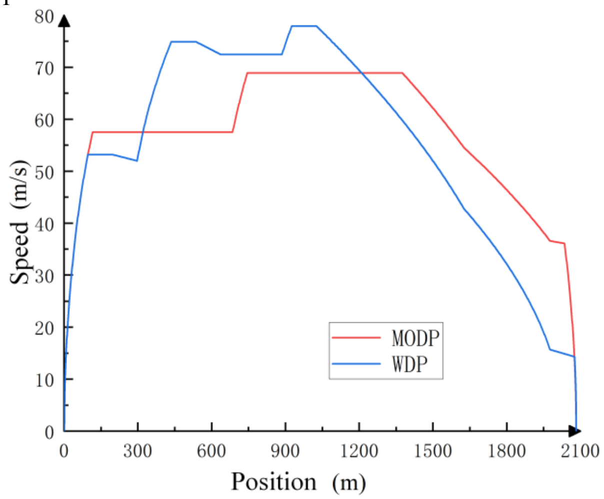

Figure 4. Optimal train trajectories of case 4

4 Conclusion

In this paper, we proposed a novel DP based approach for optimizing driving strategy. The optimal driving problem was first considered as a multi-objective problem with five optimal targets (i.e., energy saving, punctual arriving, less switching, safe driving and accurate stopping). The optimization problem was remodelled as a multistage decision problem by discretizing the continuous train movement in space. The process of DP was carried out in the velocity-space status space. Due to the discretizing rules of searching space, the optimal goals of safe driving and accurate stopping can be satisfied during the searching process. The rest of multiple goals were spilt into cost functions and constrains for each stage. Due to the multiple cost functions, a set of pareto optimal solutions can be achieved at each vertex during the process of dynamic programming. To further improve the efficiency of algorithm, regimes changes and congestion degree was introduced to maintain the capacity of the pareto set at each vertex.

Table 2. Optimal results

Case	Distance /m	Trip time /s	Flatout cost /kW·h	MODP				WDP				
				Arr. /s	Sav. %	Sw.	Cal. /s	Arr. /s	Sav. %	Sw.	Cal. /s	Δ. %
1	1334	105	18.6	0.2	56.5	8	1.4	-4.3	55.3	10	56.8	1.2
2	1286	102	29.4	-1.8	51.2	8	2.4	-4.7	50.9	9	51.8	0.4
3	2086	140	38.1	-1.6	31.9	7	5.0	-7.7	31.2	12	79.3	0.7
4	2265	150	22.8	-0.2	43.0	6	3.6	-4.0	37.9	12	88.9	5.1
5	2338	164	28.5	-1.4	52.0	9	4.1	-5.2	51.4	13	88.4	0.5
6	1354	104	22.6	-0.6	52.8	6	1.6	-3.7	52.2	9	55.4	0.6
7	1280	103	21.6	-0.8	66.1	5	1.3	-0.9	66.2	7	52.3	0.0
8	1538	114	24.2	-0.1	55.5	7	1.9	-0.7	55.3	8	61.6	0.2
9	993	90	21.4	-0.6	68.4	6	1.0	-1.8	68.3	8	41.1	0.2
10	1982	135	26.0	-0.2	54.9	8	3.2	-3.0	48.9	9	81.2	6.0
11	2366	157	21.6	-0.1	55.5	8	3.3	-2.8	53.3	14	95.4	2.1
12	1275	108	23.4	0.0	58.4	7	2.4	-6.3	55.7	12	51.7	2.7
13	2631	190	32.1	-1.5	62.2	7	6.3	-0.7	59.5	20	101.5	2.7
Avg.				-0.7	54.5	7.0	2.9	-3.5	52.8	11.5	69.7	1.7

A case study of Yizhuang urban rail line in Beijing was conducted to verify the effectiveness and the efficiency of DP based algorithms. The results show that our approach obtained higher quality solutions. Comparing with other DP based algorithm, the proposed algorithm resulted in 1.7 percent energy saving improvement, while the average running time deviation was -0.7 s and average regimes changes was 7 times. Furthermore, the computational time savings was significant compared to other DP based algorithm. In order to satisfy the real-time caculation requirement of train-borne equipment, the discretization parameters ((i.e., the number of stages K, the velocity interval Δv, the capacity of pareto set W_{limit})) should be set to the proper values The parameters setting decision made by users according to the actual subway is a trade-off between algorithm effectiveness and efficiency.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61703349, the Fundamental Research Funds for the Central Universities under Grant No. 2682017CX101.

References

1. K. Ichikawa, Application of optimization theory for bounded state variable problems to the operation of train, Bull. JSME, **11**(47), 857-865 (1968)
2. P. Howlett, The optimal control of a train, ANN OPER RES, **98**, 65-87 (2000)
3. R. Liu, I. M. Golovitcher, Energy-efficient operation of rail vehicles, TRANSPORT RES A-POL, **37**(10), 917-932 (2003)
4. E. Khmelnitsky, On an optimal control problem of train operation, IEEE T AUTOMAT CONTR, **45**(7), 1257-1266 (2000)
5. C. S. Chang, S. S. Sim, Optimising train movements thr-ough coast control using genetic algorithms, IEE P-ELECT POW APPL, **144**(1), 65-73 (1997)
6. K. K. Wong, T. K. Ho, Coast control for mass rapid tr-ansit railways with searching methods, IEE P-ELECT POW APPL, **151**(3), 365-376 (2004)
7. M. Miyatake, H. Ko, Optimization of train speed profile for minimum energy consumption, IEEJ T ELECTR, **5**(3), 263-269 (2010)
8. S. Lu, S. Hillmansen, T. K. Ho, C. Roberts, Single-train trajectory optimization, IEEE T INTELL TRANSP, **14**(2), 743-750 (2013)
9. J. T. Haahr, D. Pisinger, M. Sabbaghian, A dynamic programming approach for optimizing train speed profiles with speed restrictions and passage points, TRANSPORT RES B-METH, **99**, 167-182 (2017)
10. H. Tang, J. Zhu, Q. Wang, X. Feng, An on-line adjustable control algorithm for on-time and energy saving operations of trains, China Railway Science, **34**(4), 89-94 (2013)
11. X. Yang, X. Li, B. Ning, T. Tang, An optimisation method for train scheduling with minimum energy consumption and travel time in metro rail systems, TRANSPORTMETRICA, **3**(2), 79-98 (2015)