Train Control Policy Optimization based on Reinforcement Learning

Yanjia Zhao
NetId: yz476

April 2020

1 Introduction

In the field of train control, Automatic Train Operation(ATO) system can play the role of a driver, to control the riding of the train. By optimizing driving strategy, we can lower the energy consumption of the train operation, improve riding comfort and punctuality. In this project, I will try to use several methods in reinforcement learning to do the optimization.

2 Problem Analysis

2.1 Markov Process of Train Control

Reinforcement learning tasks are usually regarded as Markov decision process. The process of policy training is: The train has state space $\mathcal{S}$. Each state s in $\mathcal{S}$ is the running environment of the train, which are the train speed, position and running time. According to current state, the agent will take a best action. In this problem, actions are acceleration, brake, and coast. Then the train will transit to a next state s', with a transition probability $p(s', r|s, a)$. According to the reward returned by the environment, along with this transition, the agent will update the current policy. This process will repeat until the policy can converge.

2.2 Environment setting

I choose the data specification of Yizhuang subway line in Beijing to simulate the running environment of the train. The mass of the train is 192 ton. The trip distance is 1.23 mile. And in some sections of the trip, maximum speed limit is applied. To guarantee the comfort of passengers, the maximum acceleration of the train is also limited, based on [1]. And the route is not flat, we have to take care of the gradient.

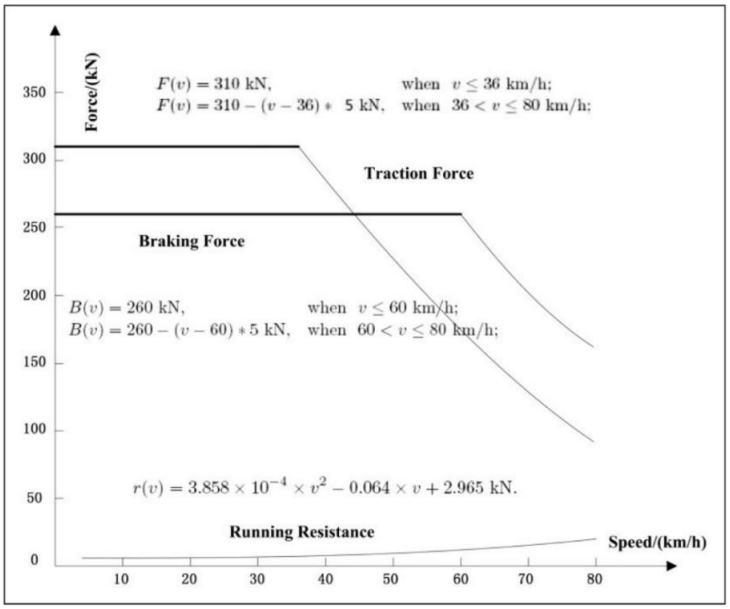

Figure 1: Traction force, braking force and resistance changes versus velocity

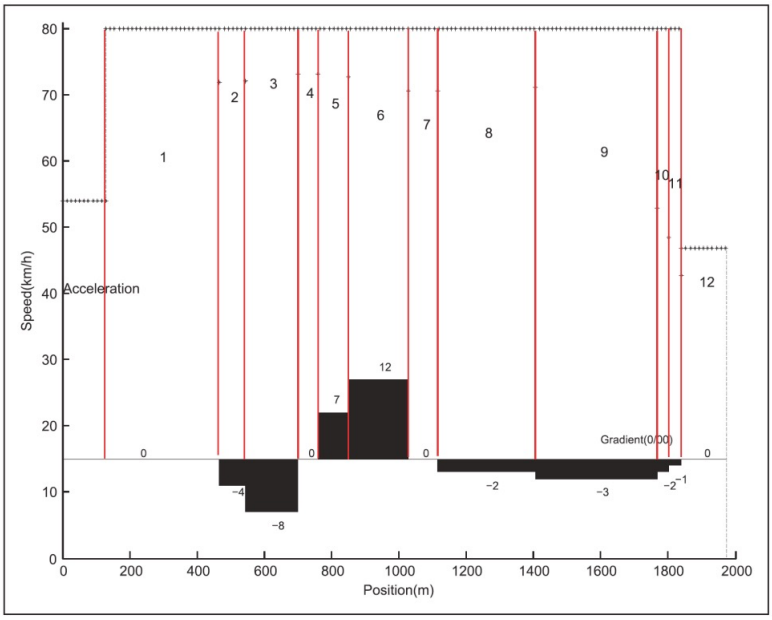

Figure 2: Slope changes along the route, and its effect on acceleration

2.3 Motion Model

The energy consumption of traction is the integration of real-time power against running time. The mechanical power is:

$$P = F(v)v(x)$$

The motion of the train can be analyzed with classical kinematics:

$$\frac{dv}{dx} = \frac{p(x)F(v) - q(x)B(v) - R(v) - G(x)}{mv(x)}$$

$$\frac{dt}{dx} = \frac{1}{v}$$

where B is braking force, R is resistance, G is gravitational acceleration, and $p, q \in [0, 1]$, which represent the restriction of traction and braking force.

2.4 Reward

The operation efficiency of the train focuses on punctuality, energy efficiency and passengers' comfort.

2.4.1 Punctuality

Punctuality is a very important specification in train operation. Especially in rush hour, when the time difference between adjacent train is very short, a minor time delay of one train will result in the delays of multiple trains, or even the whole system. Also the delay will affect efficiency of passengers when they want to transit to other lines or other transportation approaches. So we define the running time difference between scheduled running time and actual running time:

$$e_t = \alpha|T - T'|$$

where α is a weight hyper-parameter to be tuned in the training.

2.4.2 Comfort

The passengers on the train will move along with the train, due to inertia. The acceleration and jerk are usually quantified to evaluate the influence of train movement to passenger comfort. According to [1], for urban rail train, we assume the satisfaction of passengers is over 95%, the maximum acceleration is about $1.2m/s^2$ The peek to peek value of acceleration should be smaller than $0.306m/s^2$ In another word, the jerk during acceleration should be less than $0.153m/s^3$

2.4.3 Energy efficiency

$$E = \int_0^S F(v)ds$$

In a summary, we want to develop a new policy to lower the energy consumption of train operation, in the context of punctuality guarantee.

2.5 Driving Policy

Based on [2] and [4], the train control policy for energy conservation should consist of maximum acceleration, maximum braking, and coasting.

2.5.1 Maximum Acceleration and Braking

The slower when a train accelerate or brake, the more time it will take to stop. In order to achieve the same running time in a slower acceleration and braking situation, the train has to accelerate to a higher speed, which consume more energy. So the maximum acceleration and braking must be the most energy conserved.

2.5.2 Coasting

Coasting means no traction force and braking force. The train will move forward without consuming energy. So the train can save more energy if it starts to coast at an earlier time. In a summary, when the train just launches, it applies maximum acceleration to reach high speed. During the middle of the trip, control policy will transfer between maximum acceleration and coasting to save energy. In the end, applying maximum braking stops the train at the next station.

3 Reinforcement Learning Methods

3.1 Dynamic Programming

Dynamic programming can be used to compute optimal policies given a perfect model of the environment as a Markov decision process. The key idea of DP, and of reinforcement learning generally, is the use of value functions to organize and structure the search for good policies. We can easily obtain optimal policies once we have found the optimal value functions, $v*$ or $q*$, which satisfy the Bellman optimality equations:

$$\begin{aligned} v_*(s) &= \max_a E[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_*(s')] \end{aligned}$$

or

$$q_*(s,a) = E[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a')|S_t = s, A_t = a]$$

$$= \sum_{s',r} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')]$$

3.1.1 Policy Evaluation

Policy Evaluation is to compute the state-value V_π for a given policy π:

$$V_{t+1}(s) = E_\pi[r + \gamma V_t(s')|S_t = s] = \sum_a \pi(a|s) \sum_{s',r} P(s', r|s, a)(r + \gamma V_k(s'))$$

3.1.2 Policy Improvement

Based on the value functions, Policy Improvement generates a better policy $\pi' \geq \pi$ by acting greedily.

$$Q_\pi(s,a) = E[R_{t+1} + \gamma V_\pi(S_{t+1}|S_t = s, A_t = a] = \sum_{s',r} P(s', r|s, a)(r + \gamma V_k(s'))$$

3.1.3 Policy Iteration

The Generalized Policy Iteration (GPI) algorithm refers to an iterative procedure to improve the policy when combining policy evaluation and improvement.

$$\pi_0 \xrightarrow{\text{evaluation}} V_{\pi_0} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluation}} V_{\pi_1} \xrightarrow{\text{improve}} \pi_2 \xrightarrow{\text{evaluation}} ...$$

$$\xrightarrow{\text{improve}} \pi_* \xrightarrow{\text{evaluation}} V_*$$

In GPI, the value function is approximated repeatedly to be closer to the true value of the current policy and in the meantime, the policy is improved repeatedly to approach optimality. This policy iteration process works and always converges to the optimality.

3.1.4 Value Iteration

This algorithm is similar to GPI, It can be written as a particularly simple backup operation that combines the policy improvement and truncated policy evaluation steps:

$$v_{k+1}(s) = \max_a E[R_{t+1} + \gamma v_k(S_{t+1})|S_t = s, A_t = a]$$

$$= \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')]$$

The whole steps can be stated as:

Initialize array V arbitrarily

Repeat
$\Delta \leftarrow 0$
For each $s \in \mathcal{S}$:
$v \leftarrow V(s)$
$V(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
$\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$(a small positive number)

Output a deterministic policy, π, such that
$\pi(s) = arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$

And value iteration is the algorithm that I choose to optimize the train control policy at first.

3.2 Q-learning

Dynamic programming is a traversal method, where all possible states will be evaluated at each iteration. However, not all states combinations are possibly reachable for a running train. For example, it's not possible that a train will reach a very high speed when it just launches. Instead, Q-learning algorithm will start from a specified start state, take actions until a terminated state is reached, then go back to the start state and repeat. This method will avoid many unnecessary states evaluation. Moreover, this algorithm doesn't require a perfect model of the environment. The environment could be a 'black box'. We can take actions in the environment and get reward, which is enough. The whole steps can be stated as:

1. At time step t, we start from state S_t and pick action according to Q values, $A_t = arg\max_{a \in \mathcal{A}} Q(S_t, a)$.
2. With action A_t, we observe reward R_{t+1} and get into the next state S_{t+1}.
3.Update the action-value function:
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(S_{t+1}, a))$.
4. $t = t + 1$ and repeat from step 1.

4 Training Results

4.1 Value Iteration

The training process took about 20 minutes. And the trained policy can successfully guide the operation of the train, as shown in the figure 3. The energy consumption of the whole trip is 26.7 kwh, which is less than 30 kwh(the energy

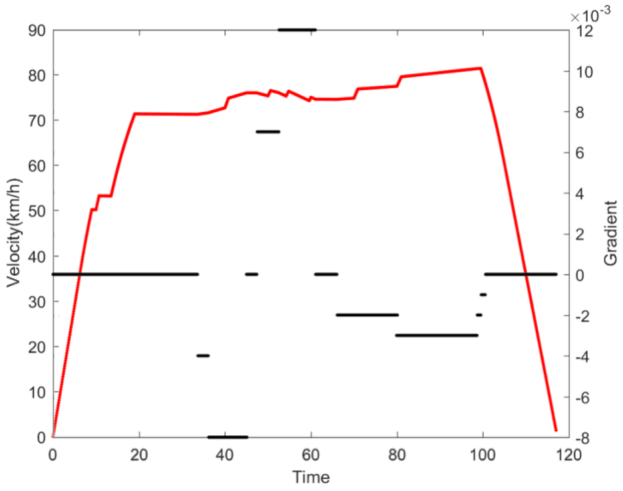

Figure 3: Time - Velocity/Gradient curve

consumption of an original policy) The running time is 112.1s, which is slightly different from scheduled running time 115s.

4.2 Q-learning

For Q-learning algorithm, the tuning of some hyper-parameter is more difficult than that of value iteration algorithm. I expect Q-learning will run faster. However, it finally took 30 minutes until convergence based hyper-parameters I chose, which is even slower than value iteration.

5 Conclusion

The trained policy can effectively lower the energy consumption based on the premise of punctuality and passengers' comfort. In this project I applied two relatively simple reinforcement learning algorithms. Other more complicated algorithms may produce better results. Comparing these two algorithms, even Q-learning is intuitively more straightforward and involve less computation, the dynamic programming algorithm is faster. I believe the reason is that when the model of the environment is fully known(which is we can use Bellman equation), DP algorithm can usually converge very fast. This idea is also introduced in [3]. So which algorithm is the best one to use depends on which kinds of problems we are going to solve. There is no a determined answer for this question.

References

[1] European committee. *Railway applications - Ride comfort for passengers - Measurement and evaluation.* British Standards, 2009.

[2] P. G. Howlett. *Optimal strategies for the control of a train.* Automatica, vol. 32, no. 4, pp. 519–532, Apr. 1996.

[3] Andrew G. Barto Richard S. Sutton. *Reinforcement Learning: An Introduction, Second edition.* The MIT Press, 2014,2015.

[4] X. Li S. Su, T. Tang. *Driving strategy optimization for trains in subway systems.* P I Mech Eng F-J Rai, 2016.