Massachusetts Institute of Technology
Engineering Systems Division

ESD Working Paper Series

An approximate dynamic programming approach for designing train timetables

Maite Peña-Alcaraz
MIT Engineering Systems Division
Room E40-235
77 Massachusetts Ave
Cambridge MA 02139

Mort Webster
MIT Engineering Systems Division
Room E40-235
77 Massachusetts Ave
Cambridge MA 02139
Email: mort@mit.edu

Andrés Ramos
Profesor Propio Ordinario
Departamento de Organización Industrial
Instituto de Investigación Tecnológica
Escuela Técnica Superior de Ingeniería (ICAI)
Universidad Pontificia Comillas
Alberto Aguilera 23
28015 Madrid, Spain

ESD-WP-2011-11 August 2011 esd.mit.edu/wps

An approximate dynamic programming approach for designing train timetables

Maite Peña-Alcaraz, Mort Webster, Andrés Ramos

Abstract— **Traditional approaches to solving the train timetabling problem – the optimal allocation of when each train arrives and departs each station – have relied on Mixed-Integer Programming (MIP) approaches. We propose an alternative formulation for this problem based on the modeling and algorithmic framework of approximate dynamic programming. We present a Q-learning algorithm in order to tractably solve the high-dimensional problem. We compare the performance of several variants of this approach, including discretizing the state and the action spaces, and continuous function approximation with global basis functions. We demonstrate the algorithms on two railway system cases, one minimizing energy consumption subject to punctuality constraints, and one maximizing capacity subject to safety constraints. We demonstrate that the ADP algorithm converges rapidly to an optimal solution, and that the number of iterations required increases linearly in the size of the rail system, in contrast with MIP approaches whose computation time grows exponentially. We also show that an additional benefit to the ADP approach is the intuition gained from visualizing the Q-factor functions, which graphically capture the intuitive tradeoffs between efficiency and constraints in both examples.**

Index Terms— **Approximate dynamic programming, train timetabling problem, robust scheduling, strategies for train timetable design**

I. Introduction

RAILWAY transportation has become a rapid, clean and efficient way to transport passengers and freight within a modern economy. Consequently, planning the capacity to be able to satisfy the expected demand represents a central problem in railway management. Moreover, railway transportation demand is increasing rapidly. An increase of 25% in its activity is expected in Europe by 2030 (Capros et al. 2007) and capacity will need to be expanded to meet this demand. However, because the rail sector is intensive in capital, careful capacity analysis should be performed before investing in a given project.

Several distinct problems arise when the infrastructure and the services are planned to satisfy the demand and when the operation is adjusted on-line to follow the operational plan. In his annotated bibliography, (Assad, 1981) collected and classified several studies in the literature using network and timetabling models. Since then, rail transportation problems have been classified into three levels according to the planning time horizons: strategic planning, tactical planning, and operational or regulation level (Cordeau et al. 1998; Crainic 2003; Ghoseiri et al. 2004).

At the strategic level, demand is projected and the line is planned in order to provide a given capacity when the infrastructure is being designed, or to increase the capacity when part of the infrastructure is being overhauled. The set of operational policies is also designed at this level. At the tactical level, timetables are formulated. These timetables result in schedules for the services. After that, trains and crew are assigned to lines (rolling-stock and crew scheduling and rostering). The main difference between tactical and strategic planning is that the investments in the rail network and in the trains are chosen at the strategic level while the tactical level decides how the available infrastructure and rolling-stock should be used. The level of detail with which the movements of the trains are considered in tactical models is considerably higher than that in strategic models. Finally, at the operational level, real-time data is used to decide how to modify the operation to maintain service quality.

The train timetabling problem appears at all three functional levels: as a tool to calculate the capacity at the strategic level (Abril et al., 2008; Lai & Barkan, 2009; Landex, 2009), as the central problem at the tactical level (Caprara et al., 2002; Crainic, 2003; Castillo et al., 2009) and as a response to important incidences on the line at the operational level (Kraay & Harker, 1995; Fernández et al., 2006); with different characteristics, uses and requirements.

Here, we focus on the tactical level version of the timetabling problem, because it is the critical set of operational decisions in railway systems with complex topologies, such as commuter lines in large metropolitan areas. However, because it is crucial also for decisions at strategic or operational levels, the development of appropriate mathematical formulations and efficient solution algorithms for the train timetabling problem will have a broad impact on planning.

The train timetabling problem consists of choosing a complete timetable (passing, arrival and departure times of each train at every point of the line) for a given mix of trains running under certain operational policies along a given infrastructure. As a tactical problem, the objective is to determine medium term decisions to maximize the performance of the operations once the long-term decisions and operational policies have been adopted.

In the literature, three different optimization approaches have been applied to the timetabling problem:

- In the first approach, the arrival and departure times of every train at every station are the decision or control variables. The constraints are specifically written in terms of the arrival and departure time of the trains, and the problem is formulated as a Mixed-Integer Programming (MIP) problem. See (Castillo et al., 2009; Ghoseiri et al., 2004; Liebchen et al., 2004; Zhou & Zhong, 2005) for different formulations with different objective functions and different traffic constraints. Traditionally, these models have been called multi-mode resource constrained project scheduling models.

- The second approach models the timetabling problem as a multi-commodity flow problem (Caimi et al., 2009; Caimi et al., 2010 (in press); Caprara et al., 2002; Caprara et al., 2007; Cordeau et al., 1998). These models use nodes to represent arrival or departures of trains to stations and arcs to represent whether that arrival or departure occurs at particular time. Consequently, the decision variable is a vector of binary controls indicating which arcs have been selected. The constraints indicate which combinations of arcs are infeasible. The models are written as MIP problems. To prevent an explosion of the size of the model, initial timetables are usually required in practice. Furthermore, time discretization, consideration of a limited number of travel times or departure times, is also required to keep the solution computationally tractable.

- The third approach uses dynamic programming (DP) (Lee & Pinedo, 2002) for solving timetabling problems instead of the traditional branch and bound based techniques for solving MIP problems. In practice, the solution of the classical DP formulation becomes intractable when the size of the problem grows too large. However, an alternative class of solution algorithms using approximate techniques (Bertsekas & Tsitsiklis, 1996; Powell, 2007) enables the consideration of a DP approach. To the best of our knowledge, very little work has been done to apply DP techniques to the general train timetabling problems, although they have been applied to related problems such as locomotives assignment (Bouzaiene-Ayari & Powell, 2010).

In this paper we propose to model the train timetabling problem using an approximate dynamic programming (ADP) paradigm, see (Bertsekas & Tsitsiklis, 1996; Bertsekas, 2005a; Bertsekas, 2005b; Powell, 2007). There are two main reasons for choosing an ADP formulation over a MIP formulation. First, the solutions obtained with an ADP formulation provides additional information about the timetables and allows the user to identify qualitative strategies for designing timetables, making it easier to understand the impact of changes on the timetable. Furthermore, the train timetabling problem is known to be NP-hard (Zhou & Zhong, 2007) and very difficult to solve when the size of the problem increases. While the solution time of a mathematical programming approach grows exponentially with the size of the problem; the solution time of an ADP formulation increases only as a polynomial function (in our case, quadratic) of the size of the problem.

This paper is structured as follows: two different train timetable problems are presented and formulated in section II. Two different algorithms used to determine the timetables are presented in section III. Finally, the main results are discussed in section IV and the conclusions are presented in section V.

II. Train timetabling problem

In this section, we present two example timetable problems. We first describe the main elements of each problem. Then, both problems are formulated as DP problems, and we motivate the use of ADP algorithms to solve it.

A. Problem description

As described above, the train timetabling problem consists of choosing the arrival, departure, and passing times of every train at every point of the infrastructure under a given operational policy.

The first example is a single high-speed rail line, for which the objective is to determine the timetable for the intermediate stations that minimizes energy consumption, subject to total travel time and punctuality requirements. The second example is a metro or commuter line, for which the timetable is chosen to maximize the used capacity subject to safety constraints.

In the first case, the trains are driven manually, so the travel time between different stations is continuous. In the second case, it is assumed that the trains are automatically driven, and therefore the set of possible travel times is constrained. Moreover, while there are no constraints coupling the trains in the first design, several safety constraints should be considered in the second design to avoid train conflicts.

1) Minimizing energy consumption with punctuality constraints

Figure 1 shows the energy consumption of a train (manually driven) for different travel times. Different driving strategies may result in different levels of energy consumption for the same total travel time, although we assume for this paper that the drivers drive in the most efficient way once the travel time has been determined. The energy consumption of the points located along the Pareto frontier decreases monotonically as the travel time increases. Thus, traveling at maximum speed (flat-out) has associated the minimum travel time (around 2180 s in this case) and the maximum energy consumption (around 2940 kWh). According to figure 1, it is possible to decrease energy consumption dramatically using a few minutes for eco-driving at each inter-station.

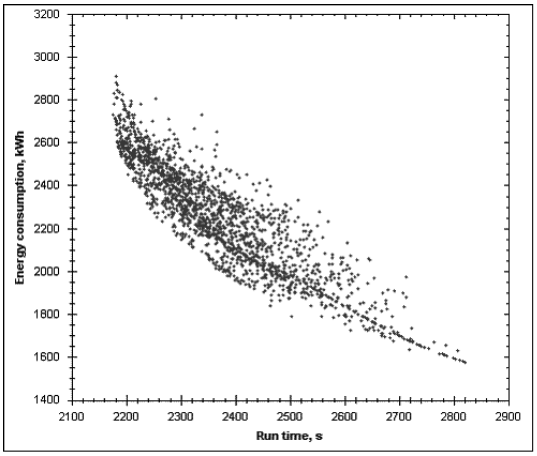


Figure 1 Energy consumption versus travel time for different driving profiles at a given inter-station

The total travel time between the first and the last station is generally greater than the sum of the minimum travel times at each inter-station. The timetable design consists of allocating the remaining time (called total slack) along the different inter-stations. The slack assigned to any inter-station must be greater than a given minimum slack to ensure punctuality using this deterministic criterion. This minimum slack depends on the delays that usually appear on the line. Once punctuality constraints are met, the remaining slack can be allocated to those inter-stations where the potential of energy saving is greater. In many lines, approximately 85% of the trains arrive with no delay (Renfe, 2010), meaning that slack is usually available to reduce energy consumption.

2) Maximize used capacity subject to safety constraints

Another central issue in designing the timetable for a railway system is the allocation of the capacity. The railway manager might be interested in improving the use of the infrastructure to better serve the demand. This example simulates the timetable design for a railway system consisting of an underground double-track circular line. All the trains circulate along the line, stopping at every station. Trains are not allowed to overtake another. The characteristics of each train determine the travel and stopping times at the stations, which varies across different trains.

Since the line is a double-track line, in nominal working conditions, each track is used by the trains travelling in one direction, so the design of the timetable for each direction can be set independently. Figure 2 represents one possible situation of the trains travelling along the line at one instant in time.

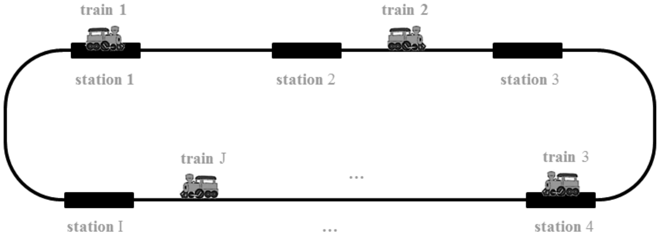


Figure 2 Circular underground line

The goal is to design a feasible timetable that maximizes the capacity of the line by increasing the frequency of trains (reducing the time between consecutive arrivals of trains to the same station). In this context, feasible means that operating constraints (stopping times must be bounded, so do travel times, etc.) and safety constraints (ensure that the distance between consecutive trains does not compromise the safety of the line) are all met.

B. DP formulation

As discussed above, the design of timetables becomes extremely difficult as the size of the problem increases. Consequently, a consistent mathematical formulation is required to automatically design the timetables. In this subsection we present a DP formulation of both problems. The index i is used for the stations and the index j is used for the trains. I and J represent, respectively, the total number of stations and the total number of trains scheduled.

1) Minimizing energy consumption with punctuality constraints

This problem is framed as a discrete time, finite horizon, stochastic DP problem with continuous state and control spaces, see (Bertsekas, 2005a; Powell, 2007).

In our framework, each decision stage represents a station.

All the information required at each stage is captured by the state variable, the departure time of the train at the station:

$$DT_i \in \mathbb{R}_0^+,\ i = 1,...,I \tag{1}$$

The state variable is Markovian; i.e., knowing the state variable at one station (stage) is sufficient to compute the timetable for all the following stations, and the states at previous stations provide no additional information.

The control variable is the total travel time from the current station to the next station:

$$T_i \in \left(\underline{t}_i, \bar{t}_i\right),\ i = 1,...,I-1 \tag{2}$$

where $\underline{t}_i$, $\bar{t}_i$ represent, respectively, the minimum and maximum possible travel time from one station to the next. These values are continuous given that the high-speed trains are manually driven.

In this problem, the stochastic disturbance (or exogenous variable) is represented by the total delay that occurs during the travel to the next station:

$$\omega_i \in \mathbb{R}_0^+,\ i = 1,...,I-1 \tag{3}$$

Once the state variable, the control and the disturbance are known for a given stage (station), the state at the next station can be determined. The state transition function gives the departure time of the train at the next station. This departure time is equal to the designed travel time (departure time plus nominal travel time) when it is possible to recover from the delay:

$$DT_{i+1} = f_i\left(DT_i, T_i, \omega_i\right) = \max\left(DT_i + T_i, DT_i + \omega_i\right) \quad (4)$$

The cost associated with a given timetable can be calculated using a cost function which represents the energy consumption at each stage (station). This consumption can be calculated from the travel time (see figure 1):

$$\begin{aligned} &g_i\left(DT_i, T_i, \omega_i\right) = C_i\left(T_i\right) \\ &g_I\left(DT_I\right) = 0 \end{aligned} \quad (5)$$

For this problem, there is no constraint coupling different trains.

The objective of the design is to minimize the cost function (energy consumption) over all the stations:

$$\min_{T \in U(DT)} \left(\sum_{i=1,\dots,I-1} \alpha^i g_i\left(DT_i, T_i\right) + \alpha^I g_I\left(DT_I\right) \right) \quad (6)$$

where $U(DT)$ is the set of feasible travel times given the departure time and α is the discount factor between decision stages, although in this paper we fix α to be equal to one[1].

To ensure that the slack is primarily used to meet the required level of punctuality p (fraction of times that the train arrives on-time), the minimum possible travel time $\underline{t}_i$ is computed as the minimum travel time (corresponding to flat-out) plus a minimum slack. Consequently, when no delay occurs, the driver uses all the time for eco-driving and hence saves some energy during the trip. In the event that a delay occurs, the driver drives faster and is able to recover any delay smaller than the slack allocated. Therefore, the minimum slack should be greater than the delay associated to the p percentile of the accumulated probability at the station.

Furthermore, to ensure that total travel time is smaller than the maximum total travel time given, the maximum possible travel time from each station $\overline{t}_i$ should be equal to the minimum of the slowest possible driving strategy allowed and the remaining slack to be allocated after ensuring punctuality in the remaining stations.

The optimal cost-to-go function, which indicates the additional minimum cost from a given station and departure time up to arrival at the last station, is:

$$\begin{aligned} &J_i^*\left(DT_i\right) = \min_{T_i \in U_i(DT_i)} \left(\underset{\omega_i}{E} \left[\begin{aligned} &g_i\left(DT_i, T_i, \omega_i\right) + \\ &\alpha J_{i+1}^*\left(f_i\left(DT_i, T_i, \omega_i\right)\right) \end{aligned} \right] \right) \\ &J_{I+1}^*\left(DT_{I+1}\right) = 0 \end{aligned} \quad (7)$$

Though this non-linear, recursive equation may seem easy to solve backwards at first glance, it suffer from three curses of dimensionality (Powell, 2007): the dimension of the state space for each stage DT_i determines the number of equations, the dimension of the control space for each stage T_i determines the size of the space over which the minimization is carried out and the dimension of the disturbance space for each stage ω_i determines the size of the space over which the expected value is computed.

Since these three variables are continuous in reality, computing the exact cost-to-go function is extremely difficult. Instead, we approximate the cost-to-go function using ADP algorithms.

2) Maximize used capacity subject to safety constraints

This problem is also framed as a discrete-time, finite horizon, deterministic DP problem with continuous state and control spaces (Bertsekas, 2005a).

As above, the decision stages correspond to the individual stations, and the state variable is a vector representing the travel time of all trains at a given station:

$$DT_i = \left(DT_{ij}\right)_{j=1,\dots,J} \in \mathbb{R}_0^+,\ i = 1,\dots,I+1 \quad (8)$$

where the initial condition of the line, DT_1, is assumed to be fixed. In this problem, several safety constraints are imposed, which require some minimal distance between trains along the line in order to avoid conflicts. This is the reason for including the subscript j in the previous equation.

The control variable is the total travel time of each train to the next station:

$$T_i = \left(T_{ij}\right)_{j=1,\dots,J} \in U_i\left(DT_i\right) \subset \mathbb{R}_0^+,\ i = 1,\dots,I \quad (9)$$

These trains are automatically driven and therefore the set of possible controls is constrained. Additional constraints are imposed on the set of possible controls $U_i\left(DT_i\right),\ i = 1,\dots,I$ as follows:

First, the control must be bounded:

$$\underline{st}_{ij} + \underline{tt}_{ij} \le T_{ij} \le \overline{st}_{ij} + \overline{tt}_{ij}, \forall j \quad (10a)$$

st_{ij}, tt_{ij} are, respectively, the (minimum and maximum) stopping and travel times of train j at/from station i.

Furthermore, the control T_{ij} must be chosen so that the headway between the departure of a train from the next station and the following arrival is greater than the minimum safety headway $\underline{h}_i$:

$$T_{ij} - \overline{st}_{ij+1} \ge \underline{h}_i, \forall j \quad (10b)$$

Finally, to ensure that the control T_{ij} results in a feasible timetable, additional constraints are needed:

[1] This formulation is formally known as a “total cost problem” as opposed to a “total discounted cost problem”. Total cost problems can have theoretical convergence issues, especially for classic solution techniques. We demonstrate consistent convergence below for the ADP algorithms used.

$$\begin{aligned}&\sum_{i'<i} T_{i'j} + T_{ij} + \sum_{i'>i}\left(\underline{st}_{ij} + \underline{tt}_{ij}\right) \le && \forall j \\ &\quad \min_j\left(\sum_{i'<i} T_{i'j} + T_{ij} + \sum_{i'>i}\left(\overline{st}_{ij} + \overline{tt}_{ij}\right)\right) \\ &\sum_{i'<i} T_{i'j} + T_{ij} + \sum_{i'>i}\left(\overline{st}_{ij} + \overline{tt}_{ij}\right) \ge && \forall j \\ &\quad \max\left(\begin{array}{r}\max_j\left(\sum_{i'<i} T_{i'j} + T_{ij} + \sum_{i'>i}\left(\underline{st}_{ij} + \underline{tt}_{ij}\right)\right), \\ DT_{1J} + \underline{h}_i + \underline{st}_{11}\end{array}\right) \\ &\sum_{i'<I} T_{i'j} + T_{Ij} = \sum_{i'<I} T_{i'j'} + T_{Ij'} && \forall j \forall j'\end{aligned} \tag{10c}$$

The idea of these constraints is that once the controls $T_{i'}$ for all previous stations $\forall i' \le i$, we can determine both the minimum period (time required to loop over all stations and return to the initial station) and the maximum period-to-go. We wish to ensure that the minimum period is less than or equal to the maximum period, $\underline{p}_i \le \overline{p}_i$. In the first constraint, the largest possible period is $\overline{p}_i = \min_j\left(\sum_{i'\le i} T_{i'j} + T_{ij} + \sum_{i'>i}\left(\overline{st}_{ij} + \overline{tt}_{ij}\right)\right)$. In the second constraint, the smallest possible period is the maximum of $DT_{1J} + \underline{h}_1 + \underline{st}_{11}$ and $\max_j\left(\sum_{i'<i} T_{i'j} + T_{ij} + \sum_{i'>i}\left(\underline{st}_{ij} + \underline{tt}_{ij}\right)\right)$. The third equation ensures that the period is the same for all trains, so all trains will loop over all stations in the same amount of time.

Notice that in order to compute the set of possible controls $U_i\left(DT_i, \sum_{i'<i} T_{i'}\right)$, we need to know the state as well as $\sum_{i'<i} T_{i'}$. Consequently it is required to augment the state adding the information $\sum_{i'<i} T_{i'}$, in order to preserve the Markov property.

The set of constraints required in a MIP model is represented simply by the range of feasible controls, $\underline{t}_{ij}, \overline{t}_{ij}$, for a given stage and state, $T_{ij} \in \left(\underline{t}_{ij}, \overline{t}_{ij}\right)$, $j = 1,\ldots,J$ in a DP framework. Note, however, that the computation of the actual $\underline{t}_{ij}, \overline{t}_{ij}, \forall i \forall j$ is not trivial and causes the state augmentation as we mention above.

The state transition function in this case is:

$$\begin{aligned}&DT_{i+1} = \left(DT_{i+1j}\right)_{j=1,\ldots,J} = && i = 1,\ldots,I \\ &\quad \left(DT_{ij} + T_{ij}\right)_{j=1,\ldots,J} = DT_i + T_i\end{aligned} \tag{11}$$

The cost function is:

$$\begin{aligned}&g_i\left(DT_i, T_i\right) = g\left(DT_i, T_i\right) = && i = 1,\ldots,I \\ &\quad \max_j\left(DT_{ij} + T_{ij}\right) - \max_j DT_{ij} \\ &g_{I+1}\left(DT_{I+1}\right) = 0\end{aligned} \tag{12}$$

Finally, the Bellman equation or cost-to-go function for this problem is:

$$\begin{aligned}&J_i^*\left(DT_i\right) = && \forall i \le I \forall DT_i \in \mathbb{R}_0^+ \\ &\quad \min_{T_i \in U_i(DT_i)}\left(g\left(DT_i, T_i\right) + \alpha J_{i+1}^*\left(DT_i + T_i\right)\right) \\ &J_{I+1}^*\left(DT_{I+1}\right) = 0, && \forall DT_{I+1} \in \mathbb{R}_0^+\end{aligned} \tag{13}$$

where $J_i^*\left(DT_i\right)$ is the optimal cost-to-go (additional time required to schedule the trains) if the departure of the trains at station i occurs at time $DT_i = \left(DT_{ij}\right)_{j=1,\ldots,J}$.

In practice, it is very difficult to solve for $J_i^*\left(DT_i\right), \forall DT_i \in \mathbb{R}$. Even knowing that $0 \le DT_{ij} \le 2 \cdot \overline{p}_i, \forall i \forall j$ and discretizing the possible values that both the departure time (state) and the total travel time (control) could take, the size of the problem grows exponentially ($\#DT, \#T$ being the number of values of departure times and total travel times considered after the discretization):

- State size (initially): $O\left(J \cdot \#DT\right) \sim O\left(J \cdot I \cdot \#T\right)$
- State size (augmented): $O\left(J \cdot \#DT^2\right) \sim O\left(J \cdot I^2 \cdot \#T^2\right)$
- Control size: $O\left(J \cdot \#T\right)$
- Time horizon: $I + 1$

For a large system (trains + stations) a classic DP algorithm is impracticable to solve this problem. Alternative approximation methods are needed to be able to obtain the solution.

III. ADP ALGORITHMS

Because of the computational demands of a classical DP approach, we apply ADP algorithms which sample across states, controls, and disturbances. However, convergence issues can arise when sampling. Specifically, when x is a random variable, a sampling strategy will introduce a bias into the computation of $y = \min_x \left(E_x \left[x \right] \right)$. Bias can be eliminated when y can be computed as $y = E_z \left[\min_z \left(z \right) \right]$ by sampling the random variable z. In this latter case, the estimation error of y will decrease with the number of samples.

To address this concern, we use a Q-factor formulation (Bertsekas & Tsitsiklis, 1996). This approach computes the so-called Q-factors value instead of computing the cost-to-go function value presented in equation (7):

$$
\begin{aligned}
&Q_i^* \left(DT_i, T_i \right) = \qquad \forall i < I \\
&\underset{\omega_i}{E} \left[\begin{array}{l} g_i \left(DT_i, T_i, \omega_i \right) + \\ \alpha \min\limits_{T \in U_{i+1}\left(f_i\left(DT_i, T_i, \omega_i \right) \right)} Q_{i+1}^* \left(f_i \left(DT_i, T_i, \omega_i \right), T \right) \end{array} \right] \qquad (14) \\
&Q_I^* \left(DT_I, T_I \right) = \underset{\omega_I}{E} \left[g_I \left(DT_I, T_I, \omega_I \right) \right]
\end{aligned}
$$

The dimensionality of the Q-factor formulation is larger, because we need to keep track of the value of the control to avoid introducing a bias. Depending on the structure of the cost function, an alternative is to use a version of the cost-to-go function formulated with post-decision variables (Powell, 2007) instead of Q-factors, but this is not always possible.

In the next subsections, we present two alternative algorithms for computing the Q-factors.

A. Look-up table algorithm

The first concern with equation (14) is that the state, the control and the disturbance presented are continuous variables. Consequently, the first approximation consists of discretizing these variables to be able to iteratively compute the value of the Q-factors. In particular, the Gauss-Seidel Q-learning algorithm for finite horizon problems is presented (Bertsekas, 2005b).

The algorithm proposed is an approximate version of value iteration, in which the expected value in the previous expression is approximated by sampling and simulation. In particular, an infinitely long succession of possible departure times (state) and total travel time (control) for every station (stage) i, DT_i^k, T_i^k, $k = 1, \ldots$ is generated according to the probabilities of the problem. Given a pair $\left(DT_i^k, T_i^k \right)$, a disturbance ω_i^k is sampled according to the probability $p_\omega \left(DT_i^k, T_i^k \right)$. Then, the Q-factors are updated using a stochastic gradient approximation (Robbins & Monro, 1951):

$$
\begin{aligned}
&Q_i^k \left(DT_i^k, T_i^k \right) = \left(1 - \gamma_i^k \right) Q_i^{k-1} \left(DT_i^k, T_i^k \right) \forall i \le I, \forall k \\
&+ \gamma_i^k \left(\begin{array}{l} g_i \left(DT_i^k, T_i^k, \omega_i^k \right) + \\ \alpha \min\limits_{T \in U_{i+1}\left(f_i\left(DT_i^k, T_i^k, \omega_i^k \right) \right)} Q_{i+1}^k \left(f_i \left(DT_i^k, T_i^k, \omega_i^k \right), T \right) \end{array} \right) \qquad (15) \\
&Q_{I+1}^k \left(DT_{I+1}, T_{I+1} \right) = \left(1 - \gamma_I^k \right) Q_I^{k-1} \left(DT_I^k, T_I^k \right) \forall k \\
&+ \gamma_I^k g_I \left(DT_I^k, T_I^k, \omega_I^k \right)
\end{aligned}
$$

where $\gamma_i^k = \dfrac{1}{n_i^k}$, and n_i^k is the number of times that the station state-control pair $\left(DT_i^k, T_i^k \right)$, has been visited up until iteration k. The Q-factor values of other pairs that have not been visited in iteration k remain unchanged.

Step 0. Initialize: Set $k = 0, n_i \left(DT, T \right) = 0$

Step 1. Set $k = k + 1$. Then for all station $i = I$ to 1, generate $DT_i^k, T_i^k, \omega_i^k$ with $T_i^k \in U_i \left(DT_i^k \right)$, $\omega_i^k \sim p_\omega \left(DT_i^k, T_i^k \right)$ and compute:

$$n_i \left(DT_i^k, T_i^k \right) = n_i \left(DT_i^k, T_i^k \right) + 1$$

$$\gamma_i^k = {}^{1}\!\Big/_{n_i \left(DT_i^k, T_i^k \right)}$$

For DT_i^k, T_i^k compute:

$$
\begin{aligned}
&Q_i^k \left(DT_i^k, T_i^k \right) = \left(1 - \gamma_i^k \right) Q_i^{k-1} \left(DT_i^k, T_i^k \right) \\
&+ \gamma_i^k \left(\begin{array}{l} g_i \left(DT_i^k, T_i^k, \omega_i^k \right) + \\ \alpha \min\limits_{T \in U_{i+1}\left(f_i\left(DT_i^k, T_i^k, \omega_i^k \right) \right)} Q_{i+1}^k \left(f_i \left(DT_i^k, T_i^k, \omega_i^k \right), T \right) \end{array} \right) \\
&Q_{I+1}^k \left(DT_{I+1}, T_{I+1} \right) = \left(1 - \gamma_I^k \right) Q_I^{k-1} \left(DT_I^k, T_I^k \right) \\
&+ \gamma_I^k g_I \left(DT_I^k, T_I^k, \omega_I^k \right)
\end{aligned}
$$

For any other $\left(DT_i, T_i \right) \ne \left(DT_i^k, T_i^k \right)$, if $n_i \left(DT_i, T_i \right) \ne 0$ compute $Q_i^k \left(DT_i^k, T_i^k \right) = Q_i^{k-1} \left(DT_i^k, T_i^k \right)$

Note that the Q-factor $Q_i^k \left(DT_i, T_i \right)$ is not defined when $n_i \left(DT_i, T_i \right) = 0$.

Step 2. If $k > 1$ compute the convergence error:

$$v_k = \sqrt{\sum_{i/n_i\left(DT_i^k,T_i^k\right)\neq 0}\left(Q_i^k\left(DT_i^k,T_i^k\right)-Q_i^{k-1}\left(DT_i^k,T_i^k\right)\right)^2 + \sum_{i/n_i\left(DT_i^k,T_i^k\right)=0}\left(Q_i^k\left(DT_i^k,T_i^k\right)\right)^2}$$

Step 3. If $v_k \leq \varepsilon$, where ε is a given tolerance, stop. Otherwise go to Step 1.

Algorithm 1 Look-up table algorithm

Initially, the size of the problem increases on the order $O(\#DT\cdot\#T\cdot I)$, where I is the number of stations (stages), and $\#DT,\#T$ are the number of discrete departure times and travel times considered in the discretization. Note that the departure time and travel time is in general a vector of the departure and travel times of all trains at a given station, so the number of points considered increases as a power of J (number of trains).

B. Continuous approximation using basis functions algorithm

The main disadvantage of the discretization strategy is that the size of the problem explodes when the number of trains or stations increases. Furthermore, since every point visited adds only local information about the Q-factor function, an increasingly large number of iterations is required to converge to the solution within a given error bound.

The idea of this algorithm is to approximate the Q-factor function using predefined basis functions:

$$Q_i^*\left(DT_i,T_i\right)\approx \overline{Q}_i\left(DT_i,T_i,r_i\right)= \sum_k r_{ik}\Phi_{ik}\left(DT_i,T_i\right) \tag{16}$$

Usually, polynomial basis functions are used. Nonetheless, the algorithm works for any other choice of the basis functions.

After that, an infinitely long succession of possible departure times and total travel times for every station i, DT_i^k,T_i^k, $k=1,\ldots$ is generated using a pure exploration strategy; i.e., sampled from the probability distribution of the problem. Given the pair $\left(DT_i^k,T_i^k\right)$, a disturbance ω_i^k is sampled according to the probability $p_\omega\left(DT_i^k,T_i^k\right)$. Every sample contributes information about the global shape of the function. The coefficients, r, of the basis functions are iteratively updated using a stochastic gradient method; $\forall i<I,\forall k$:

$$\begin{aligned} d\left(DT_i^k,T_i^k,\omega_i^k\right)&= g_i\left(DT_i^k,T_i^k,\omega_i^k\right) \\ &-\sum_r r_{ir}^{k-1}\Phi_{ir}\left(DT_i^k,T_i^k\right) \\ &+\min_{\overline{T}\in U_{i+1}\left(f_i\left(DT_i^k,T_i^k,\omega_i^k\right)\right)}\sum_r r_{i+1r}^{k-1}\Phi_{i+1r}\left(f_i\left(DT_i^k,T_i^k,\omega_i^k\right),\overline{T}\right) \end{aligned} \tag{17}$$

$$r_{ir}^k = r_{ir}^{k-1}+\gamma \mathop{E}_{\omega_i}\left[d\left(DT_i^k,T_i^k,\omega_i^k\right)\right]\Phi_{ir}\left(DT_i^k,T_i^k\right)$$

where $\gamma = O\left(\frac{1}{k}\right)$ (step size), and k is the number of iterations.

Step 0. Initialize: Set $k=0, r_{ir}^0=0$

Step 1. Set $k=k+1,\gamma = 1/k$. Then for all station i, generate possible DT_i^k, T_i^k, ω_i^k with $T_i^k\in U_i\left(DT_i^k\right)$, $\omega_i^k \sim p_\omega\left(DT_i^k,T_i^k\right)$ and compute:

$$\begin{aligned} d\left(DT_i^k,T_i^k,\omega_i^k\right)&= g_i\left(DT_i^k,T_i^k,\omega_i^k\right) \\ &-\sum_r r_{ir}^{k-1}\Phi_{ir}\left(DT_i^k,T_i^k\right) \\ &+\min_{\overline{T}\in U_{i+1}\left(f_i\left(DT_i^k,T_i^k,\omega_i^k\right)\right)}\sum_r r_{i+1r}^{k-1}\Phi_{i+1r}\left(f_i\left(DT_i^k,T_i^k,\omega_i^k\right),\overline{T}\right) \end{aligned}$$

$$r_{ir}^k = r_{ir}^{k-1}+\gamma \mathop{E}_{\omega_i}\left[d\left(DT_i^k,T_i^k,\omega_i^k\right)\right]\Phi_{ir}\left(DT_i^k,T_i^k\right)$$

Step 2. If $k>1$ compute the convergence error:

$$v_k = \left\|r^k-r^{k-1}\right\|_2 = \sqrt{\sum_{i,r}\left(r_{ir}^k-r_{ir}^{k-1}\right)^2}$$

Step 3. If $v_k \leq \varepsilon$, where ε is a given tolerance, stop. Otherwise go to Step 1.

Algorithm 2 Basis function algorithm

Using this algorithm, the size of the problem increases on the order $O(I\cdot R)$, where R is the number of basis functions considered. In general, the number of basis functions increases linearly with the number of trains. Because every sample adds global information about the Q-factor value, convergence is expected in fewer iterations than the look-up table approach (we use the data to update the coefficient of the basis functions in the whole feasible region, whereas we update only determined look-up table entries with a look-up table algorithm).

IV. Results

In this section we present the main results obtained by applying the preceding algorithms to solve both example problems. The first subsection presents the solution for the timetable design minimizing energy consumption with punctuality constraints. The second subsection presents the solution for the problem of maximizing capacity subject to safety constraints. For both examples, we compare the results and the speed of convergence using the three algorithms described above. We also use both examples to demonstrate the intuition that can be obtained by visualizing the Q-factor function surface. Finally, we explore the behavior of the algorithms as the size of the problem increases. For that purpose, we have scaled up to a larger problem size based on the second example.

A. Minimizing energy consumption with punctuality constraints

Although we have insisted on the importance of time slack to ensure good operational punctuality levels, determining an efficient allocation of the time slack is a difficult task. Typically, time slack is allocated homogeneously along the line. However, the solutions obtained using our approach have several advantages compared to the traditional, homogeneous slack distribution solution.

It is important to determine the minimum slack required to ensure the desired punctuality level. It is possible that the homogeneous slack distribution solution does not meet the punctuality constraints at all stations. Once the minimum slack has been determined, our solution allocates the remaining slack time at those inter-stations with higher energy saving potential. If no delay appears, these time slacks are used for eco-driving purposes. The results obtained for this example with four stations and 10% slack show that a reduction of energy consumption of 1.7% for each train could be obtained in comparison with a homogeneous slack distribution. We can achieve higher reductions in energy consumption designing timetable for trips with more intermediate stops. This energy consumption reduction also increases for larger time slacks.

Table 1 shows the minimum and maximum travel times along the inter-stations for a realistic example of a high speed railway line with four stations:

Data	Travel Time [time units]		
	Inter-station 1	Inter-station 2	Inter-station 3
Minimum	59	106	64
Maximum	75	120	78

Table 1 Problem parameters, minimum and maximum travel time

A total slack of 15%, equivalent to 34 time units is considered. We require 90% punctuality level, that is, 90% of the trains have to arrive with no delay, i.e., must be able to recover any unexpected delay that appears along the line.

Table 2 shows the timetables obtained using a look-up factor algorithm with different number of iterations and a MIP model solved using CPLEX:

Method	Departure Time [time units]				Cost (o.f.) [kWh]
	Station 1	Station 2	Station 3	Station 4	
ADP L-U Table (100 iter.)	0	75	190	263	3706.9
ADP L-U Table (250 iter.)	0	72	188	263	3653.9
ADP L-U Table (500 iter.)	0	70	189	263	3529.6
ADP L-U Table (1000 iter.)	0	68	190	263	3455.2
ADP L-U Table (2000 iter.)	0	68	190	263	3455.2
MIP (to optimality)	0	68	190	263	3455.2

Table 2 Comparison of the timetables designed for different number of iterations of a look-up table algorithm with the optimal timetable

Furthermore, other output of the ADP algorithms proposed in this paper is the Q-factor function, which provides insight into the problem. Figure 3 presents the second inter-station Q-factor function. It is possible to use that function to visualize the constraints of the problem. In this figure, zero travel time means that the train is travelling from station 2 to station 3 at the maximum possible speed (minimum travel time). So, in particular, the minimum travel time to ensure punctuality can be determined using these algorithms; 8 time units in this case. Furthermore, the Q-factor function provides information about the maximum possible travel time for the inter-station; 16 time units. Notice that it is not possible to travel always at the minimum speed. When the departure time is later (because the train has already spent some time at previous inter-stations), the travel time must be bounded to ensure that the trains arrive within the maximum slack (34 time units in this case), after ensuring punctuality in the next stations (where they need at least 4 time units). The minimum departure time (9 time units) indicates the slack required to ensure punctuality in previous stations. The energy consumption has been modeled as a piecewise linear time-consumption Pareto curve. The Q-factor function presents the lowest possible energy consumption for the train from station 2 to station 4, if it departures at a given instant from station 2 and arrives to station 3 after some given travel time. It can be seen that it is better to travel as slow as possible in this station (in this example, it is the station with higher energy saving potential). The value of the Q-factor function for different departure times (lower values for earlier departure times) indicates that it is better to use the remaining slack at the third inter-station. The points with zero Q-factor value are infeasible.

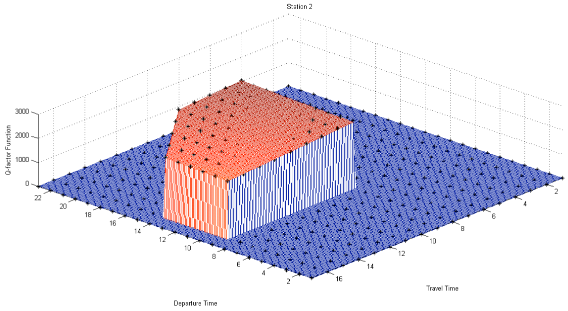


Figure 3 Q-factor function obtained with look-up table and basis functions

Figures 4 to 6 show the impact of changing the total time slack in the Q-factor function for Station 2, and hence in the solution obtained. For instance, the value of the minimum Q-factor decreases for greater slacks, since the possibility for doing eco-driving increases.

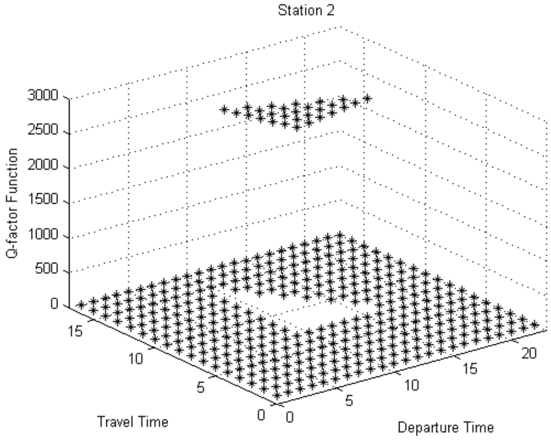


Figure 4 Q-factor function for a total slack of 12% of minimum travel time

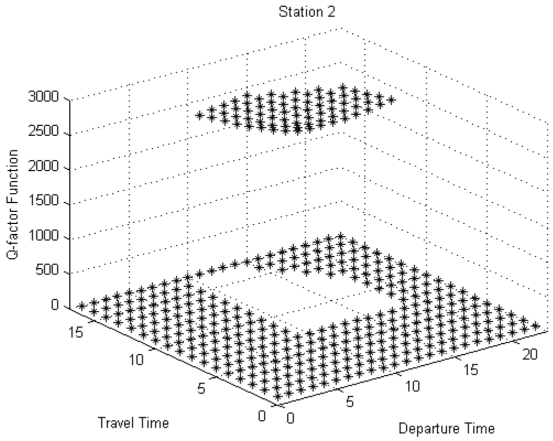


Figure 5 Q-factor function for a total slack of 15% of minimum travel time

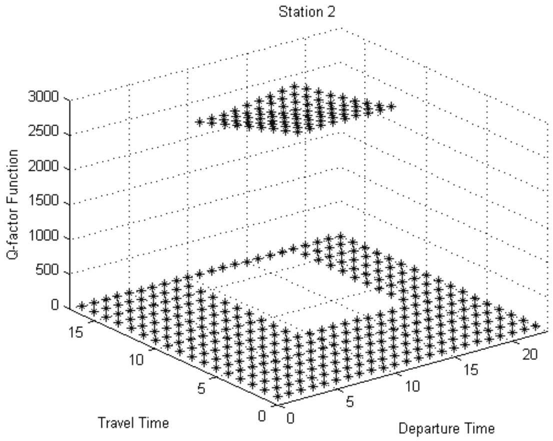


Figure 6 Q-factor function for slacks amounting to 18% of minimum travel time

In general, the effect of allocating time slacks at different inter-stations is not the same. For small total slack, the best strategy is to locate as much slack as possible where the potential of energy savings is greater. As the available slack increases, the strategy should order the energy savings impact of each incremental time slack allocation, and it is usually most efficient to allocate time slack across different inter-stations.

The effect of increasing the total time slack can also be seen in the size of the feasible region. In figures 4 and 5, possible travel and departure time are determined by the maximum slack. In figure 6, maximum departure time is sometimes determined by the maximum possible travel time at inter-station 1 and the maximum travel time may be attained too. In figure 7, there is no effect in terms of feasible region of the total time slack constraint.

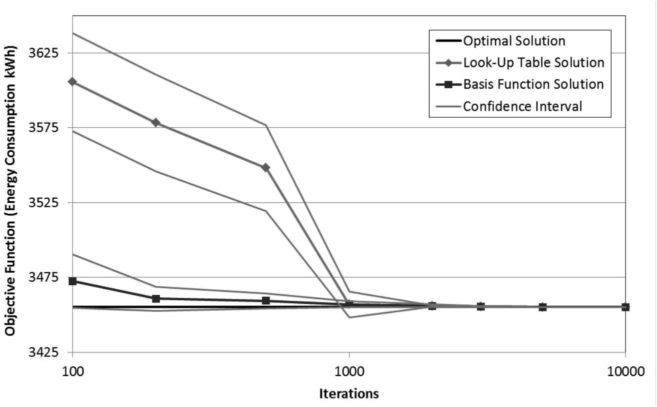


Figure 7 Convergence of look-up table and first order basis functions algorithm for 15% total slack (20 repetitions)

The convergence results for the algorithms are shown in figures 7 and 8. Figure 7 shows the mean solutions obtained for different number of iterations. Each experiment has been repeated 20 times to determine a 90% confidence interval for the solutions obtained. Note that both algorithms are converging to the optimal solution. We are using a logarithmic scale in the x axis. Figure 8 presents the difference between consecutive Q-factor matrices for the look-up table algorithm or coefficient vectors for the basis function algorithm. This measure is used as the convergence criterion.

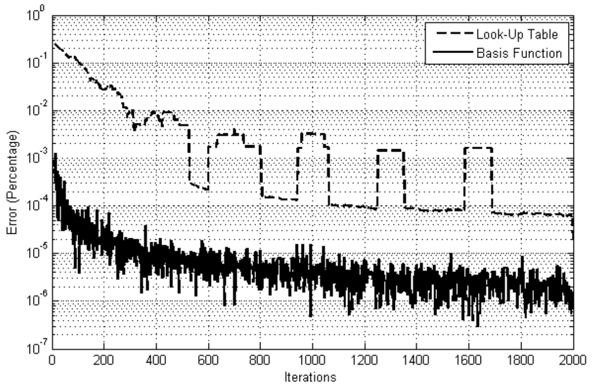
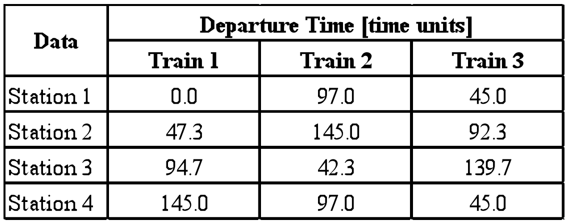


Figure 8 Convergence of look-up table and first order basis functions algorithms for 15% total slack

Figure 8 illustrates that the error when using basis functions falls more rapidly than the error obtained using a look-up table algorithm. Moreover, the computational cost of each iteration is smaller for basis function algorithms. For this application, the basis function approach has clear advantages, since the algorithms take advantage of the information obtained at each iteration. Note that we use a y-axis logarithmic scale.

B. Maximize used capacity subject to safety constraints

In this example, the objective is to maximize the use of the capacity of the line, which is equivalent to minimizing the period (time required) to schedule a given number of trains at the line. In particular, we consider three trains running in the same direction along a three stations line (we consider a four station which represents the second time that the trains arrive to the initial station). The trains need 145 time units to complete the loop (period). In table 3 we present the optimal timetable obtained using a linear basis function algorithm:



Data	Departure Time [time units]		
	Train 1	Train 2	Train 3
Station 1	0.0	97.0	45.0
Station 2	47.3	145.0	92.3
Station 3	94.7	42.3	139.7
Station 4	145.0	97.0	45.0

Table 3 Optimal timetable maximizing used capacity with safety constraints

Note that the departure timed from station 4 are equivalent (adding the period for train 1) to those of station 1, which makes it possible to repeat the timetable after one period.

The Q-factor functions for one of the trains at each station obtained using a look-up table algorithm, and two global basis function (linear and quadratic) algorithms are presented in figures 9 to 12. Notice that the departure time is fixed at the first station (time 0), but there are several possible departure times at the second station (depending on the travel time chosen at the previous station), and more possible departure times at the following stations (depending on the combination of travel times and departure times at previous stations). Furthermore, the magnitude of the Q-factor function decreases from one station to the next, because it is a measure of the cost-to-go; i.e., the remaining period to arrive to the last station given the current situation (state) and the next decision (control). The values of the minimum and maximum possible travel times aggregate the impact of the constraints, such as the interference with other trains.

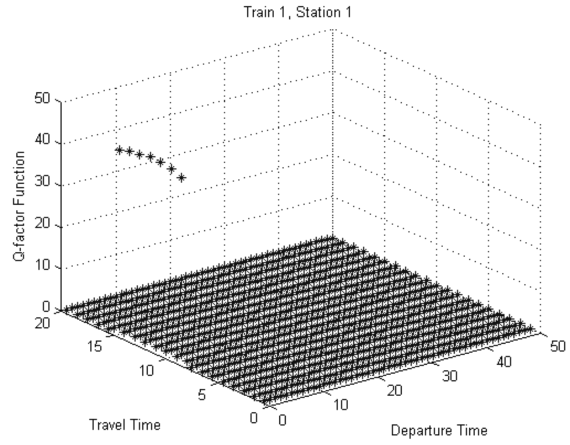


Figure 9 Q-factor function for train 1 at station 1 after 10^4 iterations

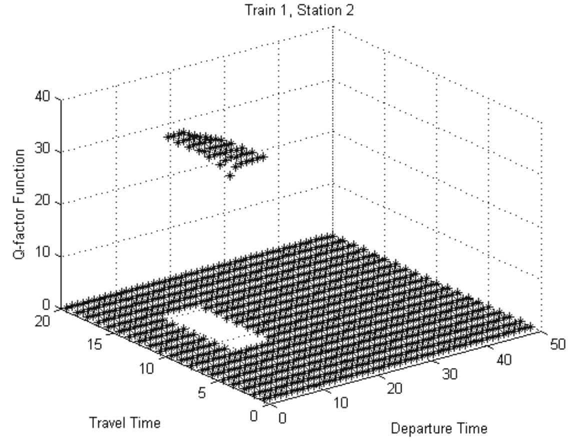


Figure 10 Q-factor function for train 1 at station 2 after 10^4 iterations

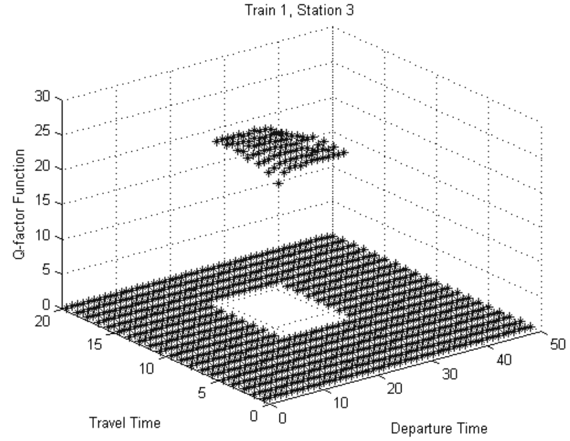


Figure 11 Q-factor function for train 1 at station 3 after 10^4 iterations

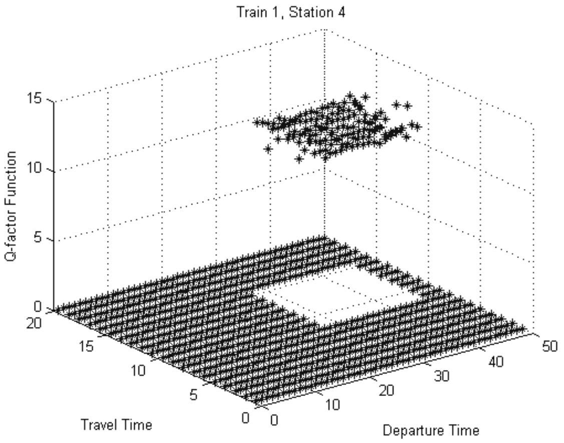


Figure 12 Q-factor function for train 1 at station 4 after 10^4 iterations

In this example, the optimal strategy is to make the trains travel as fast as possible along the stations. In general, the determination of this maximum speed is non-obvious, but can be easily determined from the Q-factor functions.

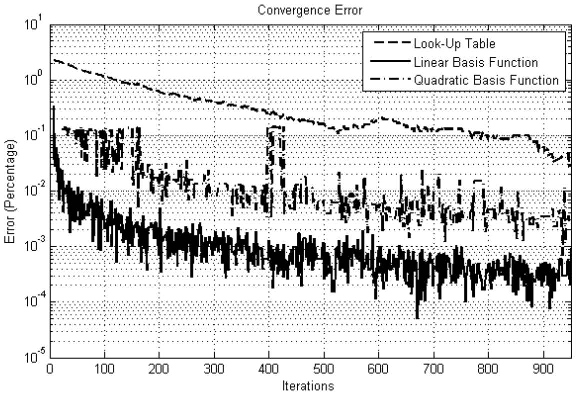


Figure 13 Convergence of look-up table and first and second order basis functions algorithms

Figure 13 shows the convergence of three alternative algorithms for this example. Although the shape of the Q-factor function does not seem to be linear, the linear basis function algorithms results in smaller errors than those obtained with the quadratic basis function algorithm. The look-up table algorithm is also converging well, but the error of a solution obtained with a given number of iterations is usually bigger than the error of the solutions obtained within the same number of iterations using the other methods.

C. Size of the model

One of the main advantages of using ADP algorithms to solve these problems is that the computational effort required to solve large problems is a polynomial function of the size of the problem. For instance, the computational effort required to design timetables maximizing the capacity subject to safety constraints, using a linear basis function approach, for different numbers of stations and trains, will increase at $O(I \cdot R)$. In this case, I is the number of stations (stages) and R, the number of coefficients, is equal to $3 \cdot J$ (a constant coefficient, the coefficient of the departure time, and the coefficient of the travel time for each train):

$$\overline{Q}_i\left(DT_i\ ,T_i\ ,r_i\right) = \left(r_{1ij} + r_{2ij} \cdot DT_{ij} + r_{3ij} \cdot T_{ij}\right)_{j=1,\dots,J} \qquad (22)$$

Figure 14 show the iterations required to design timetables maximizing the capacity subject to safety constraints for different number of stations and trains. For this experiment we consider that the problem has converged when the convergence error (gap) is smaller than 5%.

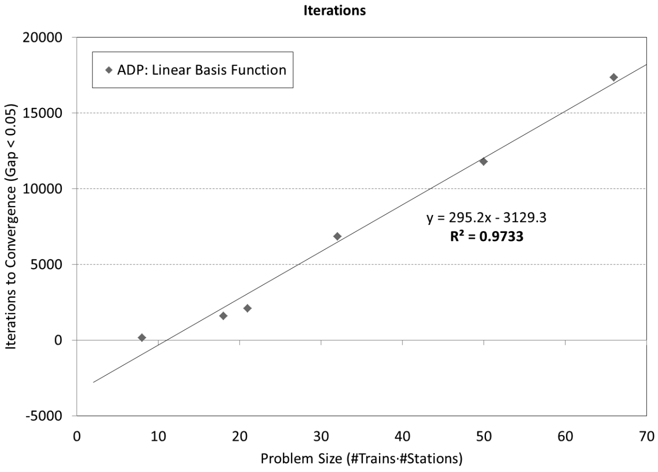


Figure 14 Number of iterations to convergence to the solution within a 5% gap for different problem sizes using a linear basis function algorithm

The number of iterations required increases linearly (the Pearson coefficient is 97.33%) with the product of the number of trains and stations as expected. This is a key result of this paper. Figure 15 compares the number of iterations required to solve the problem within a 5% gap using the ADP linear basis function algorithm proposed in this paper and a MIP model, solving an implementation of a multi-mode resource constrained project scheduling model (Castillo et al., 2009), using CPLEX. As discussed in the introduction, the number of iterations required using MIP models increases as an exponential function of the size of the problem. This result, along with other advantages already discussed, shows the convenience of using of ADP algorithms for solving train timetabling problems.

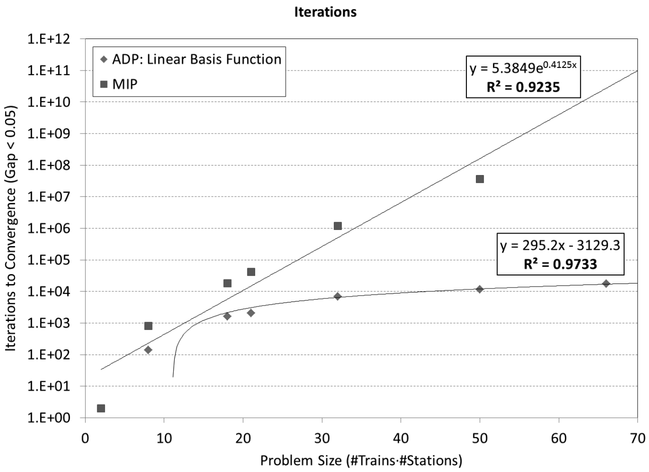


Figure 15 Comparison of the number of iterations required to convergence (within 5% gap) for an ADP linear basis function algorithm and a MIP model

V. Conclusions

In this paper, a DP formulation of the train timetabling problem has been presented. The resulting model is, in general, highly multidimensional, but can be solved tractably using several ADP techniques, even when the size of the model increases.

In particular, we have demonstrated two Q-learning algorithms using different function approximation techniques: a look-up table algorithm, which is converging very well, though the size of the look-up table grows exponentially with the size of the problem; and a basis function algorithm, which presents very good and fast convergence to the optimal solution of the problem. The results show that for large problem sizes, the ADP basis function algorithm, is specially promising compared to other MIP models proposed in the literature.

An additional benefit to this approach is that the Q-factor function determined using these algorithms provides the user with complete information about the problem. The value of the Q-factor function is not only that we can design the optimal timetable for a specific problem; but moreover, we can identify the binding constraints and good strategies to design train timetables. Sensitivity analyses should be carried out to obtain the same information using other optimization approaches to design the timetables.

In summary, we argue that the DP approach, and in particular, the ADP algorithms proposed in this paper, should be considered as a promising approach for solving train timetabling problems, especially for large rail systems or for those problems where the railroad company is more interested in what-if answers than in a specific rigid optimal timetable.

Further lines of research will analyze whether we can improve the convergence of the algorithms proposed using intelligent sampling techniques. We will also study the application of the algorithms developed in this paper for solving other railway applications.

Aknowledgement

The research presented in this paper has been sponsored by La Caixa Foundation. An earlier version of this paper was presented at the Computational Management Science Conference (CMS 2011) in Neuchatel, Switzerland, in April 2011. We would like to thank the attendants of the session for their suggestions and comments.

References

Abril, M., Barber, F., Ingolotti, L., Salido, M. A., Tormos, P., & Lova, A. (2008). An assessment of railway capacity. *Transportation Research: Part E: Logistics and Transportation Review, 44*(5), 774-806.

Assad, A. A. (1981). Analytical models in rail transportation: An annotated bibliography. *INFOR.Canadian Journal of Operational Research and Information Processing, 19*(1), 59-80.

Bertsekas, D. P. (2005a). *Dynamic programming and optimal control* (III ed.), Cambridge, Massachusetts: Athena Scientific.

Bertsekas, D. P. (2005b). *Dynamic programming and optimal control* (III ed.), Cambridge, Massachusetts: Athena Scientific.

Bertsekas, D. P., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Cambridge, Massachusetts: Athena Scientific.

Bouzaiene-Ayari, B., & Powell, W. B. (2010). An approximate dynamic model for locomotive optimization. (Railway Applications)

Caimi, G., Burkolter, D., Herrmann, T., Chudak, F., & Laumanns, M. (2009). Design of a railway scheduling model for dense services. *Networks and Spatial Economics, 9*, 25-46.

Caimi, G., Chudak, F., Fuchsberger, M., Laumanns, M., & Zenklusen, R. (2010 (in press)). A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transportation Science, Articles in Advance*, 1-16.

Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research, 50*(5), 851-861.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., & Toth, P. (2007). Passenger railway optimization. In Barnhart, C. and Laporte, G. (Ed.), *Handbooks in operations research and management science* (pp. 129-187) Elsevier.

Capros, P., Mantzos, L., Papandreou, V., & Tasios, N. (2007). *European energy and transports. trends to 2030.* Belgium: European Comission and Dictorate-General for Energy and Transports.

Castillo, E., Gallego, I., Ureña, J., & Coronado, J. (2009). Timetabling optimization of a single railway track line with sensitivity analysis. *TOP, 17*(2), 256-287.

Cordeau, J. F., Toth, P., & Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science, 32*(4), 380-404.

Crainic, T. G. (2003). Long-haul freight transportation. In R. W. Hall (Ed.), *Handbook of transportation science* (pp. 451-516) Second edition; International Series in Operations Research and Management Science; Boston; Dordrecht and London; Kluwer Academic.

Fernández, A., Cucala, P., Vitoriano, B., & de Cuadra, F. (2006). Predictive traffic regulation for metro loop lines based on quadratic programming. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 220*(2), 79-89.

Ghoseiri, K., Szidarovszky, F., & Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. *Transportation Research, Part B: Methodological, 38*(10), 927-952.

Kraay, D. R., & Harker, P. T. (1995). Real-time scheduling of freight railroads. *Transportation Research, Part B: Methodological, 29B*(3), 213-229.

Lai, Y. C., & Barkan, C. (2009). Enhanced parametric railway capacity evaluation tool. *Transportation Research, Record, 2117*(1), 33-40.

Landex, A. (2009). Evaluation of railway networks with single track operation using the UIC 406 capacity method. *Networks and Spatial Economics, 9*(1), 7-23.

Lee, C. Y., & Pinedo, M. (2002). Optimization and heuristics in scheduling. In P. M. Pardalos, & M. G. C. Resende (Eds.), *Handbook of applied optimization* (pp. 569-584). New York: Oxford University Press.

Liebchen, C., Proksch, M., & Wagner, F. H. (2004). Performance of algorithms for periodic timetable optimization. San Diego, California, USA.

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Renfe. (2010). *Informe Anual 2009 Renfe.* http://www.renfe.com/docs/informe_anual2009.pdf

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics, 22*(3), 400-407.

Zhou, X., & Zhong, M. (2005). Bicriteria train scheduling for high-speed passenger railroad planning applications. *European Journal of Operational Research, 167*(3), 752-771.

Zhou, X., & Zhong, M. (2007). Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research, Part B: Methodological, 41*(3), 320-341.