DTU Library

Construction of Discrete Time Graphs from Real Valued Railway Line Data

Harrod, Steven

Published in:
New Trends in Emerging Complex Real Life Problems

Link to article, DOI:
10.1007/978-3-030-00473-6_32

Publication date:
2018

Document Version
Peer reviewed version

Construction of Discrete Time Graphs from Real Valued Railway Line Data

Steven Harrod

Steven Harrod
Technical University of Denmark, Kgs. Lyngby, 2800 Denmark, e-mail: stehar@dtu.dk

Chapter 1

Construction of Discrete Time Graphs from Real Valued Railway Line Data

Abstract Railway timetables are frequently modeled as discrete time expanded graphs. The selection of the magnitude of the discrete time unit can significantly alter the structure of the graph and change the solutions generated. This paper presents a method for generating improved mappings of real railway track segments to discrete arc graphs given a chosen discrete time unit. The results show that the dimensions of the generated graph are not monotonic and a range of values should be evaluated.

Key words: Railway Timetable, Discrete Optimization, Railway Operations

1 Introduction

Frequently, railway timetabling problems are formulated as discrete time expanded graphs. The movements of the trains however, are measured in real valued time. In most formulations, feasible solutions require that train run times be lengthened or rounded up to the nearest discrete time unit, resulting in some increase in travel time and reduction in railway line capacity. It should be explained to the reader new to railways, that nearly all railways divide their rail networks into sections called "blocks". Train movement authorization is given according to these blocks, and a true microscopic model of a railway would represent each of these blocks as an arc. These blocks can be as small as 100 meters.

The number of alternative train paths, the density of the graph, and the complexity of the problem all increase as the size of the discrete time unit shrinks. Frequently, these timetabling problems are very large, consisting of tens of thousands of discrete arcs, each of which is represented in a mathematical program by a binary decision variable. One way to reduce the complexity of these problems is to select a larger discrete time unit, with a subsequent increase in the model approximation error. This paper demonstrates a method of optimizing the discrete arc graph for a given time unit magnitude.

The design of these discrete arc graphs is one of many tasks in a class of problems variously referred to as “train routing” problems or “timetabling problems” (TTP). Harrod (2012) provides a detailed survey on mathematical models of this class. Various prior papers in the literature have selected a discrete time unit according to environmental conditions, business rules, or their judgment, but, with one exception, the choice is not discussed at length and usually is limited to a single sentence. Examples of studies that apply a discrete time unit are Mills et al (1991), Brännlund et al (1998), Caprara et al (2002), Şahin (2006), Harrod (2011), Lusby et al (2011).

Caimi et al (2009) presents a problem of the Berne, Switzerland station area. The paper describes assigning run times to trains as the ceiling function of the run time divided by the discrete time unit. A range of discrete time values between 15 and 120 seconds are tested on one problem scenario, and the results are shown in Figure 1. The tradeoff between computation time and accuracy can be clearly seen. “Addl. Run Time” is the average additional movement time for each train path through the station due to the rounding up of the real valued run time to discrete time. The nominal run time through the station is 250 seconds. The bounding lines “RT+/-s.d.” are drawn one half of the standard deviation from the “Addl. Run Time” value. In this case, the selected discrete time unit of 90 seconds is approximately the headway between trains, minimizes the computation time, and is approximately in the midrange of the induced error in run times and capacity.

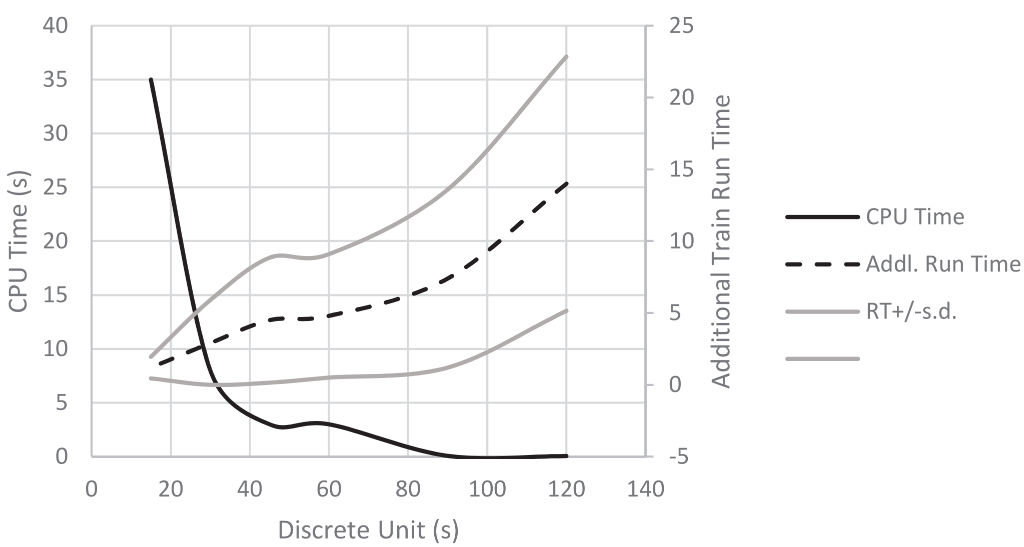


Fig. 1 Performance of Range of Discrete Time Units in Caimi et al (2009)

2 A Method for Constructing a Discrete Time Network

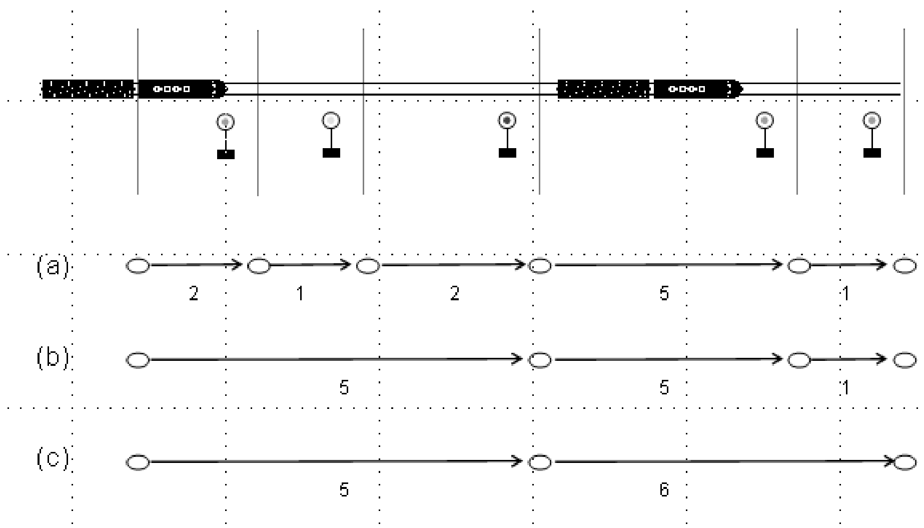
This section proposes a method for distilling a complex real-valued railway network structure into a smaller, discrete valued graph. Given a discrete time unit magnitude as a starting point, the method generates a discrete graph by merging adjacent track segments into longer segments to be represented by the graph arcs. The objective of the process is to minimize the deviation between the real valued travel time on the arcs and the assigned discrete travel time on the arcs.

The error created by the difference between the assigned integer time value and the original real valued movement time is called here "induced" model error. The chosen time unit either needs to minimize induced error over the average of all trains, or over a favored group of trains based on some objective criteria. Not all induced error is bad. Federal Railroad Administration (2005) recommends that all simulated train times be increased by 7% to compensate for operating delays not accounted for in train simulations, and some additional induced error may also serve as schedule slack to protect against stochastic delays.

There are two, sometimes opposing, objectives in this method. First, to minimize the number of blocks $|B|$ so that the number of arcs or decision variables in the model is minimized. Second, to dimension the blocks so that the resulting discrete train travel time is a close match to the real valued train time and minimizes induced error. A number of parameters support this process, such as s, the minimum number of track segments to combine, which can define a default train separation or headway. If, for example, the track segments represent signalled track segments, and the rules dictate a two block separation (which is very common), then $s = 3$ will ensure a set of blocks B which will maintain this headway. Another factor to consider in merging track segments is the rolling minimum operating headway. Track segments should not be combined in such a way that they create large variation in the arc travel times, and thus create a bottleneck and reduce the overall flow rate of the line.

Figure 2 displays an example set of track blocks to be combined. The top picture depicts a length of track with signals and two occupying trains, separated by red and yellow signals. The source data is represented below by arcs (a) with travel times labeled. The bottleneck on this route is the segment with travel time $t = 5$, so the maximum flow on this route is one train each 5 time units (because only one train may occupy each segment at a time). The middle set of arcs (b) shows the effect of combining the first three track segments. The flow is not affected. The bottom set of arcs (c) makes a poor choice in combining the last two track segments and reduces the maximum flow to one train each 6 time units.

The combination of track segments into model track blocks is determined by problem (P), which is a simple set partitioning exercise. Refer to Table 1 for explanation of the set notation. The first component of the objective is a tie-breaker. In the event that more than one combination of track blocks offers the same objective value, the one with the least number of members is preferred. The selection of a coefficient of 0.001 is arbitrary, within a range. It should be small to insure that the second component of the objective is the dominant decision maker, but it should not



Numbers are segment travel times.

Fig. 2 Example Track Segment Combination Process

be too small, as extreme magnitude differences in coefficients can lead to numerical difficulties for integer program solvers (Camm et al, 1990).

The second component of the objective is the sum of the induced error for the source train data considered. All real valued train times are rounded up to the next integer discrete time value, thus assuring operational feasibility of the resulting model solution timings. The induced error α is the difference of the real value equivalent of the discrete time value and the source real valued train time. The error is then approximately the discrete time unit (μ) minus the modulo of the real time (t) divided by the time unit (Equation 1).

$$\mu - (t \mod \mu) \tag{1}$$

All combinations of track segments are enumerated in set Ω, whose members are a couple (i,k) where i is the number of track segments combined and k is the first track segment index. For example, set member $(3,5)$ defines the combination of track segments $\{5,6,7\}$, set member $(4,3)$ defines the combination of track segments $\{3,4,5,6\}$, set member $(5,8)$ defines the combination of track segments $\{8,9,10,11,12\}$, etc. The size of the combination of track segments is limited by s, which may determine a minimum physical headway, a user selected maximum combination size l, and the maximum of the rolling minimum operating headway or bottleneck time described earlier. The single constraint requires that any solution cover all source track segments. This problem should be solved for a wide range of values of u (the discrete time unit value, see Table 1), and then the total induced error and number of model track blocks $|B|$ calculated and compared for each value prior to making a final selection.

Table 1 Components of Problem 2

Component	Type	Description
$x_{i,k}$	binary variable	Represents the selection of segment set (i,k) as a model network track block
u	parameter	Real value of discrete time unit
$t_{r,i,k}$	parameter	Total real travel time for train r on segment set (i,k)
$\alpha_{r,i,k}$	parameter	Induced error for train r on segment set (i,k) for given discrete time unit, $\alpha_{r,i,k} = u - mod(t_{r,i,k}, u)$
h	parameter	Ruling minimum headway or maximum of rolling value of minimum operating headway
s	parameter	Minimum number of track segments to combine into one model track block
l	parameter	Maximum number of track segments to combine into one model track block
Γ	set	The set of trains considered
Θ	set	The *ordered* set of source track segments or blocks, ordered by network sequence (original data)
Ω	set	The set of possible derived blocks, each member, (i,k), maps to a contiguous subset of Θ, starting at position k in set Θ and including i consecutive members (track segments) $\Omega = \left\{ i \in \{s..l\}, k \in \{1..\lvert\Theta\rvert\} \middle\vert k \leq (\lvert\Theta\rvert - i + 1), \max_r t_{r,i,k} \leq h \right\}$
Δ_θ	set	The set of derived or merged blocks from Ω that contain the indicated original source track segment θ $\Delta_\theta = \{(i,k) \in \Omega \vert k \leq \theta < k+i\}$

(P)

$$\min \sum_{(i,k)\in\Omega} 0.001 x_{i,k} + \sum_{r\in\Gamma,(i,k)\in\Omega} \alpha_{r,i,k} x_{i,k} \tag{2}$$

s.t.

$$\sum_{(i,k)\in\Delta_\theta} x_{i,k} = 1 \qquad \forall\theta \in \Theta \tag{3}$$

$$x \in \{0,1\}$$

To date, the author has solved these directly using commercial solvers such as Cplex without difficulty. What is more difficult is managing the input data for these problems. A formal database structure is valuable for managing this data and the logical relationships between various data entries. The initial objective is to format the track network into a series of minimum dimension units for pre-processing. Each unit has dimensions of start location, end location, capacity (number of tracks), connected track segments at each end, and is tabulated as a record in the database. A suitable database for these records is displayed in Figure 3. An initial impulse might be to record the track in one mile or kilometer units, but most signal systems do not allow the control of trains in these increments, so other dimensions or division points should be used. Most North American railways operate a fixed installation signalling system, with train control points fixed at the location of color light or position signals, so the locations of these signals are the best starting points for track segment records.

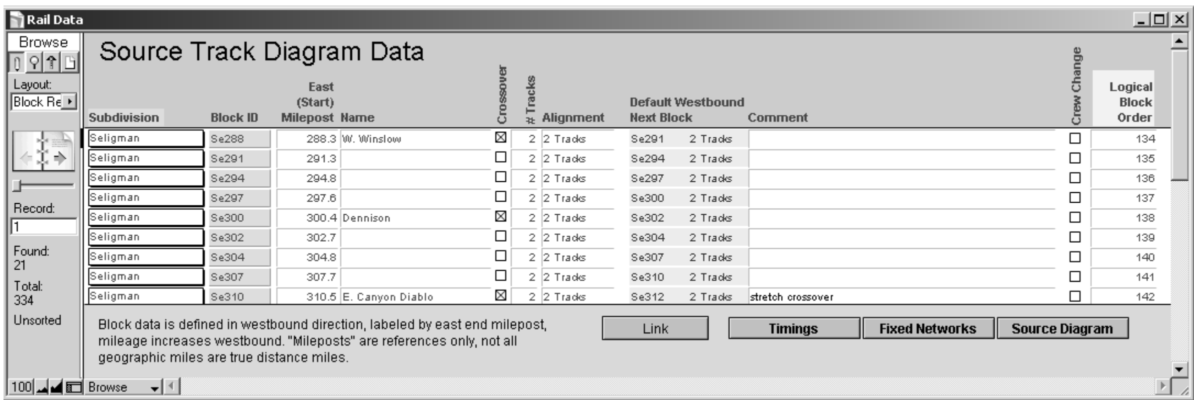


Fig. 3 Track Network Database

3 Application to a Real Test Case: the BNSF Transcontinental Railway, Winslow to Flagstaff, Arizona

This double track railway is at the midpoint of the journey between Chicago and Los Angeles, and is the dominant traffic lane for BNSF. In addition to the heaviest freight traffic on the BNSF network, the line hosts one daily passenger train in each direction, Amtrak's Southwest Chief between Chicago and Los Angeles. Winslow is a crew change point, and both Winslow and Flagstaff are station stops for the Southwest Chief. Between them lies 54 miles of double track through remote lands and a Navajo Indian Reservation.

Track network data and train timing data are supplied by BNSF. Signals are installed on this line approximately every 2-3 miles, so with source track segments of the same length, there are approximately 21 segments between Winslow and Flagstaff (approximate depending on one's interpretation of the signaling system). Train timing data is provided as computer simulations of a variety of trains. For each train timing, data is supplied westbound (WB: Winslow to Flagstaff), eastbound (EB: Flagstaff to Winslow), and for wet and dry rail in each direction, for a total of 4 timings for each train. Wet rail timings are longer than dry rail timings, because the wet rail limits acceleration and braking, so these timings are used as the more robust of the two choices. A two block separation is presumed. That is, each train is presumed to be trailed by a red signal, a yellow signal, and finally a green signal, and so $s = 3$. An arbitrarily large value of $l = 7$ is applied.

The train types considered are limited to the G and X class freight trains and the Amtrak Southwest Chief. Calculated over a 3 track segment rolling horizon, the bottleneck time for a freight train is approximately 25 minutes and occurs westbound around Darling, Arizona (milepost 326). Eastbound the bottleneck time is only 14 minutes, and at the same location. The bottleneck time for Amtrak is only 12 minutes, at the same location, and in both directions. This mix of trains was arbitrarily chosen to demonstrate a variety of train types. The typical train mix on this line is actually much more diverse, and will vary according to season, day of week, and time of day. The methods presented here may be applied to any specific scenario.

Model network track blocks are determined using Problem 2 for discrete time units in 0.5 minute increments from 1 to 20, based upon the train timings of the dominant freight trains only. The solution statistics are presented in Figure 4. The induced error is displayed as a percentage of the real valued train timing. Again, all integer valued train timings are determined by rounding up (ceiling function) the real valued train timings, and the induced error is the sum of the integer valued train timings minus the real valued train timings. The problem complexity is estimated to be proportional to the arc count, presented as a complexity factor, $(1/u)|B|$. That is, the formulation complexity is a function of the number of geographic arcs and the granularity of the time horizon to be analyzed.

Candidate discrete time unit values are tagged with vertical lines in Figure 4. The first candidate, a discrete time unit of $u = 3$, offers a desirable induced error of 6%, but a complexity factor of 1.67. At a discrete time unit value of 3.5 minutes, not only does the error increase, but the complexity increases as well. This is because the optimal number of model blocks increases from 5 to 6 at this discrete unit size. The next candidate unit size is $u = 4.5$, which offers an error of 9% and a complexity factor of 1.11, or approximately a 33% reduction in arc count for an admittedly 50% increase in error. This net error is still below 10% and is a practical level providing some schedule slack and compensating for the difference between simulated timings and expected timings. In this data set the trade-off between problem complexity and induced error becomes increasingly less favorable as the discrete unit size increases.

The resulting train timings in discrete time units of 4.5 minutes are presented in Table 2. The Amtrak trains are faster than the general freight trains by a factor of nearly 2 (compare column Freight/WB with column Amtrak/WB in Table 2), in spite of the fact that the top allowable speed of Amtrak trains is not twice that of freight trains as a rule. The authorized passenger speed between Winslow and Flagstaff is approximately 79 mph, and for general freight it is approximately 45 mph, or a ratio of approximately 1.75. The superior acceleration and braking properties of the Amtrak train allow it to navigate the route much faster than the general freight trains. Also note how the method has homogenized the arc dimensions. Block 1 is twice as many miles as block 3, but in 3 of 4 columns it is only 50% greater in travel time.

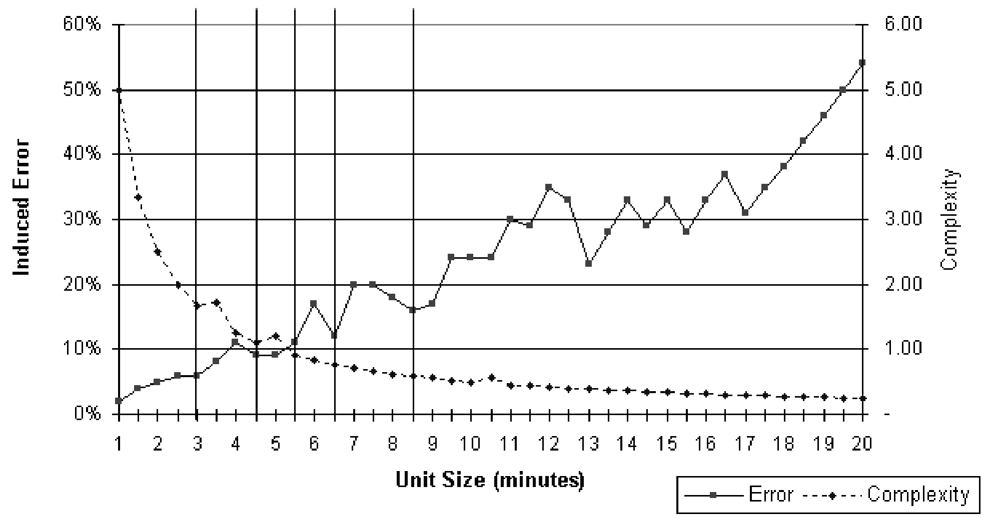


Fig. 4 Induced Error and Complexity as a Function of Discrete Time Unit

Table 2 Discrete Timings for $u = 4.5$

Milepost	Block No.	True Miles	Freight WB	Freight EB	Amtrak WB	Amtrak EB
288 (Winslow)	1	16.6	6	4	3	3
304	2	10.0	4	3	2	2
314	3	8.8	4	2	2	2
323	4	9.5	5	3	3	3
333 (Flagstaff)	5	11.3	6	4	4	4

Reflecting back to the discussed Caimi et al (2009) in the introduction, the results of this specific example demonstrate some similarities. First of all, Figures 1 and 4 display the same trends, but recall also that Caimi et al finds that the headway time between trains is a good heuristic for the preferred discrete time unit. In this example the limiting train headway on the BNSF line is not known, but it may be estimated by established methods described in Parkinson and Fisher (1996). In this case, the dominant freight traffic runs at 45 mph, a two block signal separation is expected, and the blocks are 2-3 miles long. The trains should thus typically operate with a five mile separation or about 6.66 minutes, which is a little higher than the result suggested by Figure 4, but a good initial value. The advantage of this method is that clear guidance can be obtained quickly and with direct evidence, without the necessity of actually constructing and testing alternative models.

4 Conclusion

This paper introduced the application of discrete time units in scheduling problems for railways, and cited Caimi et al (2009) as an example of the range of values that could be selected and their impact on the problem complexity and accuracy. Caimi et al obtains its results by trial and error on a complex railway scheduling problem. This paper offers a mathematical program for generating prospective time unit values specific to a given railway line and train performance. The results and application are comparable. This research could be further extended with more detailed, large problem examples.

The choice of discrete time unit must not be arbitrary, because small changes in unit size can have large effects on the model's representation of and authenticity to actual operations. The problem demonstrated here is not of a large enough size for practical application, but the limitation at the moment is not the capability of solvers such as Cplex, but the time necessary to collect and structure the data. There are approximately 119 signaled track segments between Winslow and Needles, and the westbound freight journey time is 432 minutes. Using the principles described here, this network could be abstracted to a graph of 24 track blocks and a 4 minute discrete time unit, offering approximately an 8% induced error and a complexity factor of 6.0.

The method described here provides a fast process for approximating a real valued set of sequential railway track segments as a discrete arc graph. Multiple discrete time magnitudes may be evaluated and compared on their induced error and resulting graph complexity. The progression of the graph development as the discrete time unit increases is not linear. In some cases a larger time unit offers reduced complexity without incurring larger induced error. Further investigation of this method could evaluate the robustness of the actual train timetabling solutions produced under different discrete time unit values.

References

Brännlund U, Lindberg P, Nôu A, Nilsson J (1998) Railway timetabling using lagrangian relaxation. Transportation Science 32(4):358–369

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

Camm JD, Raturi AS, Tsubakitani S (1990) Cutting big m down to size. Interfaces 20(5):61–66

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

Federal Railroad Administration (2005) Railroad corridor transportation plans. Tech. rep., Office of Railroad Development, RDV-10, Federal Railroad Administration, Washington, DC

Harrod S (2011) Modeling network transition constraints with hypergraphs. Transportation Science 45(1):81–97

Harrod S (2012) A tutorial on fundamental model structures for railway timetable optimization. Surveys in Operations Research and Management Science 17:85–96

Lusby R, Larsen J, Ryan D, Ehrgott M (2011) Routing trains through railway stations: A new set-packing approach. Transportation Science 45(2):228–245

Mills R, Perkins S, Pudney P (1991) Dynamic rescheduling of long-haul trains for improved timekeeping and energy conservation. Asia-Pacific Journal of Operational Research 8:146–165

Parkinson T, Fisher I (1996) Tcrp report 13, rail transit capacity. Tech. rep., Transportation Research Board, Washington D.C.

Şahin G (2006) New combinatorial approaches for solving railroad planning and scheduling problems. PhD thesis, University of Florida, Gainesville, FL