A COMPUTER SIMULATION OF A RAIL NETWORK

Samuel M. Graff and Peter Shenkin

John Jay College of Criminal Justice, City University of New York, Mathematics Department, 445 West 59 Street, New York, N.Y. 10019

Abstract. A computer simulation of a rail segment is presented. The goal is to provide a capability for scheduling and routing with respect to predetermined objectives. The simulation is founded on a decomposition of the given line segment into fundamental units representing node to node subsegments with each node being an interlocking of the real system. A decision subroutine is activated every time a train reaches a node; all feasible options are then examined with respect to the current configuration of the system. Ultimately, it is hoped the simulation will have on-line monitoring capabilities.

Keywords. railroad; simulation; dispatching; optimization; planning.

INTRODUCTION

The development of computer assisted dispatching models for railway systems has grown in recent years as a consequence of financial pressures on the industry. Historically, the flow of traffic over a rail system was manually controlled by the dispatcher who decided where trains traveling in opposing directions on a single track line would meet each other or where trains traveling in the same direction on either single or multiple track lines would overtake one another. Decisions were based on experience acquired through years of apprenticeship along with intimate familiarity of the geographic territory and traffic patterns. The territory assigned to any individual was uaually not more than a manageable 100 mile segment of the entire system.

In order to economize by reducint manpower and taking advantage of the vast improvement in communications between field locations and the dispatching offices, assigned territories have expanded significantly to zones exceeding 500 miles. The complexity and density of today's traffic patterns make it desirable to augment the dispatcher's decision capability. Indeed, it is more difficult to one person to keep up to date on the physical characteristics of such vast territories along with any fluctuations in traffic densities. Moreover, the centralization of control of thousands of miles of rail in one office gives rise to the possibility that an individual may be assigned to different territories on successive days. Models such as the one presented in this paper are an attempt to provide scenarios which supplement the dispatcher's options. In addition, they allow the effect of any one decision to be evaluated immediately.

Several of the largest North American carriers have developed proprietary simulations (route capacity models) for their planning and operating departments. Each is written in a simulation language such as SIMSCRIPT II and requires the use of a mainframe computer. The major difficulty with these models is that the execution time and memory requirements increase non-linearly as more trains and stations are added to the segment being considered. Our model attempts to introduce some mathematical structure in order to produce an algorithm which improves execution time even on a microcomputer such as the IBM PC. Initially, attention is restricted to rail segments which possess either single or double track subsegments. Further study is planned for the case of multiple track lines as well as the problem of memory space utilization.

THE MODEL

Single Track

A segment of railroad is viewed as being decomposed into subsegments and nodes. A <u>node</u> is any location where tracks diverge or meet and a <u>subsegment</u> is that part of a segment connecting two adjacent nodes. For the purpose of this discussion, an east-west line segment is considered with a total of n nodes, the easternmost node being the first with the others being numbered successively from east to west implying the westernmost node is the nth. Subsegments are denoted by ordered pairs e.g. (3,4) is the subsegment from node 3 to node 4 with westbound topographical characteristics while (4,3) is the same physical subsegment from node 4 to node 3 with eastbound topographical characteristics. A <u>train</u> is perceived as a row in each of three two dimensional arrays, A(i,j), D(i,j), and L(i,j) which represent the arrival time, delay time, and leaving time of the ith train at node j, $1 \le j \le n$, respectively. D(i,j) records the time the ith train is delayed at node j while either waiting for an opposing train or a preceding train to vacate the next subsegment. Although time is always presented to the user as 24 hour continental time, hh:mm, $0 \le hh < 24$, $0 \le mm < 60$, the model itself employs a counter which reckons time in minute increments or fractions thereof. (There are $60 \times 24 = 1{,}440$ minutes in one day.)

The effects of the segment's physical characteristics are stored in another two dimensional array, T(i,j), where each entry on the superdiagonal or subdiagonal, $|i-j|=1$, stores the time required to traverse a given node-to-node subsegment. Due to the possible presence of steep grades, it need not be the case that $T(i,j)=T(j,i)$. All other off diagonal entries are set equal to zero.

The diagonal entries of the traverse time array, $T(i,j)$, are used to delimit sections of double track and/or passing sidings as well as connections between each track of a double track section. If node j has double track emanating to the west then $T(j,j)=1$, to the east then $T(j,j)=-1$, in both directions then $T(j,j)=0$, otherwise, set $T(j,j)=9$. Note that this structure implicitly assumes that the traverse time between any two adjacent nodes i and j is independent of the particular track used i.e. either main track in double track territory or either the main track compared to a passing siding (which is not always the case) in single track territory. For this reason, it is assumed in the sequel that trains keep to the right.

If there are different classes of trains operating at different speeds on identical line segments, then more than one such two dimensional traverse time array must be supplied e.g. $P(i,j)$ for passenger trains, $F(i,j)$ for freight trains, $V(i,j)$ for trailvan or container trains, etc.

Multiple Track

To include the complication of multiple track territory, it would be necessary to make the time traverse array three dimensional, $T(r,i,j)$. Each entry $T(r,i,j)$ gives the time required to go from node i to node j on track r. The entries $T(r,i,i)$ are used to indicate the extent of track r; if track r emanates from node i to the west, then $T(r,i,i)=1$, to the east, $T(r,i,i)=-1$, in both directions, $T(r,i,i)=0$, otherwise, $T(r,i,i)=9$. Such a data structure would be able to account for a situation where the time it takes to go from node i to node j is dependent upon the particular track used (as in the case of a passing siding). The arrival and leaving time arrays would also become three dimensional, $A(r,i,j)$ and $L(r,i,j)$, with the new dimension recording the track upon which the ith train arrives and leaves node j respectively. (Note that a train needn't arrive and depart on the same track.)

THE SIMULATION LOGIC

Initialization of Parameters

The 0th entry in the arriving and leaving time arrays is used to designate each trains's direction; specifically,

$A(i,0)=-1$ for eastward trains,

$A(i,0)=1$ for westward trains.

The initial leaving time of the ith train is placed in an element of the leaving time array, $L(i,s)$, $1\leq s\leq n-1$, since a train may originate at any node on the line segment. The corresponding entry in the arrival array, $A(i,s)$, is set equal to the same value due to the technical requirements of the iterative step of the algorithm. (Ordinarily, one would expect it to be initialized to something else.)

The remaining entries in the arriving time Array, $A(i,j)$, $j\neq s$, are initialized so as to determine the extent of the journey of train i through the given segment subject to the condition that all trains originate and terminate at nodes. If $A(i,0)=-1$ (eastbound) and train i originates at node s and terminates at node t, $s>t$, then set

$A(i,j)=-1$, $s>j\geq t$; $A(i,j)=0$, $j>s$ and $j<t$;

$L(i,j)=-1$, $s>j>t$; $L(i,j)=0$, $j>s$ and $j\leq t$.

If $A(i,0)=1$ (westbound) and train i originates at node s and terminates at node t, $s<t$, then set

$A(i,j)=-1$, $s<j\leq t$; $A(i,j)=0$, $j<s$ and $j>t$;

$L(i,j)=-1$, $s<j<t$; $L(i,j)=0$, $j<s$ and $j\geq t$.

In other words, all entries in row i for train i are initialized to -1 if train i will eventually pass them while those entries which either remain behind the initial node or beyond the terminal node are initialized to 0. (The -1 entries in the arrival and leaving time arrays are referred to as sentinel entries for the remainder of the discussion.)

The delay time matrix $D(i,j)$ is initialized to the zero matrix.

The Iterative Step

1. Fix attention on those trains having at least one sentinel value (-1) remaining in the arrival time array, $A(i,j)$. (All other trains have already reached their final destinations.) If no -1 entries remain, then one should branch to the output module since every train has attained its terminal. Note that it is this part of the iterative step which necessitates the condition $A(i,s)=L(i,s)$.

2. Search for the train whose last non-sentinel entry (which depends upon the direction of travel i.e. $A(i,0)=1$ or $A(i,0)=-1$) is minimal i.e. earliest. It is the candidate for the next move. Suppose it is train a (so that it corresponds to the ath row of $A(i,j)$) and it is currently at node j. If $A(a,0)=-1$, then $A(a,j-1)=-1$ while if $A(a,0)=1$, then $A(a,j+1)=-1$.

3. Without loss of generality, assume $A(a,0)=1$ (westward movement); the logic is the same if $A(a,0)=-1$ except that one must examine the subsegment $(j,j-1)$ instead of the subsegment $(j,j+1)$. Four cases arise:

A. Entering double track: $T(j,j)=1$. It is only necessary to check whether any train is ahead of the given one on the subsegment $(j,j+1)$ moving in the same direction. To do this, consider all $L(x,j)$ for which $A(x,0)=1$ (same direction) and $L(x,j)<A(a,j)$ (those which have departed earlier). Calculate the maximum of these numbers,

$$M=\max_x\{L(x,j)\,|\,A(x,0)=1 \text{ and } L(x,j)<A(a,j)\}.$$

In addition, let

$$N=\max_{x}\{L(x,j+1)\,|\,A(x,0)=1 \text{ and } L(x,j)<A(a,j)\}.$$

Note that N represents the status of the preceding trains at the next node ahead of train a.

```
IF M+T(j,j+1)<A(a,j) THEN

     LET D(a,j)=MAX(N-A(a,j),0)

     LET L(a,j)=A(a,j)+D(a,j)

     LET A(j,j+1)=L(a,j)+T(j,j+1)

ELSE

     LET D(a,j)=MAX(N-A(a,j),
                    M+T(j,j+1))-A(a,j))

     LET L(a,j)=A(a,j)+D(a,j)

     LET A(j,j+1)=L(a,j)+T(j,j+1).
```

The preceding logic checks when the previous train moving in the same direction as train a departed node j (which is the calculation of M). It then determines whether than train has already reached the next node, j+1, with the statement, $M+T(j,j+1)<A(a,j)$? It also verifies the status of the preceding trains at the next node (which is the calculation of N) since the instant train cannot move until the next node, j+1, is clear (which accounts for the $N-A(a,j)$ term in $D(a,j)$). The logic also does not allow a train to leave node j until the immediate preceding train has arrived at node j+1 since the preceding train's status at node j+1 may affect the movement of the instant train; this occurs when $M+T(j,j+1)\geq A(a,j)$ and explains the additional term $M+T(j,j+1)-A(a,j)$ in the expression for $D(a,j)$ in the ELSE clause. Observe that the simplified model assumes right hand running. In order to incorporate the possibility of tracks being signaled in both directions, one need only add an additional dimension to the arrival and departure arrays respectively, $A(r,i,j)$ and $L(r,i,j)$ as previously mentioned, where the additional variable, r, records the respective tracks upon which train i arrives and leaves node j.

B. Continuing on double track: $T(j,j)=0$. The same logic as in A. above applies.

C. Continuing on single track: $T(j,j)=9$. The same logic as in A. above applies except that it might be desirable to include a check for opposing trains as in case Di. below and then produce an error message which accompanies a cessation of execution.

D. End of Double Track: $T(j,j)=-1$. It is necessary to investiage two situations:

i. Are there any opposing (in this case, eastward) movements currently between node j and the next subsegment (to the west) of double track? If so, train a must wait at node j for them to clear.

ii. Are there any trains ahead of the given one on the subsegment (j,j+1) moving in the same direction? If so, train a cannot proceed until the preceding train has cleared (left) node j+1.

To address item i., first locate the next subsegment of double track to the west. It commences at node

$$k=\min_{u}\{u\,|\,T(u,u)=1 \text{ and } u>j\}\ .$$

(If one were searching for the next subsegment of double track to the east as in the analogous situation for an eastbound train, $A(a,0)=-1$, then

$$k=\max_{u}\{u\,|\,T(u,u)=-1 \text{ and } u<j\}\ .)$$

Next, the latest prior departure time of an eastbound train from node k is obtained as

$$R=\max_{y}\{L(y,k)\,|\,A(y,0)=-1 \text{ and } L(y,k)<A(a,j)\}.$$

Suppose y^* is the value of y for which $R=L(y^*,k)$. The history of this opposing movement is then completed if it is not already complete i.e. cleared node j. The quantities

$$A(y^*,u),\ D(y^*,u),\ L(y^*,u),\ k>u\geq j$$

are calculated as in part A. (even though some of these times may be later (larger) than $A(a,j)$).

$$\text{IF } R+\sum_{u=j+1}^{k} T(u,u-1) + \sum_{u=j+2}^{k} D(y^*,u-1) < A(a,j)$$

THEN the latest prior opposing eastbound train has already arrived at node j and the program branches to case ii.;

ELSE

$$\text{LET } D(a,j)=R+\sum_{u=j+1}^{k} T(u,u-1) + \sum_{u=j+2}^{k} D(y^*,u-1) - A(a,j)$$

```
     LET L(a,j)=A(a,j)+D(a,j)

     LET A(a,j+1)=L(a,j)+T(j,j+1).

GOTO 4.
```

Note that since y^* is the latest opposing train in the subsegment (j,k), when it passes node j the track is clear from node j to node k.

If the program branches to part ii., there are no opposing (eastbound) movements ahead and the logic of A. applies.

4. GOTO 1.

When there are no more sentinel values (-1) in the arrival time array, $A(i,j)$, the simulation of the system for the given intial state is complete and the computer branches to an output module.

CONCLUSION

The simulation which has been described

possesses the capability to rapidly calculate the operating scenarios for given initial states involving a rail line segment having either single track or double track where right hand running prevails. The computer time required on the IBM PC remains manageable in the order of several minutes as long as there are no more than 15 trains and 25 nodes. Use of a compiler would offer a considerable improvement. These preliminary results give rise to the expectation that similar satisfactory execution times can be achieved for longer line segments and in the more difficult multi-track case where there is no specified current of traffic on any particular track. The authors' future plans include the development of the algorithm for the multi-track case along with the utilization of the algebraic properties of matrices to enhance the execution time. At the present level of development, the model exhibits the ability to be a useful planning tool for moderate density rail lines with the potential for application to high density corrodors.

The authors wish to acknowledge the assistance provided by the Southern Pacific Transportation Company.