ABSOLUTE CAPACITY DETERMINATION AND TIMETABLING IN RAILWAYS

Robert Burdett and Erhan Kozan

School of Mathematical Sciences

Queensland University of Technology,

PO Box 2434, Brisbane, Qld 4001, Australia

{r.burdett,e.kozan}@qut.edu.au

ABSTRACT

To determine with certainty the capacity of a railway line or network a train timetabling problem should be solved. However due to the size and complexity of this timetabling problem, simpler approximations and simulations are more often used in practice. Absolute capacity on the other hand may be more simply obtained by ignoring the possibility of collision conflicts, which are manifested as interference delays and occur as a result of insufficient passing facilities. This is achieved by allowing trains to pass through one another on non-critical sections in the timetable. Under this assumption, the full timetabling problem may be considerably reduced. In this paper details of a timetabling approach are presented and the calculation of important parameters required in the approach. Mathematical models for solving both the full and reduced timetabling problem are then developed. Alternative heuristics however are also proposed and extended because the reduced problem is still of considerable size and complexity.

Keywords: Scheduling, Capacity Analysis, Railways

1. INTRODUCTION

This paper is concerned with the exact measurement of capacity in railway lines and networks. The accurate quantification of railway capacity is of great importance especially in Australia in recent times as additional demands are placed upon railways to provide third party usage of infrastructure. There is also a trend towards the complete separation of railway infrastructure ownership and operation. Consequently to ensure fair and impartial access to infrastructure, railway operators must be able to clearly differentiate between free and used capacity. For more information on these issues Ferreira (1997) and Gibson (2003) for example may be consulted. The accurate quantification of capacity also continues to prove difficult as more advanced elements that add realism are included in the analysis.

Railway capacity may however be defined in a variety of different ways, for example see Kozan and Burdett (2004) and Kort et al (2003). In this paper capacity is defined as *the maximum number of trains that either traverse a critical bottleneck section or can traverse the network (in entirety or proportionally) in a given duration of time.*

Capacity however is not a unique value. It exists for every possible mix of trains. Hence capacity assessment is performed to decide whether the infrastructure can handle the intended traffic load. Railway capacity may be further specified as actual or absolute, and these terms remain independent of the particular definition of railway capacity. Absolute capacity is a theoretical value that is realised when only a critical (bottleneck) section(s) are saturated. It is an ideal case that is not necessarily possible in reality. Actual (sustainable) capacity is the real amount that occurs when interference delays are incorporated. Interference delays are the enforced idle time of the system and occur as a result of insufficient passing (overtaking) facilities. Both actual and absolute measures of capacity are required in practice for a variety of reasons. Fore mostly is that they allow traffic congestion to be quantified. Traffic congestion in any transportation system is an important issue, particularly its minimisation. This is because congestion can cause long delays and customer dissatisfaction, which can culminate in financial losses. Calculation of congestion also provides a very good reason for determining absolute capacity. Absolute capacity would otherwise appear to be a meaningless value because the state associated with this value is unattainable, i.e. the railway can never fully manage this number of trains in the given duration of time. Actual capacity however has other more numerous usages. For example actual capacity is used to determine train path charging (see Gibson et al (2002)). Hence more emphasis is usually placed on its calculation.

In this paper a timetabling approach is employed for determining absolute (and actual) capacity. Many researchers have avoided this avenue in the past because of its complexity however the benefits of such an approach are significant. For example the accuracy of previous capacity approximations may be verified by a timetabling approach. The inclusion of pre-planned dwell (or dwelling) times at passing facilities significantly complicates absolute capacity determination, particularly approximations. Consequently, an exact timetabling approach is again warranted. No other "full proof" avenue exists to the author's knowledge. Capacity approximations do not indicate how to optimally utilise the railway in order to achieve capacity. Timetabling approaches however achieve both.

Timetabling however does not define capacity directly. That is, the usual question, "What is the capacity of the railway line or network," cannot be directly answered. Timetabling however can be used to verify whether it is possible for a given number of trains (of a particular mix) to traverse the critical section or the entire network. If possible in exactly the given duration of time, then it can be concluded that firstly capacity exists, and secondly that the capacity of the railway is in fact the given number of trains.

The absolute capacity of a railway line or network in particular may be verified by solving a reduced timetabling problem that has an "unresolved" timetable containing train conflicts as output. The full timetabling problem does not necessarily need to be solved when determining absolute capacity due to the fact that sequences on critical sections of each distinct corridor must only be included. However to obtain achievable (or real) capacity the full timetabling model must be solved. The output is a fully "resolved" timetable devoid of any train conflicts. The optimal unresolved timetable may be used as a starting point when solving the full timetabling problem. The optimal unresolved schedule however does not guarantee an optimal resolved schedule. At least there is no strict mathematical or empirical proof that suggests this to the author's knowledge.

In the next section a review of related research is presented. Mathematical notation that is used in the remainder of the paper is then displayed in section 3. Important parameters and their calculation are addressed in section 4 before mathematical models are developed in section 5. In section 6 heuristics and other strategies for solving larger problems are developed. Lastly concluding remarks are given in section 8.

2. LITERATURE REVIEW

There has been much research in the area of railway operations in recent years. A complete survey of optimisation models for train routing and scheduling can be found in Cordeau et al (1998). In summary the re-scheduling problem has received the most attention, for example by Carey M. (1994(a)(b)) and Higgins et al (1996) for exact approaches and Cai and Goh (1994), Higgins and Kozan (1997), Cai et al (1998), Chiang et al (1998), Sahin I. (1999), and Adenso-Diaz et al (1999), Dorfman and Medanic (2004) for heuristic approaches. This is not surprising, since this task must be routinely performed in practice on an everyday basis. Improvements in performing this task can give significant improvements in infrastructure utilisation, thus leading to a reduction in various operating and investment costs.

Train platforming and pathing (routing) is another aspect that has received above average attention recently by Carey and Lockwood (1992), Zwaneveld et al (1996), Kroon et al (1997), Zwaneveld et al (2001), Billionnet (2003), Carey and Carville (2003).

The timetabling problem on the other hand has received far less reference in the literature. Szpigel proposed a generic job shop model in 1972. Quite a gap follows before Odijk (1996), Brannlund et al (1998), Nachtigall (1999), Lindner (2000), Kroon and Peeters (2003). Timetabling does not necessarily need to be performed on a daily basis, and the objectives associated with this task are also more diverse, and questionable from an industry perspective. Timetabling appears to include a number of unquantifiable and conflicting elements that only a human expert can take into account (Chiang et al 1998).

The scheduling, timetabling and routing problems can be viewed and classified as various types of job shop. The job shop is the backbone of many Operations Research problems and has attracted significant attention and this will most likely continue. A list of some recent work that is related to this train timetabling problem is as follows: Khosla (1995), Franca et al (1996), Hall and Sriskandarajah (1996), Daniels et al (1999), Nowicki (1999), Steinhofel et al (1999), Allahverdi et al (1999), Mascis and Pacciarelli (2002), Kim et al (2003), Corry and Kozan (2004).

3. NOMENCLATURE

i=1,..,N:	Train index and number of trains.
j=1,..,M:	Section index and number of sections.
$m_{i,k}$:	The kth section traversed by train i. Note: k=1,.., K_i .
L_i :	The length of train i.
Z:	A large value
$E_{i,k}, D_{i,k}, SRT_{i,k}$:	The planned entry, exit, and sectional running time of train i on the kth section respectively that it traverses.
$DWELL_{i,k}$:	The dwelling time between the k-1st and kth section traversed.
$X_{i,i',j}$:	Binary decision variable signifying whether train i precede's train i' on section j.
$H^{F-S}_{i,i',j}, H^{S-S}_{i,i',j}$:	The finish-start and start-start headway time between train i and i' on section j if $i \prec i'$.
$\delta_{i,i',j}$:	A binary parameter signifying whether trains i and i' traverse section j in the same direction.
N_j :	The set of trains that traverse section j. The size of the set is $\left\|N_j\right\|$ and a train i is a member, i.e. $\left(i \in N_j\right)$ iff $\exists k \mid m_{i,k} = j$.
O:	Set of omitted sections
P_c, Q_c :	The set of sections and locations respectively in corridor c.
$F_{c,q}, F'_{c,q}$:	The id of the qth location respectively in indistinct and distinct corridor c.
$C, \bar{C}$:	The set of indistinct and distinct corridors.
Φ, Φ' :	The set of IO and pseudo IO points.

REF, *XL*: The set of section pairs $\left(\text{i.e.}\left\{\left(j, j'\right) \mid j \neq j', j, j' \leq M\right\}\right)$ that indicate respectively whether a reference/signal location or a crossing loop occurs between these sections.

4. PARAMETERS

This paper addresses generic networks and not just single lines. Because of this factor, the calculation of certain parameters becomes more difficult and involved, and requires further elaboration. Railway timetabling requires significant quantities of data, and any automatic generation of these values is hugely beneficial to researchers and practitioners in academia and industry.

4.1. Headway Calculations

Determining the correct separation (or headway time) between trains is vitally important for safety. There are two headway types. The first occurs between the rear of a train and the front of the following train (i.e. finish-start (F-S)), while the second occurs between the front of a train and the front of the following train (i.e. start-start (S-S)). Both headway types are necessary and may be calculated by equations (1)-(8). Consequently the burden of providing these values as input is not required. It should be noted that the index's k and k' are chosen such that $m_{i,k} = m_{i',k'} = j$. Headways also do not need to be calculated between two trains on any section that both trains do not traverse.

For trains in the same direction (i.e. $\delta_{i,i',j} = 1$)

$$H_{i,i',j}^{S-S} = SRT_{i,k} \quad \forall j \in SOC, \forall i, i' \in N_j \tag{1}$$

$$H_{i,i',j}^{F-S} = 0 \quad \forall j \in SOC, \forall i, i' \in N_j \tag{2}$$

$$H_{i,i',j}^{S-S} = SRT_{i,k} - SRT_{i',k'} \quad \forall j \notin SOC, \forall i, i' \in N_j \mid SRT_{i,k} > SRT_{i',k'} + h \tag{3}$$

$$H_{i,i',j}^{F-S} = -SRT_{i',k'} \quad \forall j \notin SOC, \forall i, i' \in N_j \mid SRT_{i,k} > SRT_{i',k'} + h \tag{4}$$

$$H_{i,i',j}^{S-S} = \min\left(h, SRT_{i,k}\right) \quad \forall j \notin SOC, \forall i, i' \in N_j \mid SRT_{i,k} \leq SRT_{i',k'} + h \tag{5}$$

$$H_{i,i',j}^{F-S} = -SRT_{i',k'} + h \quad \forall j \notin SOC, \forall i, i' \in N_j \mid SRT_{i,k} \leq SRT_{i',k'} + h \tag{6}$$

For trains in opposite directions (i.e. $\delta_{i,i',j} = 0$)

$$H_{i,i',j}^{S-S} = SRT_{i,k} + h \quad \forall j, \forall i, i' \in N_j \tag{7}$$

$$H_{i,i',j}^{F-S} = h \quad \forall j, \forall i, i' \in N_j \tag{8}$$

4.2. Direction of Travel Calculation

In train timetabling, direction of travel (DOT) is important particularly when determining minimum separation of trains. In railway lines there are only two directions, and these are usually defined by one of the following alternatives: (up/dn), (inbound/outbound), (+/-), (0/1). The direction of travel for each train based upon one of the above is then given as an input.

In networks however defining direction of travel in this way doesn't really make sense particularly if sections are identified by a single index. For this reason an alternative approach is proposed. The approach is based upon the logic that determining whether two trains travel in the same direction does not require that a direction be explicitly defined for each section. Consequently an explicit direction is not required for each train and section that it traverses.

Direction is a relative measure and requires two points to determine. A train's path is therefore sufficient to determine its direction of travel as long as more than one section is traversed. The binary parameter $\delta_{i,i',j}$ is defined and signifies that train i and i' traverse section j in the same direction when $\delta_{i,i',j}$=1. The variable is valid $\forall j, \forall i, i' \in N_j \mid i \neq i'$. When trains may only traverse one section (as occurs between two adjacent IO points) the approach must be modified by introducing $m'_{i,k}$ as the kth location to be passed by train i. The minimum number of locations that can be passed is therefore two, which is sufficient in working out direction of travel. Linking direction of travel between two locations to a particular section requires variable $\lambda_{l,l',j}$ which signifies whether section j occurs between location l and l'. The following two equations may then be used determine direction of travel.

$$\delta_{i,i',j} = 1 \ \forall i, i', j \mid i \neq i', \exists(l, l', k, k') \mid l \neq l', \lambda_{l,l',j} = 1, m'_{i,k} = m'_{i',k'} = l, m'_{i,k+1} = m'_{i',k'+1} = l'$$
$$\delta_{i,i',j} = 1 \ \forall i, i', j \mid i \neq i', \exists(l, l', k, k') \mid l \neq l', \lambda_{l,l',j} = 1, m'_{i,k} = m'_{i',k'+1} = l, m'_{i,k+1} = m'_{i',k'} = l'$$

4.3. Distinct Corridor Calculation

An IO point is any point on the network where trains may enter and leave the system. A corridor is therefore the set of sections that are sequentially traversed (i.e. the path) from one IO point to another. Any corridor that does not overlap any other is defined as distinct. More specifically distinct corridors occur between IO and pseudo IO points. **A** pseudo IO point occurs at the point where two corridors overlap. It is referred to as a type of IO point because trains enter and leave a distinct corridor at this point. Therefore when distinct corridors are considered independently, the boundaries may be viewed as pure IO points. A pseudo IO point is characterised by a node with degree of two or greater in an undirected graph of the layout (where nodes are locations and arcs are sections).

When determining absolute capacity one sequence is required for every distinct corridor. For standard railway lines where there is a single distinct corridor this can be accomplished by finding the sequence of trains on the critical section. In networks however each train cannot be assumed to traverse every section and therefore a single critical section does not exist. A corridor may also overlap other corridors and hence the mix of trains on all sections of the corridor will not be the same. All non-critical sections can be omitted when determining absolute capacity.

Determining distinct corridors is equivalent to determining all serial components that begin and end between an IO or pseudo IO point. This topic is addressed because railway network information is stored in terms of general (indistinct) corridors. The following algorithm may be used for the determination of distinct corridors in a railway network.

Algorithm: Distinct corridor calculation
Begin
  for (c=0,..,$|C|$) // For each standard (indistinct) corridor
  begin
    q=0; s=0; $c' = 0$
    while ($q < |Q_c|$) // For each location in the corridor
    {
      q=q+1; // Increment the position
      $id = F_{c,q}$; // Current location id
      if ($id \in \Phi$ or $id \in \Phi'$) // Location is an io or pseudo io location
      begin
        $c' = c' + 1$; // Increment number of distinct corridors.
        if a distinct corridor with boundaries $F_{c,s}$ and $F_{c,q}$ has not been defined **then**
        begin
          Create a new distinct corridor c'
          Set boundaries as $F_{c,s}$ and $F_{c,q}$.
          Create new path for this distinct corridor, i.e.
          for($q' = s,..,q$) $F'_{c',q'-s+1} = F_{c,q'}$;
          Compute the associated path of sections;
          Calculate the length of this new corridor.
          s=q; // Set starting location of next distinct corridor (if one exists).
        end
      end
    end
  end
End

4.4. Railway Network Reduction

When determining absolute capacity, the majority of the sections in the railway become redundant in the analysis. This is particularly true when distinct corridors have many sections. Therefore it can be beneficial if the input information is reduced. This is however quite complex and involved from a programming perspective. The following algorithm provides some information on how this is accomplished.

Algorithm: Railway Network Reduction
Begin

Step 1: Copy IO and pseudo IO locations to the reduced railway as they remain the same.
Step 2: Redefine section and distinct corridor information using the following:
for ($c \in \bar{C}$) // For each distinct corridor *c*
begin
Create a new reduced corridor;
Set the boundaries of this corridor as the boundaries of the distinct corridor;
Set the length as the length of the distinct corridor;
Set *s* as the starting location of the distinct corridor;
length=0; // Set new section length as zero
For each location *q* in the path of distinct corridor *c*
begin
Increment the accumulated length (i.e. *length*) by adding the length of the q-1st section of distinct corridor *c*;
Set the *e* as the location id of the *q*th location;
Set *b*1 and *b*2 as the boundaries of the critical section of distinct corridor *c*;
if (*b*1=*e* or *b*2=*e* or *e* is the final location)
begin
Create a new section with boundaries (*s*,*e*) and length *length*.
Set number of tracks as one;
s=*e*; // Set new section starting location as the end of the previous new section
length=0; // Reset next section length as zero
Add location to the path of the new corridor;
end
end
Add corridor to reduced distinct corridor list;
end
Step 3: Redefine corridor information by reapplying step 2 with respect to indistinct corridor *c* instead. Add corridor to reduced corridor list.
Note: Do not redefine new sections in step 3 as this was accomplished in step 2.
Step 4: Convert critical section information for both the reduced distinct and indistinct corridors by looking up the new label of the previous critical sections.
Step 5: Redefine corridor path information in terms of the new section information;
End

The train path information must also be redefined in terms of the new railway network information. The following algorithm provides some information on how this is accomplished and the procedure is called for each train that is to be scheduled.

Algorithm: Train path data redefinition
Begin
Clear *SRT* values for recomputation;
transit=0; // Initialise transit time for re-computation
Initial dwell value does not change, i.e. keep $DWELL_0$;
temp=0; *length*=0; // Set counters
Set *s* as the id of the starting location in the train's original path;
For each location *q* in the trains original path starting at the second location
begin
Set *e* as the id of the current location, i.e. the *q*th;
Increment the accumulated length (i.e. *length*) by adding the length of the q-1st section.
if ((*s*,*e*) is a legitimate section or *q* is the last location)
begin
s=*e*; // Set new section starting location
Add a new dwell point with value $DWELL_q$;

During this procedure dwell times are added as additional sectional running time on merged sections.

5. MATHEMATICAL MODELS

The following model timetables specific trains according to the makespan minimisation objective.

Makespan Calculation:

$$Cmax \geq D_{i,K_i} \quad \forall i \tag{9}$$

Scheduling constraints:

$$E_{i,k} = D_{i,k-1} \quad \forall i,k \mid 1 < k \leq K_i, \left(m_{i,k-1}, m_{i,k}\right) \notin XL \tag{10}$$

$$E_{i,k} \geq D_{i,k-1} + DWELL_{i,k} \quad \forall i,k \mid 1 < k \leq K_i, \left(m_{i,k-1}, m_{i,k}\right) \in XL \tag{11}$$

$$D_{i,k} \geq E_{i,k} + SRT_{i,k} + DWELL_{i,k} \quad \forall i,k \mid 1 \leq k < K_i, \left(m_{i,k}, m_{i,k+1}\right) \notin XL \tag{12}$$

$$D_{i,k} = E_{i,k} + SRT_{i,k} \quad \forall i,k \mid 1 \leq k < K_i, \left(m_{i,k}, m_{i,k+1}\right) \in XL \tag{13}$$

$$D_{i,k} = E_{i,k} + SRT_{i,k} \quad \forall i,k \mid k = K_i \tag{14}$$

Separation constraints:

$$\left(1 - X_{i,i',j}\right)Z + E_{i',k'} \geq D_{i,k} + H^{F-S}_{i,i',j} \quad \forall j \notin O, \forall i,i' \in N_j \mid i < i', \delta_{i,i',j} = 0 \tag{15}$$

$$\left(X_{i,i',j}\right)Z + E_{i,k} \geq D_{i',k'} + H^{F-S}_{i',i,j} \quad \forall j \notin O, \forall i,i' \in N_j \mid i < i', \delta_{i,i',j} = 0 \tag{16}$$

$$\left(1 - X_{i,i',j}\right)Z + E_{i',k'} \geq E_{i,k} + H^{S-S}_{i,i',j} \quad \forall j \notin O, \forall i,i' \in N_j \mid i < i', \delta_{i,i',j} = 1 \tag{17}$$

$$\left(X_{i,i',j}\right)Z + E_{i,k} \geq E_{i',k'} + H^{S-S}_{i',i,j} \quad \forall j \notin O, \forall i,i' \in N_j \mid i < i', \delta_{i,i',j} = 1 \tag{18}$$

$$\begin{gathered} X_{i,i',j'} - X_{i,i',j} \leq 0 \quad \forall j, j', \forall i,i' \in N_j \cap N_{j'} \mid i < i', j \neq j', \\ \exists\left(k,k'\right): k < K_i, k' < K_{i'}, m_{i,k} = j, m_{i,k+1} = j', m_{i',k'} = j', m_{i',k'+1} = j \end{gathered} \tag{19}$$

Integrality Conditions:

$$X_{i,i',j} = 0 \text{ or } 1 \quad \forall i,i', j \mid i < i' \tag{20}$$

The number of binary variables is reduced by half in this model because, $X_{i,i',j} + X_{i',i,j} = 1 \quad \forall j,i,i' \in N_j \mid i < i'$. Note however that this is not true on any section which is traversed by one or no trains in the pair. That is, $X_{i,i',j} + X_{i',i,j} = 0 \quad \forall j,i,i' \mid i \notin N_j$ or $i' \notin N_j$.

This model however, is insufficient as it is for determining network capacity. To calculate network capacity by mathematical programming one of three approaches may be taken which require a number of modifications and additions. The base-scheduling model with these additions and modifications is then suitable. For determining actual capacity the set of omitted sections is empty, otherwise it contains sections that are not critical on a distinct corridor. Crossing loop capacity conditions are also unnecessary when determining absolute capacity because trains are not delayed at crossing loops to avoid collision conflicts, i.e. there are no interference delays. Otherwise additional constraints are required. This aspect however is outside the scope of this paper.

Approach 1:

In this approach the objective is to maximise throughput. The following additional constraints are needed.

$$Throughput = \sum_i \xi_i \tag{21}$$

$$\sum_i \xi_i \le C^{\text{abs}} \tag{22}$$

$$\sum_i \left(\tau_{i,y}\right) \ge \left\lfloor n_y \sum_i \xi_i \right\rfloor \quad \forall y \tag{23}$$

$$D_{i,K_i} \le T\left(\xi_i\right) \quad i = 1,..,C^{\text{abs}} \tag{24}$$

$$\sum_y \tau_{i,y} = \xi_i \quad i = 1,..,C^{\text{abs}} \tag{25}$$

$$\tau_{i,y} = 0,1\,\forall i,y, \quad \xi_i = 0,1\,\forall i \tag{26}$$

$$E_{i,k} \le \left(\xi_i\right) Z \text{ and } D_{i,k} \le \left(\xi_i\right) Z \quad \forall i,k\,. \tag{27}$$

C^{abs} is an approximation of capacity and is used as an upper bound on the number of trains that must be timetabled. To calculate C^{abs} we refer the reader to Burdett and Kozan (2004). ξ_i is a binary variable that signifies whether the *i*th train can be timetabled and $\tau_{i,y}$ is a binary variable that signifies whether the *i*th train is of type *y*. The makespan is defined as time *T* and the mix of trains is defined by $\underset{\sim}{\eta}$. In particular η_y is the proportion of trains that must be of type *y*.

Constraint (22) enforces that the value of capacity is less than the value obtained through the bottleneck analysis. Constraint (23) ensures that the number of trains of each type roughly satisfy the given proportional requirements. This constraint however is non-linear because of the floor function. An alternative is to split this constraint into two parts as follows:

$$\sum_i \left(\tau_{i,y}\right) \ge n_y \sum_i \xi_i - 1 \quad \forall y \text{ and } \sum_i \left(\tau_{i,y}\right) \le n_y \sum_i \xi_i \quad \forall y \tag{28}$$

Constraint (24) ensures that all timetabled trains finish before time *T*. Constraint (25) ensures that a timetabled train is given a type while un-timetabled trains are not. In the original timetabling constraints a number of additional modifications are required for this capacity determination model. Firstly the following constraint is added.

$$m_{i,k} = \xi_i \sum_y \left(\hat{m}_{y,k} \tau_{i,y}\right) \forall i,k \left(\text{or } m_{i,k} = \sum_y \left(\hat{m}_{y,k} \tau_{i,y}\right) \forall i,k : \xi_i = 1 \right) \tag{29}$$

$\hat{m}_{y,k}$ is a parameter that signifies the *k*th section traversed by a train of type *y*. Constraint (29) provides a link between the path of a particular train type and a particular train (that exists). That is, a correct path is given to each train that is to be timetabled. Unfortunately this constraint is non-linear or endogenous. The other standard timetabling constraints also become endogenous because they can only be formulated where trains satisfy the existence condition, i.e. where $\xi_i = 1$. Consequently the model is dynamic.

It should also be noted that $X_{i,i',j} = 0 \quad \forall j,i,i' \mid \xi_i = 0 \text{ or } \xi_{i'} = 0$. That is, there is no precedence relationship between non-existent trains.

Approach 2:

In this model, X_y is a decision variable for the number of each type of train being timetabled.

$$Throughput = \sum_y \left(X_y \right) \tag{30}$$

$$X_y \geq \left\lfloor \eta_y \sum_{y'} \left(X_{y'} \right) \right\rfloor \quad \forall y \left(i.e.\ X_y \geq \eta_y \sum_{y'} \left(X_{y'} \right) - 1 \text{ and } X_y \leq \eta_y \sum_{y'} \left(X_{y'} \right) \right) \tag{31}$$

$$\sum_y X_y \leq C^{\text{abs}} \tag{32}$$

$$m_{i,k} = \sum_y \left(\hat{m}_{y,k} \tau_{i,y} \right) \forall k, i = 1,.., \sum_{y'} X_{y'} \tag{33}$$

$$\sum_i \tau_{i,y} = X_y \quad \forall y \tag{34}$$

$$\tau_{i,y} = 0,1 \; \forall i, y \tag{35}$$

$$X_y \text{ integer} \quad \forall y$$

Constraint (31) ensures that the number of trains of each type satisfy the given proportional requirements. Constraint (32) has the same effect as constraint (22). The model is again endogenous.

Approach 3:

All C^{abs} trains are timetabled in this model, however the timetabling process is performed in such a way that the largest number of these trains finishes before time *T*. Therefore, the train mix and the train type constraints and binary variables are not necessarily required unless strict proportional requirements are needed. The model also stays linear, and is not endogenous as previous models were. This is because the number of trains to be timetabled is fixed.

$$Throughput = \sum_i \varsigma_i \tag{36}$$

$$D_{i,K_i} \leq T + \left(1 - \varsigma_i \right) Z \quad \forall i \tag{37}$$

ς_i is a binary variable which signifies if train *i* finishes before time *T*. Constraint (37) ensures that $\varsigma_i = 1$ only if train *i* finishes before time *T*.

5.1. Upper and Lower Bound Calculations

The following upper and lower bounds on the timetable makespan are proposed for railway lines and networks.

$$UB = \max_c \left(T_c + \max_{\forall i,i' | i \neq i', \Gamma_{i,c} = \Gamma_{i',c} = 1} \left(\alpha_{i,c} + \beta_{i',c} \right) \right) \tag{38}$$

$$LB = \max_c \left(T_c + \min_{\forall i,i' | i \neq i', \Gamma_{i,c} = \Gamma_{i',c} = 1} \left(\alpha_{i,c} + \beta_{i',c} \right) \right) \tag{39}$$

In these equations: $\Gamma_{i,c}$ signifies whether train *i* traverse's all sections in distinct corridor *c*. More specifically, $\Gamma_{i,c} = 1$ if there exists for each $j \in P_c$ a $k \in \left[1, K_i \right]$ such that $m_{i,k} = j$.

$T_c = \sum_{\forall i:\Gamma_{i,c}=1} \left(SRT_{i,CS_c} \right)$ and is the total time required by trains to traverse the critical section (i.e. CS_s) of distinct corridor *c*. The time to reach the critical section on the *c*th distinct corridor from a train's origin and the time to reach a train's destination from the critical section on the *c*th distinct corridor respectively is $\alpha_{i,c}, \beta_{i,c}$. These values are calculated in the following way and only on distinct corridors that a train traverses.

$$\alpha_{i,c} = \sum_{1 \le k < \gamma_{i,c}} \left(SRT_{i,k} \right) \text{and } \beta_{i,c} = \sum_{\gamma_{i,c} < k \le K_i} \left(SRT_{i,k} \right) \tag{40}$$

The number of sections that train *i* must be traverse before the critical section of corridor *c* is reached is defined as $\gamma_{i,c} - 1$ where $\gamma_{i,c} \in [1, K_i]$ and $\gamma_{i,c} = k : m_{i,k} = CS_c$.

These bounds may be generalised for the standard (or full) timetabling problem. The only modification required is for the distinct corridor index *c* to be replaced with section index *j*. Also, CS_c is replaced with *j*, and the condition $\Gamma_{i,c} = 1$ in (38) and (39) is replaced with $\exists k : m_{i,k} = j$.

Headways and pre-planned dwell times have not been included in the upper and lower bound calculations currently and could be at some stage. Neither has the removal of SOC's been included, i.e. bounds assume there is an SOC of one train.

6. SOLUTION TECHNIQUES

Scheduling problems are NP–hard and hence mathematical programming approaches will be insufficient, especially when the number of trains is large as occurs in an analysis of absolute capacity.

The capacity determination problem is equivalent to a job shop scheduling problem which allows that theory to be used as a basis to solve this problem. Several modifications and extensions however are required which are now discussed after a quick review. We assume however that readers have a basic knowledge of job shop theory. For a more comprehensive review the reader may refer to Pinedo (1995) or Blazewicz et al (1996) for example.

A section is analogous to a machine and a train is analogous to a job. An activity on node (AON) directed graph is constructed. The application of the longest path algorithm schedules trains. The longest path values on certain nodes are associated with the entry and exit time of trains from sections. A solution is a selection of disjunctive arcs. Disjunctive arcs are added to a directed graph containing conjunctive arcs. Conjunctive arcs link successive operations of a job and are fixed while disjunctive arcs link operations of different jobs on a common machine resource. There is a chain of nodes and conjunctive arcs for each train, which starts at the source node and ends at the sink node. The source and sink nodes must be added to ensure correct scheduling.

Disjunctive arcs define the sequence of jobs on each machine, i.e. the sequence of train movements! A disjunctive arc record (i.e. a disjunctive arc) consists of two different directed arcs and is associated with two particular trains on a particular section. Usually the arcs are the opposite of each other however this is not always true. Only one arc may be in the solution. A binary parameter may be used to signify which arc is inserted is the actual solution.

6.1. Modifications

Disjunctive arc weightings are the headways between trains. Usually they are zero in the usual AON digraph**.**

Trains have length and this affects capacity, particularly on sections that are smaller than the length of the train. The graph structure must be modified to cope with this addition. Train length causes time lags, and in conjunction with dwell times can significantly affect the schedule.

Dwell time occurs on a section at its far boundary in the direction of travel. This is because a train has length. It is stopped over a particular distance, not a single point. The section before a crossing loop normally has no dwelling time because it occurs on the crossing loop instead. Dwell time at an origin location (that is an IO point) does not count as dwell time on the train's origin section. To be included a dummy section with no length but sectional running time equal to the dwell time must be added. This section will also have a finite or infinite capacity to store trains.

The modified graph structure will now be explained with respect to an example. Consider a train (shaded) in the following time-distance diagram for a small serial railway line.

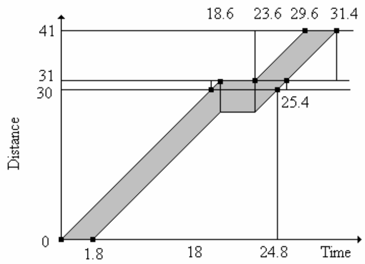
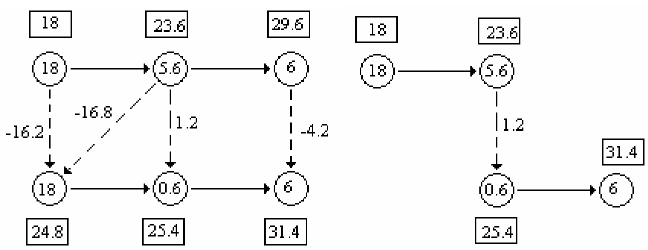


Figure 1. A train trajectory including train length

Dwell time at the third location (i.e. position 31km) for example increases the time spent on the second section and also the first. The time spent on the first section is greater than the SRT and time lag. To obtain the correct entry and exit times shown in the previous figure, the following directed graphs may be used. The node values are the train's section occupation time (SOT), and the square boxes are the section exit times determined after the longest path method has been applied. The dotted arcs are pretend disjunctive arcs, which represent the precedence between two independent jobs (i.e. the front and rear of a train in this case). They are pretend disjunctive arcs because they are fixed unlike the standard situation. For these pretend disjunctive arcs an arc weight less than zero indicates an overlap while a value greater than zero indicates a time lag.



(a) (b)

Figure 2. AON digraphs for scheduling a train with length

Not all sections are smaller than a train. Therefore it is not necessary to split a train into two parts as shown in Fig 2(a). The front and rear of a train are split only for particular sections that are shorter than the train as shown in (b). A consequence of this type of extension is that the disjunctive arcs between different trains will no longer be the reverse of each other. This complicates implementation considerably.

To get around this a combined front-rear (i.e. no split) approach may be considered. The following graph gives the correct solution for this example. The data was pulled off the previous digraphs.

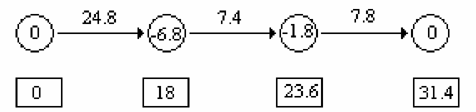


Figure. AON digraph for the example

6.2. Graph Reduction

For simplicity the directed graph is built for all sections and disjunctive arcs are only added for those sections that are to be sequenced (i.e. which are not omitted). The directed graph however may be simplified by reducing the original network as described in section 4 so that redundant sections are not required. This means that the directed graph will be much smaller, i.e. will contain far fewer nodes and conjunctive arcs. Consequently graph generation and the longest path algorithm will take considerably less computational time. However network simplification algorithms are very complex. Solutions of the simplified problem may then also need to be converted back to the original full sized problem and this is another complex task.

6.3. Initial Solution Construction

To construct an initial timetable a number of different approaches were investigated. A description of these however is outside the scope of this paper. We therefore provide details of the best approach, which is based upon the NEH insertion algorithm by Nawaz, Enscore and Ham (1983).

Details: The NEH insertion algorithm is applied to each machine in a sequential order. The ordering of the machines is determined according to the amount of processing required on a machine. In our case this is the planned (minimal) section occupation time, i.e. the time the section will be occupied with trains. The ordering is from largest to smallest. The logic is that jobs with larger processing requirements have a greater impact on the solution and have a greater priority during insertion. The NEH insertion algorithm is a greedy approach that builds a sequence by inserting jobs (one at time) into the partial (incomplete) sequence. Each possible insertion point (in the partial sequence) is inspected and the best is chosen for the current jobs insertion. There will be $\sum_j \left(\frac{1}{2} N_j \left(N_j + 1 \right) - 1 \right)$ number of insertions if N_j is the number of jobs that require machine *j*. The insertion of a specific train in a particular position of the partial sequence involves two direct precedence relationships. That is, it involves the insertion and subsequent removal of two disjunctive arcs. Other disjunctive arcs associated with general precedence relationships are redundant and do not need to be inserted. Linking the insertion of a job with disjunctive arcs makes the implementation of this approach more complex than in other scheduling applications. An insertion may also cause a cycle in the directed graph. During evaluation by the longest path algorithm, a topological traversal is performed. If this traversal does not include each node then a cycle exists, and the move is not allowed.

6.3. Heuristics

Simulated Annealing (SA) and Tabu Search (TS) are ideally suited to this problem. Evolutionary strategies among others may also be applied with more effort.

For an SA approach, disjunctive arcs are viewed as either critical or non-critical. A new solution is obtained by an operator that changes critical disjunctive arcs. During the longest path method disjunctive arcs on one critical path from source to sink are identified. An alternative is to identify all longest paths and hence all critical disjunctive arcs may be obtained.

7. CONCLUSIONS

In this paper train timetabling approaches for determining primarily absolute capacity were proposed. These timetabling approaches however may be applied (after slight modification) to generic train timetabling problems that do not necessarily have the same objective function and do not necessarily address capacity determination.

Preliminary results from the application of these approaches are promising but a complete numerical investigation is the source of another paper.

REFERENCES

Adenso-Diaz B., Gonzalez M. O., Gonzalez-Torre P. (1999), On line timetable re-scheduling in train services. *Transportation Research Part B*, **33**, 387-398.

Allahverdi A., Gupta J. N. D. and Aldowaisan T. (1999), A review of scheduling research involving setup considerations. *Omega, International Journal of Management Science*, **27**, 219-239.

Billionnet A. (2003), Using integer programming to solve the train platforming problem. *Transportation Science*, **37.2,** 213-222.

Blazewicz J., Ecker K., Pesch E., Schmidt G., Weglarz J. Scheduling Computer and Manufacturing Processes, Springer-Verlag, Berlin , 1996.

Brannlund U., Lindberg P. O. Nou A. Nillson J.-E. (1998), Railway timetabling using lagrangian relaxation. *Transportation Science*, **32**, 358-369

Kozan E. and Burdett R. L. (2004), Capacity Determination Issues in Railway Lines. CORE 2004, New Horizons for Rail Conference Proceedings, Darwin, Australia, 31.1-31.6

Cai X., and Goh C. J. (1994). A fast heuristic for the train scheduling problem. *Computers and Operations Research*, **21.5,** 499-510.

Cai X., Goh C. J., and Mees A. I. (1998). Greedy heuristics for rapid scheduling of trains on a single track. *IIE Transactions*, **30**, 481-493.

Carey M. and Lockwood D. (1992), A model, algorithms and strategy for train pathing. Research Report, Faculty of Business and Management, University of Ulster,North Ireland. *J Operational Res. Soc.*

Carey M. (1994), A model and strategy for train pathing with choice of lines, platforms, and routes. *Transportation Research B*, **28.5**, 333-353.

Carey M. (1994), Extending a train pathing model from one-way to two-way track. *Transportation Research B*, **28.5,** 395-400

Carey M and Carville S. (2003), Scheduling and platforming trains at busy complex stations. *Transportation Research A*, **37**, 195-224.

Chiang T. W., Hau H. Y., Chiang H. M., Ko S. Y., and Hsieh C. H. (1998), Knowledge based system for railway scheduling. *Data and Knowledge Engineering*, **27**, 289-312.

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

Corry P. and Kozan E. (2004), Job scheduling with technical constraints. *Journal of the Operational Research Society*, **55**, 160-169.

Daniels R. L, Hua S. Y., and Webster S. (1999), Heuristics for parallel machine flexible resource scheduling problems with unspecified job assignment. *Computers and Operations Research*, **26**, 143-155

Dorfman M. J. and Medanic J. (2004), Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research B*, **38**, 81-98.

Ferreira L. (1997), Rail track infrastructure ownership: investment and operational issues. *Transportation*, **24**, 183-200.

Franca P. M., Gendreau M., Laporte G. and Muller F. (1996), A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times. *International Journal of Production Economics*, **43**, 78-89

Gibson S. (2003), Allocation of capacity in the rail industry. *Utilities Policy*, **11**, 39-42.

Gibson S., Cooper G., and Ball B. (2002). The evolution of capacity charges on the UK rail network. *Journal of Transport Economics and Policy*, **36**, 341-354

Goverde R. M. P. (1999), Improving punctuality and transfer reliability by railway timetable optimisation. P. Bovy, ed. *Proc. TRAIL 5th Annual Congress*, Vol. 2. Delft, The Netherlands.

Hall N. G., and Sriskandarajah C. (1996), A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, **44.3**, 510-525

Higgins A., Kozan E., and Ferreira L. (1996), Optimal scheduling of trains on a single line track. *Transportation Research B*, **30.2,** 147-161.

Higgins A., Kozan E., and Ferreira L. (1997). Heuristic techniques for single line train scheduling. *Journal of Heuristics*, **3**, 43-62.

Hooghiemstra J. S., Kroon L. G., Odijk M. A., Salomon M., Zwaneveld P. J. (1999), Decision support systems support the search for "win-win" solutions in railway network design. *Interfaces*, **29**, 15-32.

Khosla I. (1995), The scheduling problem where multiple machines compete for a common buffer. *European Journal of Operational Research*, **84**, 330-342

Kim D. W., Na D. G., and Chen F. (2003), Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. *Robotics and Computer Integrated Manufacturing*, **19**, 173-181

Kort A. F., Heidergott B. and Ayhan H. (2003), A probabilistic (max,+) approach for determining railway infrastructure capacity. *European Journal of Operational Research*, **148**, 644-661

Kroon L. G., Romeijn H. E., Zwaneveld P. J. (1997), Routing trains through railway stations: complexity issues. *European Journal of Operational Research*, **98**, 485-498.

Kroon L. G., and Peeters W. P. (2003), A variable trip time model for cyclic railway timetabling. *Transportation Science*, **37.2**, 198-212 (2003).

Lindner T. (2000), Train schedule optimisation in public rail transport. Ph.D. thesis, Technical University Braunschweig, Braunschweig, Germany.

Mascis A., and Pacciarelli D. (2002), Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, **143**, 498-517

Nawaz M., Enscore E. E., and Ham I. (1983), A heuristic algorithm for the m-machine n-job flow shop sequencing problem. *Omega*, **11.1,** 91-95

Nowicki E. (1999), The permutation flow shop with buffers: a tabu search approach. *European Journal of Operational Research*, **116**, 205-219

Odijk M. A. (1996), A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research*, **30**, 455-464.

Pinedo M. Scheduling Theory, Algorithms, and Systems. Prentice Hall Inc. New Jersey, USA (1995).

Sahin I. (1999). Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research B,* **33**, 511-534.

Steinhofel K., Albrecht A., and Wong C. K. (1999), Two simulated annealing based heuristics for the job shop scheduling problem. *European Journal of Operational Research*, **118**, 524-548.

Szpigel B. (1972), Optimal train scheduling on a single track railway. Operational Research`72, Ross, M. (ed), North-Holland, Amsterdam, 343-352

Van Laarhoven P. J. M., Aarts E. H. L., Lenstra J. K. (1992), Job shop scheduling by simulated annealing. *Operations Research*, **40.1,** 113-125

Zwaneveld P. J., Kroon L. G., Romeijn H. E., Salomon M., Dauzere Peres S., Stan P. M. van Hoesel, and Ambergen H. W. (1996). Routing trains through railway stations: model formulation and algorithms. *Transportation Science*, **30**, 181-194

Zwaneveld P. J., Kroon L. G., Van Hoesel S. (2001), Routing trains through a railway station based on a node packing model. *European Journal of Operational Research*, **128**, 14-33.