Downloaded from UvA-DARE, the institutional repository of the University of Amsterdam (UvA)
http://dare.uva.nl/document/172158

File ID	172158
Filename	Chapter 3: Detailed modelling of the ForkR junction

SOURCE (OR PART OF THE FOLLOWING SOURCE):

Type	Dissertation
Title	Dynamic delay management at railways: a Semi-Markovian Decision approach
Author	A. Al-Ibrahim
Faculty	Faculty of Economics and Business
Year	2010
Pages	xii, 233
ISBN	978 90 3610174 5

FULL BIBLIOGRAPHIC DETAILS:
http://dare.uva.nl/record/338751

Copyright

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)

Chapter 3

Detailed modelling of the Fork$_R$ junction

Let us now formulate the SMD model for the most common junction, the so-called Fork$_R$ junction. At this junction, trains arrive from R different directions and come together to share from that point onwards the same infrastructure.

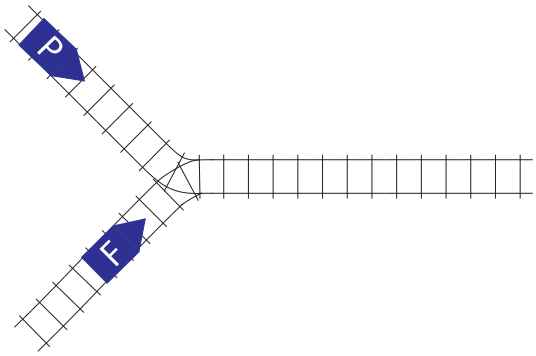


Figure 3.1: Fork$_2$ junction with a fast Intercity train (P) and a slow freight train (F) approaching

We will begin by describing the state space. Then in Section 3.2 the decisions will be outlined. In Section 3.3 the decision moments are discussed. Next the transitions, the most extensive part of the model will be explained. This will be done in Section 3.4. We will continue by defining the costs in Section 3.5. The discussion about how to reduce the state space will be conducted in Section 3.6. In Section 3.7 we will address the computational complexity of the SMD model and we will conclude the chapter by showing how the model can easily be extended to support some more complicated junctions where some train types leave the destination track earlier than others.

3.1 States

The state description is the most fundamental part of the model. The formulation of the state should incorporate all important elements while remaining compact. The size of the state space directly influences the computational complexity of the model.

In Chapter 2 we have spoken about the elements of the junction. We have learned that a typical junction consists of two types of tracks: arrival tracks and a destination track. The trains approach the junction via the arrival tracks and move towards the destination track after crossing the junction. The junction consists then of R arrival tracks, numbered 1 up to R and of one destination track numbered 0. There are S different train types denoted by $s \in \{1, \cdots, S\}$.

In order to optimize the fork junction one needs to know the location of every train that is within the scope of the junction. The type of these trains is also important. Fast trains have another impact on the junction than slow trains. Further, the current speeds of these trains and some information about the current track situation are needed. We introduce two variables that together will provide us with this information.

To begin with, the variable x will provide us with the positions and the types of all trains in the system. This variable is an array of smaller arrays x_0, x_1, $\cdots$, x_R, each denoting the situation on a single track. These track-related variables have the following structure $x_r = (x_{r1}, x_{r2}, \cdots, x_{rN_r})$. The variable x_r is thus an N_r dimensional array. The definition of this variable depends on the type of the track it represents. This is because the arrival tracks are modelled as queues while the destination track is not.

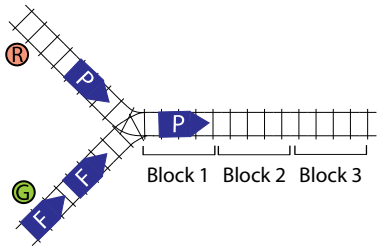
When x_r represents the destination track (i.e. $r = 0$), each entry refers to the block position of that track. The dimension of the array is the number of blocks the destination track is divided into. If a certain block is empty, the corresponding entry will be zero, otherwise the entry is equal to the type of the train that occupies that block. Thus the entry x_{ri} can have values in the range of $\{0, \cdots, S\}$ and denotes the type of the train that occupies block i of the destination track r.

If the variable x_r refers to an arrival track (i.e. $r > 0$) then the definition is slightly different. The dimension N_r varies and is equal to the number of trains on that track. Every element refers to the position of the train in the queue of the arrival track r (e.g. $x_{r1} = s_1$, $x_{r2} = s_2$ means that on arrival track r a train of type s_1 is followed by train of type s_2). In this case entry x_{ri} can only have values in the range of $\{1, \cdots, S\}$. If there are no trains on the track then $x_r = ()$ is an empty vector.

Now, we know the position and the type of every train in the system, we still need another variable for providing us with the train speed information. We introduce variable

y to this end. The variable y is an array of $(y_1, \cdots, y_R)$ where every element y_r marks the degree by which the trains on that track are hindered by the train that is crossing the junction. A non negative integer value indicates this degree of hindrance; the lower the value, the more hindrance. The idea behind this is that the trains move according to their speed profile unless they are hindered by other trains in which case they move slower. The more hindrance the train experiences from other train the slower the train will move. In the most extreme case the train will get to a complete stand still and will have to wait until the train that caused hindrance has left. The state space is the combination of the variables x and y.

Example:



Consider the Fork$_2$ junction as depicted. A passenger train (P) is located at arrival track 1 while two freight trains (F) are located at arrival track 2. The destination track is divided into 3 blocks of 4 kilometre each. Block 1 is occupied by a passenger train (P) while the other two blocks are empty. Next, consider that upon a conflict, a train is either stopped or may proceed, there is nothing in between. In this example, the traffic on arrival track 1 is stopped (the track is labelled red (R)), while the traffic on arrival track 2 is not hindered (the track is labelled green (G)).

The state of the example is denoted by $x_0 = (P, 0, 0)$, $x_1 = (P)$, $x_2 = (F, F)$, $y_1 = R$, $y_2 = G$.

3.2 Decisions

Let us spend some time on the kind of decisions that the model should optimize. The model should be able to solve the train conflicts and come up with an optimal train order which optimizes some criteria. Thus, in case of a Fork$_R$ junction where trains from R different directions come together, the decision should be to give one of these trains the right of way first. Then, when this train has crossed and cleared the junction, it has to be decided which of the trains is to be 'served' next.

We introduce the variable a (a stands for action) for this purpose. When the variable has a positive value, the train of track $r = a$ may proceed. If all arrival tracks are empty, then there is no train that can cross the junction, then a is equal to 0. The valid range of the values for the decision a is thus $\{0, \cdots, R\}$.

3.3 Decision moments

After a decision is made to give a certain train the right of way, the junction becomes blocked for other trains. The train, that has received the permission, will then cross the junction. Meanwhile, no other decisions can take place. The time that the train needs to approach the junction, cross it and clear it for other trains depends on the train type and the current speed of the train. Let us denote this time by the variable $\tau(x, y, a)$, which clearly depends on the current state (x, y) and the decision a being taken.

If in state (x, y) decision a is taken, the junction is blocked for the next $\tau(x, y, a)$ time units. The state of the system will not be reviewed in the mean time, therefore, the system jumps $\tau(x, y, a)$ time units forward. The values of $\tau(x, y, a)$ are deterministic given the combination (x, y, a) but vary per (x, y, a).

3.4 Transitions

By transition we mean the state change between the time instance when the system is viewed and the next review point in time. The transition from the current state (x, y) to a new state is the result of a decision a being taken and the corresponding time advance of $\tau(x, y, a)$ time units. This transition involves many changes. We will describe these changes one by one. In global lines, we can say that a transition consists of the following phases:

1. The destination track phase: Trains on the destination track $r = 0$ change their position.

2. The junction crossing phase: The train that received permission to cross the junction, crosses it and enters the destination track. The speed indicators of the arrival tracks are updated.

3. The new arrivals phase: new trains arrive at the arrival tracks.

The order in which the phases are executed, is the order presented above. First, the trains on the destination track move forward which frees the first position on the track. This position is then used by the train that crosses the junction which in turn creates extra space at the arrival tracks where new trains can enter.

If the decision is $a = 0$, i.e. no train crosses the junction, then the time advances with headway h time units. Note that in this case the junction crossing phase is still part of the transition since the speed indicators are to be updated.

3.4.1 The destination track phase

The destination track phase is essentially the phase that takes care of the train movements on the destination track $r = 0$ without new arrivals to this track taking place. Recall that x_0 denotes the situation at the destination track, and is defined as $x_0 = (x_{01}, x_{02}, \cdots, x_{0n})$, where x_{0i} for $i \in \{1, \cdots, n\}$ is the train type that is found on block i of the destination track. If the block is empty then the corresponding x_{0i} is zero. Since all blocks have the same length, the train on block i has already travelled $(i-1) \cdot l$ kilometres where l is the length of the block of the destination track. Figure 3.2 gives the graphical representation of the array.

x_{01}	x_{02}	...	x_{0n}

Figure 3.2: Graphical representation of x_0

When the time advances, the position of the trains changes. The movement of each train depends on the speed of the train, the length of the block and the speed and the position of the predecessor trains. Through the concept of 'coupling' it is enforced that no overtakings take place and that the trains with equal speeds, remain at the same distance from each other. Moreover, the coupling ensures that no more than one train at a time can be found at a block. The concept of the 'coupling' has been explained in Section 2.2.7.

3.4.2 The junction crossing phase

Definition By the junction crossing we mean the movement of trains from an arrival track (i.e. $r > 0$) to the destination track $r = 0$. This movement depends on the decision taken. Decision a can take values between $\{0, 1, \cdots, R\}$ with the restriction that $a = r$ with $r > 0$ is possible only if track r is not empty ($N_r > 0$). When a is positive, the train is moved from track $r = a$ to track $r = 0$. The train type that is moved to track $r = 0$ is then x_{a1} and the amount of hindrance the train experienced on that track is y_a.

We can identify three elements that together form the junction crossing phase:

1. Train leaves the arrival track
2. Train enters the destination track
3. Update of the speed indicators at the arrival tracks

Note that if $a = 0$ then no train crosses the junction. The junction crossing phase reduces to only one action, that is updating the speed indicators of the arrival tracks.

Train leaves the arrival track The essence of this phase is simple: the first train of the queue a leaves the queue. All other trains in that queue move one position towards the front of it.

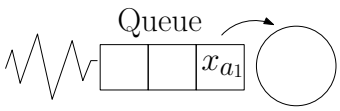


Figure 3.3: The crossing of the junction by the train x_{a1}

Train enters the destination track The train, that receives permission to cross the junction, will need $\tau(x, y, a)$ time units to arrive at the destination track. Within this time interval the train has approached the junction, crossed it and cleared it for other trains. During this time the trains, that were already on the destination track, have moved. This transition has already been explained in Section 3.4.1. The only change to the destination track that still needs to be done is putting the train that has crossed the junction on the first block of the destination track. The value of x'_{01} is thus set to x_{a1}.

Update of the speed indicators Recall that the y-part defines the state of the arrival track. It defines the degree of hindrance the trains on the arrival track suffer from the train that crosses the junction. Let us first consider the easiest case where Y_r's have only two possible values: 1 indicating that the trains on track r are not hindered and are moving with their planned speed and 0 indicating that the trains on track r are standing still.

After decision a is made, the trains on track $r = a$ will get moving. The value of y_a is set to 1. The trains at other arrival tracks will have to stop and wait until their turn comes to cross the junction. The value of y_r for $r \neq a$ is set to 0. The arrival tracks that do not have any trains on them are per definition not hindered. The value of y_r for those tracks is then 1. In short:

$$y_r = \begin{cases} 1 & \text{if } r = a \\ 1 & \text{if } (r \neq a) \text{ and } (N_r = 0) \\ 0 & \text{if } (r \neq a) \text{ and } (N_r > 0) \end{cases} \tag{3.4.1}$$

Now, let us think of an extension to this model. Suppose that there are n different values of hindrance. If $y_r = n$ then the trains on track r are not hindered. If $y_r = n - 1$ then the trains are slightly different. If $y_r = 0$ then the trains are maximally hindered and

as a result are standing still. If two identical trains are approaching the junction from two different directions and train A receives permission to cross it, train B will need to lower its speed a bit. With this new speed the train suffers only a limited hindrance. Train A crosses the junction and train B can accelerate again to its desired speed. If however, in the meantime, train C enters the track and receives permission to cross the junction before train B crosses it, the hindrance of train B increases as it needs to adjust its speed again. This situation is depicted in the following figure:

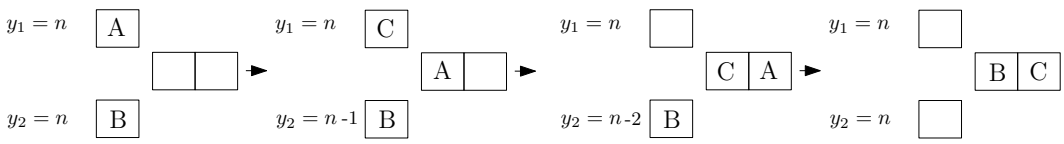


Figure 3.4: The evolution of the speed indicator variable y (assuming that $n \geq 2$)

Of course, this can be extended even more since the amount of hindrance on train B by giving permission to train A can depend on the train types of the trains A and B. In this case the speed indicator y'_r will be equal to $y_r - d_{AB}$ where d_{AB} is the amount of hindrance that train B suffers when train A may go first. Formally, the speed indicator is calculated as follows:

$$y'_r = \begin{cases} n & \text{if } r = a \\ n & \text{if } (r \neq a) \text{ and } N_r = 0 \\ y_r - d_{x_{a1},x_{r1}} & \text{if } (r \neq a) \text{ and } (N_r > 0) \text{ and } (N_a > 0) \\ y_r - d_{0,x_{r1}} & \text{if } (r \neq a) \text{ and } (N_r > 0) \text{ and } (N_a = 0) \end{cases} \tag{3.4.2}$$

Example (continued):

Consider again the example on page 49. Given the values of the speed indicators, used in the example, and the fact that there are only two values possible (hindered, and not hindered) one can deduce that at the earlier stage the two passenger trains must have been in conflict with each other. Otherwise, the current state of the speed indicators could not have been reached.

3.4.3 The arrival process

Between the moment the decision has been taken, and the next decision can take place, new trains can arrive at the arrival tracks. The arrival process follows the $\mathcal{HP}$-process which has been explained in Section 2.2.2.

The $\mathcal{HP}$-process defines the probabilities of the arrival of a certain number of trains to a certain track. To calculate the probability of an arrival of a certain combination of trains to that track when in total n trains have arrived on that track, we use binomial probabilities conditioned on the probability of n arrivals. Next we combine these probabilities with the arrival probabilities of other tracks. This way, the arrival probabilities to the junction as a whole are derived.

We want to stipulate that the arrival process is the last process of the transitions, i.e. the decision has already taken place, the trains on the destination track have already moved to their new positions and the train that has received the permission to cross the junction is already on block 1 of the destination track.

3.4.4 Summary of the transitions

In the previous sections the transition process has been explained that takes place whenever a decision a is taken. In this section we will summarize this process. In Figure 3.5 the flowchart of the transition process is given. The process starts when decision a is made. At this time, the value of the time jump $\tau(x, y, a)$ can be determined (process 1.1 in the figure) after which the destination track phase begins. During this phase, trains on the destination track $r = 0$ move according to their speeds. This results in a number of possible new states $x_0'^{(i)}$ with corresponding probabilities $p_d^{(i)}(x, y, a)$. Next, it is checked whether the decision involves $a > 0$. If a is indeed non-zero, then the first train of track $r = a$ crosses the junction and enters the destination track (process 1.3). On the other hand, if $a = 0$, no train makes the transition to the destination track and process 1.3 is omitted. The junction crossing phase ends with an update of the speed indicators (process 1.4). Finally, new trains arrive at the arrival tracks.

As a consequence the current state (x, y) through decision a leads to a number of possible new states (x', y'). We will denote these transition probabilities by $p((x, y), a, (x', y'))$ which are basically a combination of the destination track probabilities (p_d) and the arrival probabilities.

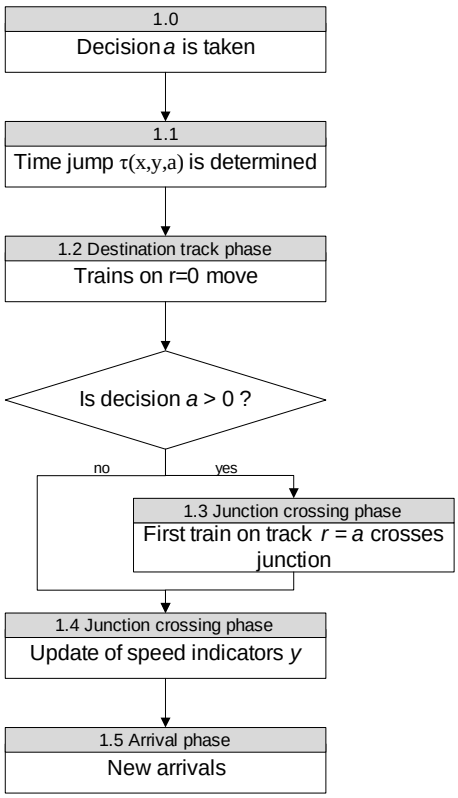


Figure 3.5: Flowchart of the transition process

3.5 Costs

Choosing the best decision for a given state requires choosing among alternatives based on some criteria. In Section 1.4 we already discussed a number of criteria that could be used for this purpose. While it is possible to incorporate all kinds of criteria into the SMD model, in this thesis we will focus on minimizing the total traverse time of the trains. The idea is that minimizing the traverse time of the trains will both optimize the capacity of the junction and indirectly minimize the total train delays. From now on, we will speak of traverse time (or the stay time of trains) as being the costs that need to be minimized.

We will introduce three cost components that together form the total costs. These are the *arrival track costs*, the *destination track costs* and the *train rejection costs.*

Arrival track costs The first cost component relates to the time, trains need to approach the crossing. These costs reflect the stretch of time beginning at the time instant the train enters the arrival track and ending at the instant the train has crossed the junction.

According to this definition, every time a decision is taken, the trains at the arrival

tracks are counted. This number is then multiplied by the length of the time jump. Let $c^a(x, y, a)$ be the arrival track costs, then we can express these costs as

$$c^a(x, y, a) = \tau(x, y, a) \cdot \sum_{r=1}^{R} N_r \tag{3.5.1}$$

Destination track costs The destination track costs are the costs that represent the stay time of the trains on the destination track. For this end, one could use the same tactics we used for the calculation of the arrival track costs, i.e. counting all the trains on the destination track every time a decision is made and multiplying it by the time jump, but doing so will lead to an overestimation of the stay time of the trains. In the most extreme case, some decision is taken which involves a time jump of $\tau(x, y, a)$ time units while there is a train at the end of the track, which according to its position and speed will leave the model in a fraction of $\tau(x, y, a)$. In this case, counting the number of trains at the destination track of the state (x, y) will result in the overestimation.

In order to get around this, we will calculate the stay times differently. Each time a decision is taken to let some train cross the junction, its total stay time on the destination track is calculated at once. This stay time can be computed since the position and the speeds of the trains on that track together with the position and the speed of the train that is crossing the junction are known.

Let us denote the destination track costs by $c^d(x, y, a)$ and let us number the blocks on the destination track by 1, 2, $\cdots$, N where N is the number of blocks at the destination track. Block 1 being the closest to the junction and block N the furthest. Let further x_i be the i-th train on the track where x_1 is the train furthest to the track and x_M the last train that has entered the track. Moreover, denote the time that train x_i departs from block b by $T(x_i, b)$, then the following is true: (1) based on the speed of the train x_i and the length of the blocks, the train needs $T^{block}(x_i)$ time units to cross one block and thus

$$T(x_i, b+1) - T(x_i, b) \geq T^{block}(x_i) \quad \forall i, b \tag{3.5.2}$$

and (2) since by regulation, the trains must be separated in time from each other, two subsequent trains need to respect the headway h when departing from a block, thus

$$T(x_{i+1}, b) - T(x_i, b) \geq h \quad \forall i, b \tag{3.5.3}$$

Then, the train x^* that is to enter the destination track, may not leave the destination track sooner than (1) the time it needs to cross the destination track ($T(x^*, N)$) and (2) this time should respect the minimal headway to its predecessor ($T(x_M, N)+h$) and thus:

$$c^d(x, y, a) = max(T(x^*, N), T(x_M, N) + h) \tag{3.5.4}$$

Costs of rejecting a train In Section 2.2.8 the idea behind the train rejection costs have already been explained. In this section we will show what these costs look like.

The service time b_s of the trains of type s is equal to the time the train of that type needs to approach the junction, cross it and clear it for other trains. The waiting time w for the rejected train will then be approximated with the following expression:

$$w = \sum_{s=1}^{S} Q_s b_s + \rho R + w \sum_{s=1}^{S} \lambda_s b_s \tag{3.5.5}$$

where Q_s is the number of trains of type s in the queue, b_s is the service time of train of type s, R is the residual service time and ρ is the load of the junction. The above expression can be rewritten to:

$$w = \frac{\sum_{s=1}^{S} Q_s b_s + \rho R}{1 - \rho} \tag{3.5.6}$$

The total stay time of the rejected train of type s is then equal to $w + \frac{b_s}{1-\rho} + u_s$ where u_s is the time the train needs to cross the destination track.

As has been explained is Section 2.2.8 the values of b_s in Equation 3.5.6 and ρ are calculated iteratively. But within an iteration, these values are set.

Total costs The total costs of choosing decision a when in state (x, y) are then equal to the sum of the arrival track costs, the destination track costs and the train rejection costs

$$c(x, y, a) = c^a(x, y, a) + c^d(x, y, a) + c^r(x, y, a) \tag{3.5.7}$$

3.6 State space reduction

To reduce the size of the problem it is interesting to examine the different states. It turns out that not all states of the destination track are that important. Some states are

essentially equivalent. Recall that the costs the trains on the destination track incur are calculated at the moment the trains enter the destination track. Thus, the only purpose of keeping these trains on the destination track is because these trains may cause hindrance to the trains that will enter the track in the future. In some cases however, the trains on the destination track have travelled sufficiently far as to not form a hindrance any more. These trains may safely be removed from the destination track without affecting the optimal strategy. As a result the destination track will have much fewer states especially when long destination tracks are considered.

To find the 'redundant' trains we need to calculate for each train the time it needs to reach the end of the track. If this time is not effected by its predecessor trains then we can safely remove all preceding trains. Moreover, the remaining trains may be removed as well if these will not be able to affect any possible train that can enter the track in the immediate future, that is, if the trains will not hinder even the fastest train that can enter the track already after the minimal headway time units.

This is calculated in the following way, let us number the blocks on the destination track by 1, 2, $\cdots$, N where N is the number of blocks at the destination track. Block 1 being the closest to the junction and block N the furthest. Let further x_i be the i-th train on the track where x_1 is the train furthest to the track and x_M the last train that has entered the track. Moreover, denote by $T^{desire}(x_i, b)$ the time that train x_i desires to depart from block b, based on its speed and the length of the blocks. If we denote by $T^{block}(x_i)$ the time that train x_i needs to cross one block depending on its speed and the length of the block then

$$T^{desire}(x_i, b+1) - T^{desire}(x_i, b) \geq T^{block}(x_i) \tag{3.6.1}$$

Now, let $T^{can}(x_i, b)$ be the time that the train can depart from block b when respecting the departure time of the predecessor trains, i.e.

$$T^{can}(x_i, b) = max(T^{desire}(x_i, b), T^{can}(x_{i-1}, b) + h) \tag{3.6.2}$$

Then the trains x_{i+1} up to x_M may be safely removed from the state if the following holds

$$T^{desire}(x_i, N) = T^{can}(x_i, N) \quad \forall i \tag{3.6.3}$$

Furthermore, all trains, i.e. $x_1, x_2, \cdots, x_M$, may be safely removed from the state if

$$T^{desire}(x^*, N) \geq T^{can}(x_1, N) \tag{3.6.4}$$

where x^* is the fastest train that can enter the destination track. Note, that in this equation the term h is emitted, this is due to the fact that the next arrival can take place no sooner than after h time units. Thus, the term eliminates.

3.7 Computational complexity

The computational complexity is determined by the number of states, the number of transitions per state, the number of possible decisions and the convergence rate of the iteration process. The number of states is defined by the total number of tracks R, the capacity of these tracks, the number of train types S and the number of speed indicators per arrival track y.

The destination track is divided into n_r number of blocks. Each block can be occupied either by a train of some type or the block can be empty. So the total number of possible states of the destination track before reduction is equal to $(S+1)^{n_r}$. The arrival tracks are modelled as queues. Each track thus can be either empty, or occupied by a certain number of trains. If an arrival track r has a capacity for a maximum of n_r trains (so the track is a queue with capacity n_r) and the track is used by S_r different train types then the total number of states at this queue is $1 + S_r + {S_r}^2 + \cdots + S^{n_r} = \frac{{S_r}^{n_r+1}-1}{S_r-1}$. In addition, every arrival track can be in one of y states of the speed indicator which accounts for another y^R possible states.

The number of transitions is state and action dependent. This number is the result of the train movement on the destination track and the number of possible new arrivals. If all the trains on the destination track move an integer number of blocks forward then there is only one possible new state of the destination track. However, often the trains will move a non-integer number of blocks forward. In this case, each movement is modelled by means of probabilities. The train either moves above average (rounded up) number of blocks forward or the train moves below average (rounded down) number of blocks forward. Since, different train types can be found on the destination track simultaneously, this can lead to 2^S number of transitions. Due to coupling, however, this number is substantially lower. The bigger contribution to the number of transitions form the new arrivals. Recall that these arrivals follow the $\mathcal{HP}$-process and depend on the time jump $\tau(x, a)$.

Example (continued):
Consider again the example on page 49. In the example two types of trains arrive at the junction from two arrival tracks. Both arrival tracks have a capacity of 2. The number of possible states on each arrival track is $\frac{S_r{}^{n_r+1}-1}{S_r-1} = 7$. The number of possible states at the destination track is $(S+1)^{n_r} = 3^3$, but as has been said in Section 3.6, this number can be brought back. Since only two possible values of the speed indicator are considered per track, the number of possible speed indicators is 2^2 which brings the total number of states to $7^2 \cdot 3^3 \cdot 2^2 = 5292$.

3.8 Trains with different destinations

Now we have explained the model let us consider an important extension which will be very useful in practice. Suppose that n train types enter the destination track after crossing the junction but one of these types does not travel all the way until the end of the destination track but will leave the destination track at some point to continue its journey to another destination. This situation is depicted in the following Figure 3.6.

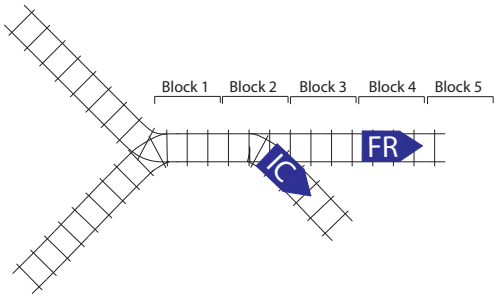


Figure 3.6: IC train leaves the destination track halfway block 2, while the FR train will reach the end of the destination track halfway block 5

To support this kind of situations only a little adjustment needs to be done.

- The destination track costs of the train should be calculated only for the distance the train crosses on the destination track and not over the full length of the destination track.

- The train will leave the model when crossing block i where $i = \lceil \frac{distance\ the\ train\ travels\ on\ the\ destination\ track}{length\ block} \rceil$.

- When calculating the destination track costs, the trains that are already at the destination track are only taken into account until the last block of their route on the track and not until the last block of the destination track.

Regarding the second adjustment, the following remark should be made: even if the train exits the destination track halfway through the block, the arrival time of that train at that block is still used when computing the time the next train can enter that block.

Note, that the very same principle can be used when the sum of the length of the blocks exceeds the total length of the destination track. Then all trains exit the last block prematurely at the distance that corresponds with the end of the destination track.