CASPT 2015

An optimisation framework for determination of capacity in railway networks

Lars Wittrup Jensen

Abstract Within the railway industry, high quality estimates on railway capacity is crucial information, that helps railway companies to utilise the expensive (infrastructure) resources as efficiently as possible. This paper therefore proposes an optimisation framework to estimate the capacity of a railway network based on a mix of train types, the infrastructure and rolling stock used. The framework consist of two steps. In the first step the maximum number of trains is found according to the predefined mix of train types. In the second step additional trains are added based on weights assigned to the train types. This is done using a mathematical model which is solved with a heuristic. The developed approach is used on a case network to obtain the capacity of the given railway system. Furthermore, we test different parameters to explore computation time, precision and sensitivity to input of the approach.

Keywords Capacity · Networks · Railways

1 Introduction

Railway capacity is a scarce resource that has to be utilised in the best way possible to satisfy demand and provide the best service for passengers and freight customers. However, the capacity of a railway system is not easily determined or even defined. This is a consequence of the interdependencies in railway systems as capacity depends not only on the infrastructure, but also on operational constraints and the rolling stock used.

In this paper, we propose an optimisation framework to estimate the capacity in a railway network, and thus provide the basis for better utilisation of capacity. We define the capacity as the number of trains of each train type

Lars Wittrup Jensen (Corresponding Author)
Department of Transport, Technical University of Denmark
Bygningstorvet 116B, 2800 Kg. Lyngby, Denmark
Tel.: +45 45256696
E-mail: lawj@transport.dtu.dk

that is able to transverse the network under the predefined mix (of train types) plus trains that additionally can be added. The solution is subject to the given infrastructure, the train types, characteristics of the train types, a desired mix of train types and the routes used within a defined threshold, C_{max}, (to account for robustness). Following definitions from existing literature (Abril et al, 2008; UIC, 2004), it is possible to estimate both the theoretical and practical capacity with the framework by adjusting the threshold (C_{max}). Respectively, theoretical capacity is the maximum capacity that can be utilised under perfect circumstances and practical capacity is the capacity that can be utilised in daily operation. In this paper, we will mainly focus on practical capacity. The proposed model does not require a timetable and is thus suitable for the strategic level to evaluate infrastructure alternatives and provide capacity estimates for e.g. line planning.

2 Previous and related approaches

Within the railway field, there already exist several methods to analyse railway capacity. One of the most well-known is the UIC 406 method (UIC, 2013). This method can be used to calculate the capacity consumption of line segments and routes based on a timetable. The method is used throughout Europe due to its simplicity. Contrary to the aim of this paper, the UIC method cannot be used to determine the capacity of railway system on its own. It can, however, be extended in many ways. For instance by saturating a timetable with additional trains. Or couple it with timetable planning methods to form a closed loop where a timetable is generated and subsequently assessed with the UIC 406 method.

Timetabling within railways is very related to capacity due to a large dependency between the two. The timetable problem, that is generating a feasible (and possible optimal) timetable, is a topic given much research attention. Due to the dependency between timetabling and capacity, timetabling methods can be used indirectly to estimate capacity. However, this is not an ideal approach as the resulting capacity will rely heavily on the input given to the timetable generator and thus the structure of the generated timetable. Additionally, long computation times makes it undesirable to run a timetable generator a given number of times to find the maximum capacity.

In addition to timetable generators, many models and tools exist that can be used to saturate existing timetables (which can be empty) with additional trains, locate bottlenecks and estimate stability of a generated timetable. These can be used to estimate the number of trains that can traverse a given network under a set of operational constraints. However, these models are more suitable on a tactical level as they require a significant amount of input. For an overview of these models and tools we refer to Abril et al (2008).

Recently, the work of Burdett and Kozan (2006); de Kort et al (2003); Mussone and Wolfler Calvo (2013), deals explicitly with the determination of capacity in railways. These approaches are suitable for strategic planning as they require no timetable. de Kort et al (2003) uses a probabilistic approach using max-plus algebra to determine the capacity of railway infrastructure. The approach is based on the bottleneck approach which determines the critical section (bottleneck) that limits the capacity of the system considered. The approach does not explicitly account for different train types. However, it is possible to implement this by using the probability that train will be of a given type. Inspired by the work of de Kort et al (2003), Mussone and Wolfler Calvo (2013) present an optimisation framework to maximize the total number of trains in the railway system based on train conflict probabilities, where different train types can be considered. de Kort et al (2003); Mussone and Wolfler Calvo (2013) are able to include knock-on delays in the railway system in a simple way. Lastly, Burdett and Kozan (2006) describe and discuss terms for absolute capacity and utilisation levels based on sectional running times of train types, dwell times and proportional mix of train types. Based on these, they propose an optimisation model to maximize the number of trains in a railway network. The model proposed is non-linear and is therefore only solved to a local maximum (Burdett and Kozan, 2006). The approach by Burdett and Kozan (2006) is quite flexible and is able to capture a large amount of operational characteristics. Furthermore, the approach is able to give a lower and an upper bound for capacity. Common for the approaches by de Kort et al (2003); Mussone and Wolfler Calvo (2013) is that they do not depend on the exact train sequence, but rather consider a weighted average. Thus, in a network with heterogeneous operation the average absolute capacity is obtained, and this capacity might therefore be increased by bundling trains.

3 Method

Unlike existing methods, described in the previous section, our optimisation framework is based on a model that estimates the capacity consumed by a given set of trains. Using this model in our optimisation framework makes it possible to capture the distribution of capacity in railway networks with heterogeneous operation as the capacity can be measured relative to how many trains sequences should be feasible. The capacity consumption model used in the framework is briefly described in the following section 3.1, while the optimisation framework is described in section 3.2.

3.1 Model for calculation of capacity consumption

In the following section, we shortly describe our model used to calculate the capacity consumed by a given set of trains. For a more in-depth description of the model see Jensen et al (2015).

The purpose of the model is to calculate the capacity consumed by a given set of trains in a railway network. We define the capacity consumption as the ratio between the time a train mix in a given sequence occupy the network

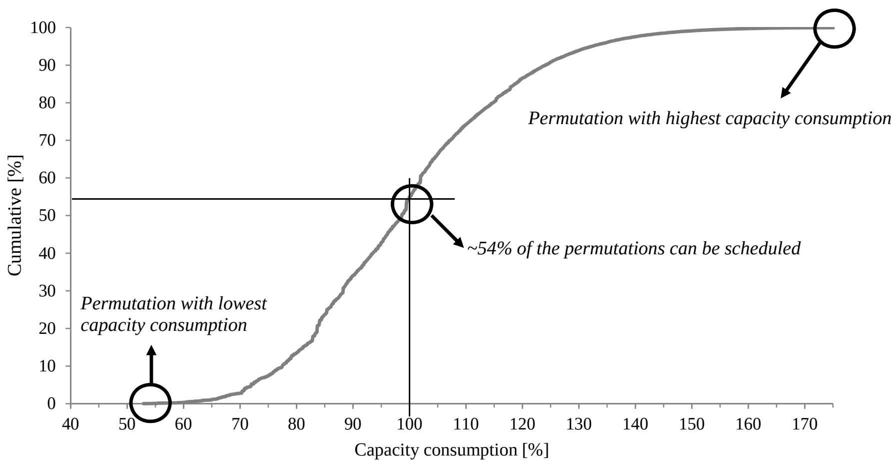

Fig. 1 Cumulative capacity consumption distribution.

and the time period considered. This is similar to the compression method to obtain infrastructure occupation known from the UIC406 method (UIC, 2013). In a network with different train types and heterogeneous running times the capacity consumption will differ depending on the sequence (order) of the trains considered. For instance, if trains of the same type are bundled the consumption is low, while the consumption is high if they are mixed (unbundled). As capacity consumption differs depending on the sequence of trains, the capacity consumed by a given set of trains in no particular order is therefore a distribution of capacity consumption rather than a single value.

An example of a capacity consumption distribution is shown in figure 1. The y-axis show the cumulative percentage. This percentage show how many of the possible permutations of train sequences that fit within the capacity consumption depicted on the x-axis. In the remainder of this paper, we will denote the cumulative percentage as the the percentile, p, while the capacity consumption is denoted by C. For a specific set a trains X the maximum capacity consumed at a percentile, p, is denoted as $C(X, p)$. For instance in figure 1, approximately half of the analysed train sequences are feasible ($C(X, 54) = 100\%$). That is, the 54th percentile (close to the median) is below 100% capacity consumption, where sequences yielding capacity consumption values larger than 100% are infeasible.

For our capacity consumption model, the input is the number and type of trains, the network, routes as well as minimum headway times and running times. The infrastructure is represented by a mesoscopic infrastructure model that makes it possible to model double and single track as well as junctions.

For each possible unique train sequence, the trains are scheduled strictly according to this train sequence with fixed train routes, and the capacity consumed is measured. The model is not dependent on the exact timetable as an input, as every permutation of train sequences are considered. The output is the capacity consumed by each unique train sequence and thus a distribution as depicted in figure 1 when train types are heterogeneous.

For larger problem instances every permutation cannot be calculated due to factorial computational complexity. In that case sampling of random permutations is therefore used to reduce computation time. Furthermore, some sequences are not desirable from a demand point of view and can therefore be discarded.

Train sequences yielding capacity consumption values of 100% or close will not be feasible in practice due to the stochastic behaviour of railway systems (resulting in delays and thereby higher consumption). The aspect of robustness is included in the model by using a stochastic simulation. In each iteration of this simulation, initial delays are added to the deterministic output of the model and the delay propagation is calculated. Initial delay perturbations are obtained by sampling from one or more delay distributions. The output is the capacity consumption including delay. If the (stochastic) capacity consumption result is higher than 100%, the consumption of capacity is not robust. However, in the optimisation framework presented in this paper, we only use the deterministic version of the model due to significantly increased running times in the stochastic version.

3.2 Optimisation framework

The optimisation framework for the determination of capacity, proposed in this paper, is shown as a flowchart in figure 2. As stated in the introduction (section 1), we define the capacity as the number of trains respecting a given train type mix plus additional trains that the network can handle under a predefined threshold, C_{max}. A high threshold will result in a less robust system than for lower thresholds as there will be less buffer times. As mentioned earlier, the optimisation framework relies on the capacity consumption model described in section 3.1. However, the framework can be used with any method or model that evaluates the capacity consumed by a given set of trains in a given network. Although, it will not be possible to derive the capacity span if the alternative model is not able to calculate the distribution of capacity consumption. If this is indeed the case, existing models may be more suitable (see section 2). If an alternative method is used for capacity consumption determination in the optimisation framework it must fulfill the following requirements:

- If an extra train is added to a network the capacity consumed will never decrease compared to the previous solution (without the extra train)
- Likewise, if a train is removed the capacity consumed will never increase

Not all models fulfills these requirement. For instance, models where scheduling decisions is made by slowing down fast trains and/or re-routing trains, the addition of a train might lead to a lower capacity consumption.

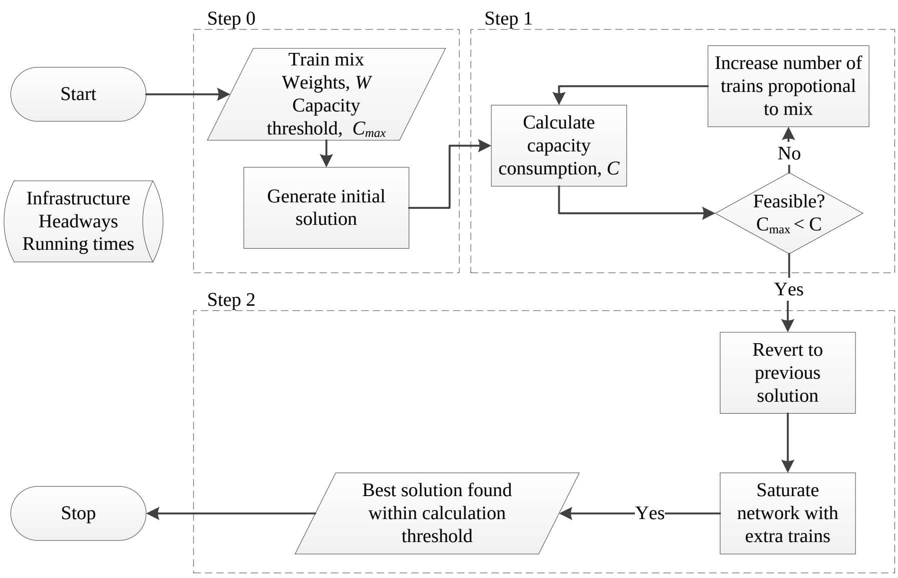

Fig. 2 Proposed framework.

The proposed optimisation framework consists of three steps which will be described in the following:

Step 0: In this step, the data is loaded and an initial solution is generated. This initial solution generated is the minimum solution respecting the given mix. For instance with four train types constrained to a mix of 20-30-10-40%, the initial solution will be 2, 3, 1 and 4 trains of each type, respectively.

The capacity consumption threshold, C_{max}, is an important input to the framework. This will be 100% if the maximum theoretical capacity has to be obtained or less if the maximum practical capacity has to be obtained. As mentioned earlier, the maximum practical capacity is the maximum capacity that can be utilised if the operation has to remain stable when smaller delays occur. The UIC (2013) has made recommendation for maximum capacity consumption for stable operation. For instance, 75% in the peak hour for heterogeneous operation.

As described in section 3.1, the output from the capacity consumption calculation is a distribution of capacity consumption when train operations are heterogeneous. Thus in the optimisation a percentile, p, of this cumulative distribution has to be chosen to obtain one single capacity consumption figure, C. The 0th percentile will yield the maximum number of trains (upper bound), while the 100th percentile will yield the minimum number of trains (lower bound). A natural choice would be the 50th percentile or the 100th percentile (lower bound). Due to sample variance it is however not recommended to use the 0th and 100th percentiles. A span of capacity, as multiple of predefined mix plus additional trains, can be derived by executing the optimisation several times with different percentiles, for instance for the 5th, 50th and 95th percentiles.

Step 1: In this step, the capacity respecting the train mix is estimated. First the capacity consumption of the initial solution is calculated. If the capacity consumption calculated, C, is higher than the threshold C_{max}, it is not possible to schedule the given train mix with the threshold, C_{max}, and percentile, p, chosen. However, it will still be possible to add trains in step 2.

The train mix is given for the calculations in step 1, and thus the solution space will usually be quite small, due to the large train increments necessary to respect the train mix. A simple incremental approach is therefore used as the number of iterations are very few (in most cases below 5).

Step 2: As described in step 1, the train mix induces large increments in the number of trains. There might therefore be a large excess of capacity as the last feasible solution found in step 1 might have a capacity consumption, C, well below C_{max}. Thereby, we assume that adding a train of type i, will lead to a benefit of w_i. In step 2, we therefore saturate the solution from step 1 with additional trains. For this purpose we use the following mathematical model:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n}(w_i \cdot x_i) \\ \text{subject to} \quad & x_i \geq 0 \qquad\qquad\qquad \forall i \in \{1, 2, ..., n\} \\ & C(X+Y, p) \leq C_{max} \end{aligned}$$

Where $X = (x_1, ..., x_i, .., x_n)$ is the decision variables for step 2. That is the number of extra trains of each train type. $Y = (y_1, ..., y_i, .., y_n)$ is the number of trains found in step 1. w_i is the weight of each train type, with $1 \geq w_i \geq 0$ and $\sum_{i=1}^{n}(w_i) = 1$. The percentile, $p \in \{0, 1, ..., 100\}$, determines the ratio of possible sequences that should consume capacity less than or equal to the capacity threshold, C_{max}. As described earlier, a high percentile will yield a lower capacity result as more train sequences has to be feasible compared to a low percentile. $C(X+Y, p)$ is the function (the model described in section 3.1 or an alternative) that calculates the capacity consumption given a set of trains, $X+Y$, and a percentile p.

The optimisation problem defined above looks simple. Unfortunately, it is not straightforward to solve since the constraint $C(X+Y, p) \leq C_{max}$ is computationally expensive to evaluate. Thus, to solve the problem we need to construct an algorithm that finds good solutions, and desirably the optimal solution, with a minimum amount of evaluations of $C(X+Y, p)$.

It is well known that a homogeneous train mix provides the possibility to run more trains than a heterogeneous train mix. Based on this characteristic, we use a greedy heuristic to search for solutions with as many trains of type as possible. The greedy heuristic is described in pseudocode in algorithm 1. The search works by considering the train types after weight in descending order. For each train type as many trains as possible are added. This will yield an upper bound on how many trains of a single train type can be added. This solution is subsequently improved, if possible, by adding as many trains as possible of the remaining types by weight. In a network such an improvement will generally be possible due to certain train types having little or any interaction with one another.

Algorithm 1: Pseudocode for greedy heuristic.

Data: Weights, $w_i \in X$
Result: $X^* = (x_1, ..., x_i, ..., x_n)$ a solution that maximize $\sum_{i=1}^{n}(w_i \cdot x_i)$
$x_i^* = 0 \quad \forall i \in \{1, ..n\}$
for $i \in X$ *in descending order of* w_i **do**
  $x_i = 0 \quad \forall i \in \{1, ..n\}$ `// Reset solution`
  $x_i = dSearch(i)$ `// Find maximum amount of trains to add`
  for $j \in X$ *in descending order by* w **do**
    if $j \neq i$ **then**
      $x_j = dSearch(j)$
    end
  end
  $X^* = max(obj\_val(X), obj\_val(X^*))$ `// Set best solution`
end
return X^*

To determine how many trains that can be added of single train type, given a start solution, we use a dichotomic search algorithm which is an extension of the well-known binary search algorithm. The search is a divide-and-conquer algorithm that works by dividing the search interval into two parts at each iteration. For our problem the search works by initially determining a upper bound on the search interval. That is, a bound on the number of trains of a train type that yields an infeasible solution. Given this bound we know that the best feasible solution is in the interval between 0 and this upper bound. A good upper bound is close to the best solution. A bad upper bound will result in extra iterations which are computationally expensive due to the simulation model used to evaluate the capacity consumed. To test if it is even possible to add a train of the given type, the upper interval bound is initially set to 1. If one train yields an feasible solution, this is used as the lower interval bound, and the upper interval bound is increased by a guess. The guess is an estimate of how many trains that can be added of a single type. A good guess will provide a good upper interval search bound. Given that it is the same train type that is added, a guess could be based on the headway divided by the time period. This will result in the number of trains of the type in a total homogeneous situation.

When the upper interval bound has been found the search for the maximum number of trains is started. For each iteration the number of trains of the train type to be tested for feasibility is the midpoint between the lower interval bound and upper interval bound. If the tested number of trains yields an infeasible solution, the upper interval bound is set to the tested number of trains. If an feasible solution is found the lower interval bound is set to the tested number of trains. The algorithm is stopped after convergence and the best (maximum number of trains) is returned. This dichotomic search algorithm is also described in pseudocode in algorithm 2.

Algorithm 2: *dSearch*(i) Pseudocode for dichotomic search.

```
Data: i ∈ X: train type for which as many trains as possible should be
      added
X = (x_1, ..., x_i, ..., x_n), Y = (y_1, ..., y_i, ..., y_n): current solution
p: percentile
C_max: capacity consumption threshold
Result: x_i^* maximum number of trains that can be added
x_i^* ← 0
guess ← guess on upper interval bound
intervalLowerBound ← 0
intervalUpperBound ← 1
/* Find a upper bound on how many trains of type i that can
   be added                                                */
loop ← true
while loop do
    x_i ← intervalUpperBound
    // If infeasible
    if C(X + Y, p) > C_max then
        loop ← false // Upper bound for interval found
    else
        // If feasible
        intervalUpperBound = intervalUpperBound + guess
        if x_i > x_i^* then
            x_i^* ← x_i // Save as current best
        end
    end
end
/* Use binary search to find the maximum number of trains
   that can be added of type i                             */
x_i ← binarySearch(intervalLowerBound, intervalUpperBound)
return x_i^*
```

The greedy algorithm used finds the optimal solution if the solution space is concave and in certain cases convex. However, as the solution space might be neither convex or concave, the optimal solution can be a point in the solution space that is not a corner point. To investigate how the solution space looks, two simple cases are considered where all feasible solutions are constructed. The two simple cases, we consider is a line (only one link) which is traversed by two heterogeneous train types and an extended line traversed by three train types. In the extended line which is composed of three links in sequence, one train type runs on all three links and the two other runs only on a single, but different, link. The line case is the simplest possible. The extended line case is an extension which is constructed to investigate the solution space when there are train types that does not have conflicts.

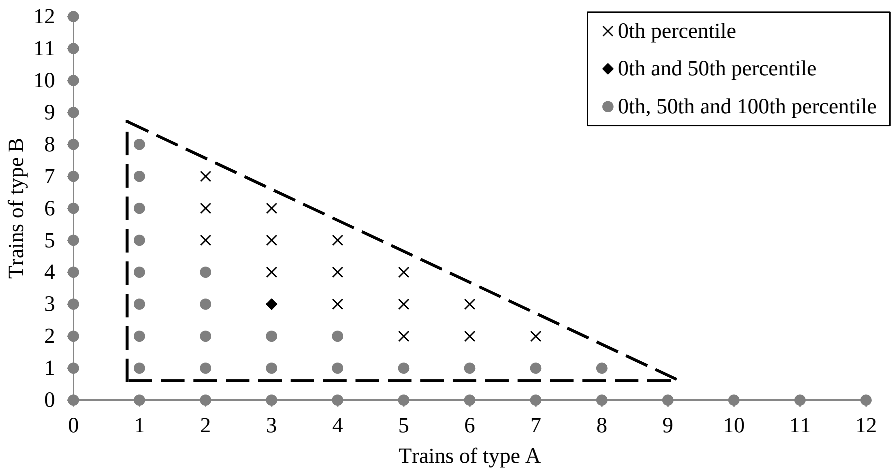

Fig. 3 Solution space on a railway line with two heterogeneous train types. 0th, 50th and 100th percentile depicted. The dashed box depicts an arbitrary restricted solution space.

The simple line case is shown in figure 3. As observed in the figure the solution space is concave. This means that only the corner points compose the convex hull of the solutions given a linear objective function. Thus, the optimal solution will be either (12,0) or (0,12) depending on the weights defined and if no mix solution has been found in step 1. If a mix solution has been found in step 1, the solution space will be restricted (no train types with 0 trains) and the extreme points are therefore cut off (shown as a dashed box in figure 3). However, the convex hull will still be composed by the corner points in the restricted solution space (due to the concave property).

The extended line case is shown in figure 4. As it can be observed in the figure the solution space is concave in two of three planes, just as for the line case. In the last plane the solution space is convex, which means that it is possible to add extra trains to some of the solutions in this plane with out removing trains of the other types. This is caused be the fact that two of the train types only run on one link which is not the same. It is therefore possible to saturate some of the solutions with extra trains. However, it should be noted that while this does not increase the deterministic capacity consumption, it will most likely increase the stochastic capacity consumption as buffer times are removed.

While figure 3 and 4 show solutions spaces which has desirable properties in form of a concave or rectangular (convex) solution space, then figure 5 show a solution space that is neither convex or concave. This solution space stems from the extended line case for the 25th percentile instead of the 50th percentile (median) depicted in figure 4. Thus in some cases the solution space will be neither convex or concave and an optimal solution is therefore not necessarily composed by a corner point. Therefore, we can conclude that the

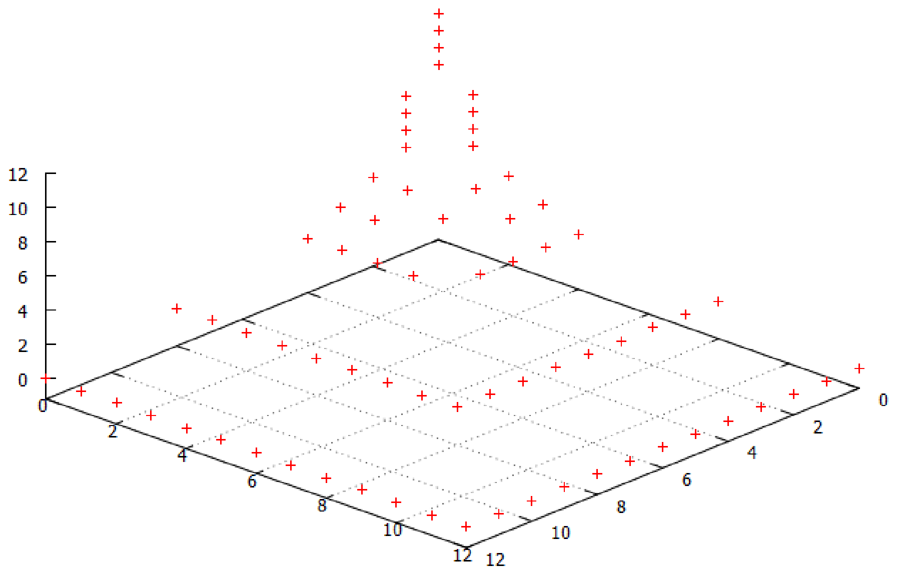

Fig. 4 Solution space on an extended line with three heterogeneous train types. 50th percentile depicted.

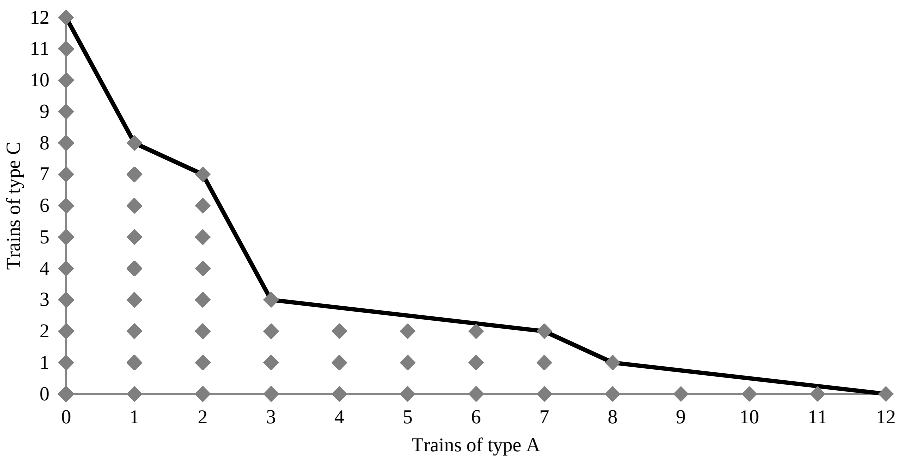

Fig. 5 Solution space which is neither concave or convex. 25th percentile for A,B-combination in the extended line case.

solutions found be our greedy heuristic are not necessarily optimal and thus optimality of the generated solutions cannot be guaranteed.

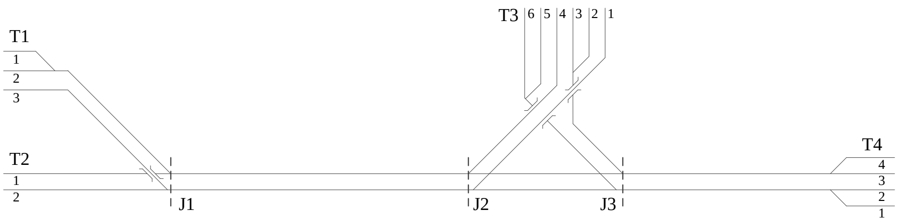

Fig. 6 Case network.

4 Case study

In this section, we apply the proposed optimisation framework to a network consisting of 161 kilometres of double track, at-grade and out-of-grade junctions and four terminal stations. In one hour, the goal is to estimate the number of trains the network can handle under a given mix plus additional trains maximizing utility (sum of weights). The network is depicted schematically in figure 6 with tracks used in normal operation.

Table 1 shows an overview of the train types that should traverse the network and their route. The train types are heterogeneous in running time, especially between T4 and J3 where the slowest train type (RE-A) is 17 minutes slower than the express train type that uses 31 minutes between T4 and J3. For the case a minimum (block) headway time of 150 seconds is used.

Table 1 Train types in the network with routes used.

Train type	**Route**
1: Express train	1: T4(4) $\rightarrow$ T3(2)
2: $\hookrightarrow$	2: T3(6) $\rightarrow$ T4(2)
3: IC-A	1: T4(4) $\rightarrow$ T3(2)
4: $\hookrightarrow$	2: T3(6) $\rightarrow$ T4(2)
5: IC-B	3: T4(3) $\rightarrow$ T1(1)
6: $\hookrightarrow$	4: T1(3) $\rightarrow$ T4(1)
7: RE-A	5: T4(4) $\rightarrow$ T3(3)
8: $\hookrightarrow$	6: T3(5) $\rightarrow$ T4(1)
9: RE-B	7: T3(4) $\rightarrow$ T1(2)
10: $\hookrightarrow$	8: T1(3) $\rightarrow$ T3(1)
11: Freight	9: T4(3) $\rightarrow$ T2(2)
12: $\hookrightarrow$	10: T2(1) $\rightarrow$ T4(2)

This case study mainly serves as a basis for theoretical experiments to explore how the optimisation framework performs. In section 4.1, we test different sample size to investigate the impact on the resulting capacity figures. Furthermore, we report some computational statistics. In section 4.2 we use the mix and weights listed in table 1 to test different capacity thresholds and the appertaining capacity results. The optimisation is executed on a Windows 7 desktop PC with an Intel Core i7-2600 (3.4 GHz) CPU with 8 GB of RAM.

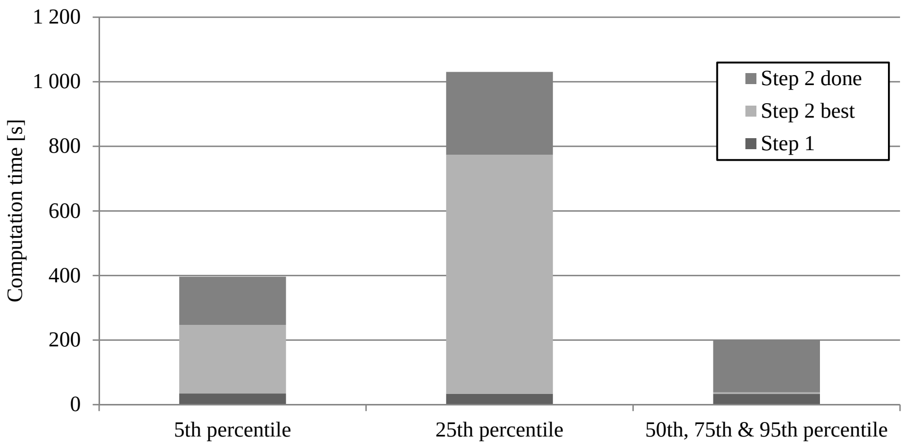

Fig. 7 Calculation time for each step in the algorithm.

4.1 Sample size and computational results

To investigate the impact of sample size on capacity results, we have tested three different sample sizes for the capacity consumption model. The mix and weights used are equal, thus no train type is given more importance than the others. 1, 5 and 10 million samples are tested using 75% (UIC recommendation (UIC, 2013)) as the capacity threshold, C_{max}, for the 5th, 25th, 50th, 75th and 95th percentile. As described earlier using the 0th and 100th percentile is not advisable due to variance in the capacity consumption model. Furthermore, the different percentiles represent how many percent of all permutations of train sequences that can be scheduled within the capacity threshold.

The test runs show that the three different sample sizes yield the same results for all percentiles. However, for the 25th percentile convergence were not obtained for 5 and 10 million samples as the 1 hour calculation threshold was exceeded, these results are therefore discarded. Given that the results are the same for the three sample sizes, it is concluded that using more than 1 million samples is not necessary for the case considered.

For the five percentiles calculated two unique results are found. For the 5th and 25th percentile the given mix can be scheduled once, that is one of each train type. In addition to this 2 extra trains of type 3, 1 train of type 4 and 11 trains of type 9 can be scheduled within the threshold with an objective value of 1.27. For the 50th, 75th and 95th percentile, the mix cannot be scheduled. However, 18 trains of type 3 and 18 trains of type 4 can be scheduled with an objective value of 3 within the capacity consumption threshold of 75%.

Figure 7 shows the amount of time used in each step of the algorithm for the five percentile tested with a 1 million sample size. In the figure, it can be observed that the calculation is completed significantly faster for the 50th, 75th and 95th, than for the 5th and especially the 25th percentile. The first step is completed equally fast for all the five percentile, however the best so-

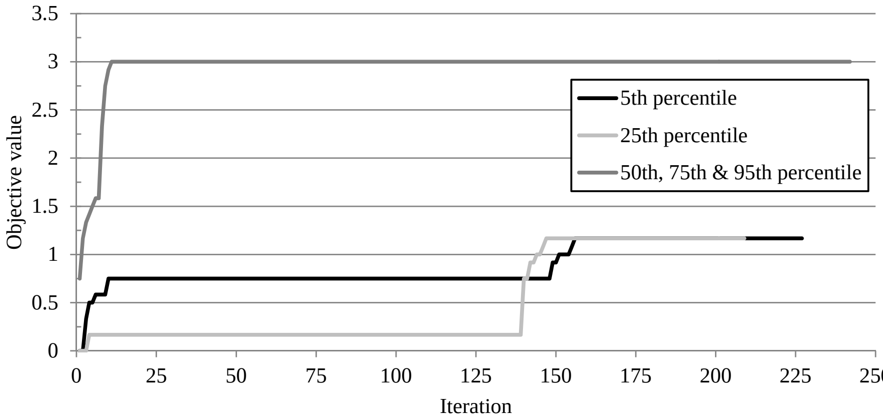

Fig. 8 Best objective value during search in step 2.

lution in step 2 is found much faster for the 50th, 75th and 95th percentile which is also illustrated in figure 8. This is caused by the fact that for the 5th and 25th percentile the mix could be scheduled. This leads to a heterogeneous solution compared to the 50th, 75th and 95th percentile, where only two train types compose the solution. Due to this heterogeneity, the capacity consumption model has considerably longer running times for the 5th and 25th percentiles, which then propagate to the running times of step 2 in the optimisation framework. For all the percentiles the calculation terminates after 211-245 iterations, as also seen in figure 8, thus this is not the reason for the longer running times for the 5th and 25th percentile.

The cause of the longer calculation time for the 25th percentile compared to the 5th percentile is not clear. However, in figure 8 it can be observed that the solution is only improved significantly after approximately 140 iterations. For the 5th, 50th, 75th and 95th percentile this happens already within 15 iterations. Thus the greedy heuristic used in step 2 evaluates a different set of solutions for the 25th percentile than 5th percentile. This leads to longer running times for the 25th percentile as the solutions examined takes longer to time to evaluate.

4.2 Mix and capacity threshold

From the results in the previous section, we conclude that a sample size of 1 million is sufficient for the case we consider. Using this sample size, we investigate the capacity results for different capacity consumption thresholds for the 5th, 25th, 50th, 75th and 95th percentile. The thresholds examined are 75%, 90% and 100%. The 75% is the UIC recommendation for lines with mixed traffic in the rush hour (UIC, 2013). 100% is the absolute maximum and can only be utilised under perfect conditions, that is no delays can occur. The 75% by UIC is given for line sections and is not a suitable threshold for routes and networks as the capacity consumption is higher when the whole network is considered instead of only a line section (Jensen et al, 2015). Therefore a threshold of 90% is also examined, which is considered the most realistic threshold of the three. Furthermore, we do not use an equal mix and equal weights for the train types, but differentiate them as listed in table 2.

Table 2 Train types in the network with route used, mix percentage for step 1 and weights for step 2 optimisation.

Train type	**Mix**	**Weight**
1: Express train	1/16	3/26
2: ↪	1/16	3/26
3: IC-A	1/16	2/26
4: ↪	1/16	2/26
5: IC-B	1/16	2/26
6: ↪	1/16	2/26
7: RE-A	1/16	1/26
8: ↪	1/16	1/26
9: RE-B	2/16	1/26
10: ↪	2/16	1/26
11: Freight	2/16	4/26
12: ↪	2/16	4/26

Table 3 Results with three different capacity thresholds for five different percentiles. Numbers in parentheses is extra trains added in step 2.

C_{max}	75%	90%	90%	90%	100%	100%	100%
Percentiles	5-95th	50-95th	25th	5th	75-95th	50th	5-25th
Mix scheduled			•	•		•	•
Train type	Number of trains (extra trains added in step 2)						
1	0	0	1	2 (1)	0	1	2 (1)
2	0	0	1	1	0	1	1
3	0	0	1	1	0	1	1
4	0	0	1	1	0	1	1
5	0	0	1	1	0	1	1
6	0	0	1	1	0	1	1
7	0	0	1	1	0	1	1
8	0	0	1	1	0	1	1
9	0	0	16 (14)	2	0	18 (16)	2
10	0	0	2	2	0	2	2
11	16 (16)	19 (19)	2	9 (7)	22 (22)	2	11 (9)
12	18 (18)	21 (21)	2	5 (3)	24 (24)	2	7 (5)
Mix - objective	0 - 5.2	0 - 6.2	1 - 0.5	1 - 1.7	0 - 7.1	1 - 0.6	1 - 2.3

In table 3, the results of the capacity calculation is shown. Within a capacity threshold of 75%, it is not possible to schedule a mix and 16 trains of type 11 and 18 trains of type 12 is therefore scheduled yielding an objective value of 5.2. These are the freight trains that have the highest weight of all the types. As the mix cannot be scheduled the final solution with the freight trains is very homogeneous and a total of 34 trains can therefore be scheduled. For the 90% capacity consumption threshold it is possible to schedule the mix for the 5th and 25th percentile, but not the 50th, 75th and 95th percentile. For the 100% threshold it also possible to schedule the mix for the 50th percentile. In addition to this it is possible to add extra trains in all solutions (usually freight trains as they have the highest weight) as listed in table 2.

Contrary to what could be expected, the total capacity increases for the 90% and 100% threshold with percentile. However, this is only when considering the number of trains. Results are as expected, when capacity figures are written as the number of times a train mix can be scheduled plus additional trains that can be added given a certain objective value.

5 Conclusions

In this paper, we have proposed an optimisation framework for the estimation of capacity in a railway network. The approach is able to determine the number of times a certain mix of train can be scheduled within a capacity consumption threshold, C_{max}. Additionally, it is estimated how many trains that can be added to this mix solution according to predefined train type weights until the capacity threshold, C_{max}, is met. Differentiated weights makes it possible to prioritise certain types as there may be higher demand for these.

In railway networks with heterogeneous operation the capacity consumed depends on the sequence (order) of trains. As this is unknown in early planning phases, our framework is able to give a span of capacity based on the relative amount of train sequences that should be feasible.

For a Danish long distance network of 161 kilometres of double track lines, the capacity can be estimated in 2-3 minutes in most of the test instances used. However, it may take up to 18 minutes in the worst case. Thus for strategic planning the model runs quite fast and should therefore be able to handle larger test instances than used for this paper.

In the second step of the framework, where additional trains are added, as many trains as possible is added of one single type. This is a natural consequence of the problem formulation and the heuristic used in this second step. This will usually not be desirable as solutions might contain a very high proportion of one train type. For further work, we therefore suggest to work with an alternative formulation of the mathematical model to ensure that not only one train type is added in the second step. And furthermore, to ensure directional symmetry between two train types that are "the same" just reversed.

For the second step of the framework a simple greedy heuristic is used to find the best solutions. However, this greedy heuristic is not the best approach to find the best solutions as the solution space is not necessarily concave or convex. For future work, we therefore suggest to test metaheuristics to improve the solutions found by the greedy heuristic. Although, it may increase calculation times.

Lastly, we propose to use the stochastic version instead of deterministic version of the capacity consumption model which forms the core part of the framework. This will yield more realistic capacity results that includes the aspect of robustness much better. However, this requires improvements in calculation time of the stochastic version of the model, which we are currently working on.

Acknowledgements The research described in this paper is part of the research project RobustRailS funded by the Danish Council for Strategic Research (now part of Innovation Fund Denmark). The authors would like to thank the council for making it possible to conduct this research.

References

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

Burdett R, Kozan E (2006) Techniques for absolute capacity determination in railways. Transportation Research Part B: Methodological 40(8):616 – 632, DOI 10.1016/j.trb.2005.09.004

Jensen L, Landex A, Nielsen O (2015) Assessment of stochastic capacity consumption in railway networks. In: Proceedings of The 6th International Seminar on Railway Operations Modelling and Analysis (RailTokyo2015)

de Kort A, Heidergott B, Ayhan H (2003) A probabilistic (max, +) approach for determining railway infrastructure capacity. European Journal of Operational Research 148(3):644 – 661, DOI http://dx.doi.org/10.1016/S0377-2217(02)00467-8

Mussone L, Wolfler Calvo R (2013) An analytical approach to calculate the capacity of a railway system. European Journal of Operational Research 228(1):11–23, DOI 10.1016/j.ejor.2012.12.027

UIC (2004) Capacity (UIC Code 406), 1st edn, pp 1–21

UIC (2013) Capacity (UIC Code 406), 2nd edn, pp 1–51