Rail Capacity Modelling With Constraint Programming

Daniel Harabor[1,2] and Peter J. Stuckey[1,2]

[1] National ICT Australia, Victoria Research Lab
[2] University of Melbourne, Department of Computing and Information Systems
firstname dot lastname @ nicta.com.au

Abstract. We describe a constraint programming approach to establish the coal carrying capacity of a large (2,670km) rail network in north-eastern Australia. Computing the capacity of such a network is necessary to inform infrastructure planning and investment decisions but creating a useful model of rail operations is challenging. Analytic approaches exist but they are not very accurate. Simulation methods are common but also complex and brittle. We present an alternative where rail capacity is computed using a constraint-based optimisation model. Developed entirely in MiniZinc, our model not only captures all dynamics of interest but is also easily extended to explore a wide range of possible operational and infrastructural changes. We give results from a number of such case studies and compare against an industry-standard analytic approach.

1 Introduction

Mining is one of the most important industries in Australia, and other parts of the world, and making mining supply chains efficient requires careful investment in the infrastructure that makes up the supply chain. The Bowen Basin in Central Queensland is home to 59 individual open-cut and underground mines. The large majority of all material is export coal with over 207 million tonnes having been produced in 2014. Once extracted, coal is railed from one of 37 different loadout points to one of 3 nearby coal ports. The set of all rail infrastructure serving the Bowen Basin is known as the Central Queensland Coal Network (CQCN).

Capacity planning in the context of the CQCN is an important and challenging topic. Investment decisions for infrastructure are typically highly expensive and have an effect over many years. In order to make the right decisions we need to model a range of competing alternatives and estimate in each case the maximum capacity (or throughput) of the rail network, typically measured in millions of tonnes of coal per annum (Mtpa). Key parameters that must be carefully considered include: the type of rolling stock, availability and performance of mines and ports, the number of lines in the network, the number and location of junctions and passing loops and operational constraints such as refuelling, crew changeover and temporal separation between trains. Figure 1 gives a small artificial example of an export coal supply chain. There are two typical approaches used to establish the capacity of rail in such a context:

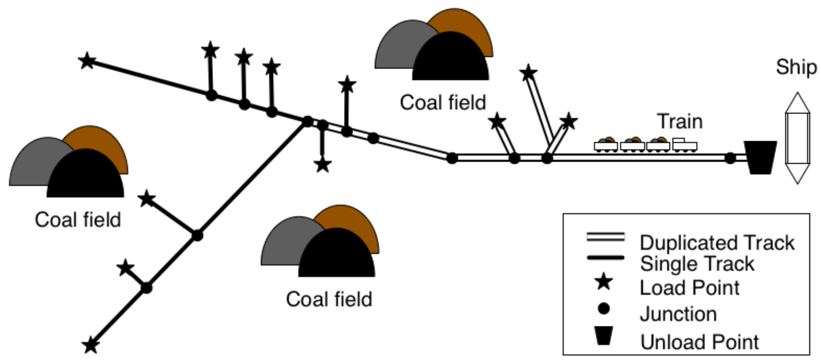

Fig. 1. The export coal supply chain. Raw material is extracted from large open-cut and underground mines. Once crushed and sorted, the coal is loaded onto trains and carried to unload points at a waterfront terminal. There the material is blended into various products and loaded onto ships for export. The rail component of such a supply chain comprises the load and unload equipment, rolling stock (locomotives and wagons), the physical rail network (lines and signals) and a set of operational parameters, in the form of rules, that govern how the infrastructure can be used in practice.

Analytic models This approach estimates the *theoretical capacity* of a rail system by creating simple mathematical models of operations that aim to saturate available infrastructure. A common approach is to consider capacity of a single line under e.g. fixed values for headway and travel time [3]; periodic traffic patterns [2] or; a set of fixed variables that represent mixed traffic and dwell times [12]. The primary advantage of these approaches is simplicity. The chief disadvantage is accuracy.

Simulation models Simulation methods can be used to model the physical infrastructure and the many operational requirements and constraints that arise in practice. An overview of such methods is given in [9]. In particular, tools such as OpenTrack [18] are intended to be very accurate but their primary strength is checking proposed schedules for feasibility; not deciding them in the first place. In cases where simulation models are extended to include a scheduling component, e.g. MultiRail [17], the typical approach is to add greedy algorithms to the simulation. The primary advantage of this approach is that many infrastructural and operational variables can be modeled together. The chief disadvantage is the time required to build the simulation and the quality of the decisions made within it.

In this paper we advocate a third approach, much less frequently used: building a CP-based *optimisation model* of the infrastructure system. While early examples of such works do exist (e.g. [13, 16]), they are typically limited to small single-track networks with few junctions and trains. Alternatively one could consider a mixed integer programming (MIP) based optimisation model, and there are a number of such approaches e.g.[1, 10]. These approaches are usually quite coarse grained, constraining capacities and using flow-based models, rather than actually building a scheduling model since time discretization is not feasible.

This accords with experience in other minimum make span scheduling problems where CP is usually superior to MIP. We observe that recent years have seen massive increases in computing power as well as as significant algorithmic gains in solving complex optimisation problems. Moreover, modelling and model transformation technology has also improved and the time required to create an optimisation model with modern constraint programming languages is much reduced [15]. To wit, we suggest that the time is ripe for switching to CP-based optimisation modelling for infrastructure planning.

To support this position, and at the request of a financial-industry partner with an interest in Queensland coal, we have created a scheduling-based constraint programming model of the CQCN. The model is written entirely in MiniZinc and offers many advantages: (i) the model describes all key infrastructural parameters of interest; (ii) the model considers decisions that actually reflect the best usage of the infrastructure; (iii) the model requires substantially less effort to produce than an equivalent simulation; (iv) the model makes it very easy to consider many "what if" situations. Indeed in many cases setting up such scenarios can be achieved by only changing input data.

We give a full description of the system and evaluate its performance in a range of freight-task scenarios. We also compare our model against a standard analytic approach to establishing rail capacity. Finally we apply the model to a number of "what if" infrastructural scenarios in order to demonstrate the flexibility of this approach and the benefits it can offer to industry planners.

2 The Central Queensland Coal Network

The Central Queensland Coal Network (CQCN) spans 2,670km of rail track and is the primary means of transporting export coal volumes; from 37 regional loadout points in Queensland's Bowen Basin to the nearby ports of Gladstone, Hay Point and Abbott's Point. Owned and operated by Aurizon Pty Ltd, the CQCN can be naturally divided into four separate but centrally managed and connected rail systems. These are known as Blackwater, Goonyella, Moura and Newlands. Each system imposes different constraints on train operations and each is configured to feed coal volumes to a specific port. Table 1 gives an overview of the four rail systems in terms of some key parameters. This data is sourced from a range of publicly available system descriptions [5–8, 4, 3].

When attempting to establish the coal-carrying capacity of a network such as the CQCN industry planners first create an idealised model of rail operations. This model is used in two ways: (i) to compute a maximum throughput figure for the as-is network and; (ii) to explore a range of what-if scenarios where infrastructure is added or modified or in which different operational practices are employed. The main difficulty facing industry planners is the large number of variables that need to be modeled and accounted for. For example there are 49 separate load and unload points in the CQCN and more than 130 junctions where

[3] We use as reference infrastructure equipment supplied by Techniplan to loadout points in the Goonyella system (at Carborough Downs and Isaac Plains).

	Blackwater	Goonyella	Moura	Newlands
Track Length	1108km	978km	261km	320km
Track Type	Single + Duplic'd	Single + Duplic'd	Single	Single
# Junctions	60	41	17	18
Travel Speed	80km/h	80km/h	80km/h	80km/h
Headway Time	20min	15min	90min	36min
Shunt Speed	10km/h	10km/h	10km/h	10km/h
Train Payload (Max)	10.6Kt	13.14Kt	10.6Kt	8.7Kt
Wagon Type	Hopper	Hopper	Hopper	Hopper
Wagon Capacity	106t	106t	106t	106t
Wagon Length	16.7m	16.7m	16.7m	16.7m
Load Points	10	20	4	3
Load Rate (Avg. max)[3]	4Kt/hr	4Kt/hr	4Kt/hr	4Kt/hr
Unload Points	4 (Shared)	5	4 (Shared)	2
Unload Rate (Avg. max)	5Kt/h	5.5Kt/hr	5Kt/hr	5Kt/hr

Table 1. Key infrastructural parameters for the CQCN. Applicable units are Kt (kilo-tonnes) and Kt/h (kilo-tonnes per hour). NB: When reporting number of junctions, we count only intermediate locations (not endpoints) that appear on a mine-to-port path.

trains can be scheduled to operate. In addition there are various operational requirements and constraints that can affect the efficacy of even idealised train services. These include: signalling, shunting, single track, crewing, refuelling, maintenance, and unexpected downtime.

3 Rail Capacity With Analytic Models

A common approach for analytically computing rail capacity is to combine a set of fixed operational parameters (train length, train payload, *headway* and *service time*[4]) together with simple models of relevant infrastructure. We create three such models to respectively characterise the maximum theoretical capacity of a single-track railway line, a mine loadout point and a port unload point:

$$A_{Line} = \frac{Total\ Time}{Headway\ Time} \times Train\ Payload \tag{1}$$

$$A_{Mine} = \frac{Total\ Time}{Load\ Time + Shunt\ Time} \times Train\ Payload \tag{2}$$

$$A_{Port} = \frac{Total\ Time}{Unload\ Time + Shunt\ Time} \times Train\ Payload \tag{3}$$

[4] In industry terminology, headway refers to the minimum temporal separation between two trains traveling in the same direction on the same rail line. Meanwhile, service time is the time necessary to fully load or unload a train, including shunting.

Parameters such as load, unload and shunt time are dependent on the exact characteristics of the train at hand and on the throughput capacity of load and unload points. Each of these can be varied to develop different scenarios. Where multiple parallel resources exist (e.g. duplicated rail lines or multiple loaders/unloaders) the models can likewise be extended appropriately. Every such analysis is obviously limited. For example the model A_{Line} assumes all trains are identical and always travel in the same direction. Meanwhile A_{Port} and A_{Mine} ignore the rail line altogether. Despite these drawbacks such methods are nevertheless attractive for their simplicity. Moreover, by computing analytic capacity from several different perspectives useful insights can often be attained. For example a very similar analytic approach to the one described here is currently used by Aurizon to "support pre-concept and concept studies" (in the CQCN) [3].

4 Rail Capacity With Optimisation Modelling

In order to establish rail capacity we will build a schedule of train trips to and from each mine. Since we are only creating a strategic model we will omit consideration of many operational matters (e.g. fleet-size and mix, crew pairing and rostering, variable travel times and any type of delay). We also do not model some existing dwell times; e.g. to facilitate refuelling and crew changeover, though these can be easily added. As such our results can be interpreted as assuming all trains are electric and autonomous.

Our model depends on two key parameters. The first of these, *loads per mine*, reflects the fact that we schedule the same number of round-trips from every mine site. It implicitly assumes that coal production is not a limiting factor any mine site.[5] The second parameter, *trains per mine*, reflects the fact that we assign a fixed number of dedicated trains to carry loads from each mine. This is not realistic (in practice the amount of rolling stock is usually limited) but appears quite reasonable for the purposes of infrastructural capacity estimation.

Next, rather than describe the entire rail network (which can be quite large), we simply model track segments between key junctions. These junctions are (i) load and unload points; (ii) rail yards where trains can be staged before/after servicing; (iii) junctions at the intersection of two or more branch lines; (iv) certain (hand chosen) passing loops which allow trains to share a single-track line. We also exploit the fact that in the CQCN (as in many rail networks) there is usually a single fixed path between each mine and the port. Every such path is computed a priori and made available as an input parameter to the model.

Notice that the underlying problem we solve is just train scheduling. Our model supports a variety of constraints relevant to this context including minimum headway time, single-track constraints and optional waiting at selected junctions (including time allowances for stopping and starting).

[5] With more data the model could be made more accurate in this regard.

4.1 MiniZinc

We now present a slightly simplified (for ease of exposition) version of our capacity planning model, written in MiniZinc [15]. The most important data are:

- a set of mines, `MINE`, where cargo originates.
- a set of junctions, `JUNC`, that split the rail network.
- the number of loads or round trips, `LPM`, to schedule from each mine.
- the number of trains available for each mine, `TPM`.
- a path, `path`, from each mine to the port, represented as list of at most `maxleg` junctions, using a `dummy` junction when we need less than `maxleg`.
- a set of locations, `LOC` $\supset$ `MINE` of things of interest.
- a mapping from junctions to locations, `junc_loc`.
- an expected travel time from location $l1$ to location $l2$, `travel_time`$[l1, l2]$.

We represent the trips between mines and ports using the array `TRIP`. Full trips, designated `FTRIP`, are assigned even indexes while empty trips, `ETRIP`, have odd. We now introduce the key decision variables and discuss associated constraints.

Decision Variables: The key decisions are at the level of each mine and trip:

- `mine_time`, decides when a train leaves (full) or arrives (empty) at each mine.
- `junction_time`, decides when a train (full or empty) should arrive at each junction and at the port. Note that most trips will not arrive at all junctions.
- `junction_wait`, decides how long a train (full or empty) waits at a junction.

We measure time in minutes, though wait times are discretised to be divisible by 5. Time granularity could easily be changed in the model if required. We additionally employ an array of convenience variables, `port_time`, each of which is associated with a corresponding variable from the `junction_time` array. These redundant variables simply collect the times each train arrives at the port (full) and leaves the port (empty). Their definition makes use of a parameter, `stop_allowance`, which is the number of minutes required to bring the train to a full stop, minus the usual time it would take to travel the distance of the stop. There exists a corresponding term, `start_allowance`, that is defined similarly and encountered later in the model. The decision variable declarations are:

```
set of int: LEG = 1..maxleg;
set of int: XJUNC = JUNC union { dummy };
array[MINE,LEG] of XJUNC: path; % path of junctions from mine to port
set of int: TRIP = 0..2*LPM-1;
set of int: FTRIP = { 2*i | i in 0..2*LPM-1};     % full trips
set of int: ETRIP = TRIP diff FTRIP;              % empty trips
array[MINE,TRIP]      of var TIME: mine_time;    % time leaving/arriving mine
array[JUNC,MINE,TRIP] of var TIME: junction_time; % time arriving at junction
array[JUNC,MINE,TRIP] of var WAIT: junction_wait; % wait time at junction
array[MINE,TRIP]      of var TIME: port_time =   % time arriving/leaving port
  array2d(MINE,TRIP, [ junction_time[port,m,t] +
                       stop_allowance*(t in FTRIP) | m in MINE, t in TRIP ]);
```

Mine loading constraints: We require each full trip to be loaded and to depart in order. The first train can leave after loading and the remaining trains follow. After `TPM` departures trains can return but only in the same order.

```
forall(m in MINE, t in FTRIP)
  (if t = 0 then                          % first train
     mine_time[m,t] >= load_time[m] + start_allowance
   elseif t div 2 < TPM then              % next few trains up to TPM
     mine_time[m,t] >= mine_time[m,t-2] + load_time[m] + start_allowance
                       + headway_time
   else
     mine_time[m,t] >= max(mine_time[m,t-2],mine_time[m,t-2*TPM+1])
                       + load_time[m] + start_allowance + headway_time
   endif);
```

Port unloading constraints: We require each empty trip to depart the port immediately after its full trip has unloaded, capturing the requirement that trains do not remain in the port after unloading. Note that our unload time includes a shunting component which is a function of the length of the train (this could also be modeled separately on a per-train basis).

```
forall(m in MINE, t in ETRIP)
      (port_time[m,t] = port_time[m,t-1] + unload_time + start_allowance);
```

Port capacity constraints: We ensure that no more trains are unloading at the port than there are dump stations, `unload_capacity`.

```
cumulative([port_time[m,t] | m in MINE, t in FTRIP],
           [unload_time | m in  MINE, t in FTRIP],
           [ 1 | m in MINE, t in FTRIP], unload_capacity);
```

Unused junctions: We record a time for each trip at each junction, since there are not that many junctions, but of course almost no trips will visit all junctions. The unused junctions are set to have time and wait of 0.

```
array[MINE] of set of JUNC: junctions_for_mine =
  [ {path[m,l]|l in LEG where path[m,l] != dummy} | m in MINE];
array[JUNC] of set of MINE: mines_for_junction =
  [ {m | m in MINE where j in junctions_for_mine[m] }| j in JUNC ];
forall(m in MINE, t in TRIP, j in JUNC diff junctions_for_mine[m])
  (junction_time[j,m,t] = 0 /\ junction_wait[j,m,t] = 0);
```

Travel time: leg-to and leg-from mine: We model (separately) the travel time for full trips, from the mine to the first junction on its path to the port. In a similar way we also model travel time for empty trips, from the last junction in the path to the mine. Note how full trips constrain the times between junctions in the opposite order to empty trips.

```
forall(m in MINE, t in FTRIP)
  ( let { JUNC: j = path[m,1]; LOC: l = junc_loc[j]; } in
    junction_time[j,m,t] >= mine_time[m,t] + travel_time[m,l] );
forall(m in MINE, t in ETRIP)
  ( let { JUNC: j = path[m,1]; LOC: l = junc_loc[j]; } in
    mine_time[m,t] >= junction_time[j,m,t] + junction_wait[j,m,t]
                      + stop_allowance + travel_time[l,m] );
```

Travel time: inter-junction legs: Travel time between adjacent junctions gives rise to a similar constraint.

```
forall(m in MINE, t in FTRIP)
  ( forall(s in 1..maxleg-1 where path[m,s+1] != dummy)
          ( junction_time[path[m,s+1],m,t] >= junction_time[path[m,s],m,t]
            + junction_wait[path[m,s],m,t]
            + travel_time[junc_loc[path[m,s]],junc_loc[path[m,s+1]]] ) );
forall(m in MINE, t in ETRIP)
  ( forall(s in 1..maxleg-1 where path[m,s+1] != dummy)
          ( junction_time[path[m,s],m,t] >= junction_time[path[m,s+1],m,t]
            + junction_wait[path[m,s+1],m,t]
            + travel_time[junc_loc[path[m,s+1]],junc_loc[path[m,s]]] ) );
```

Minimal wait times: A train needs to come to a complete stop to wait at a junction hence there is a minimal amount of time it is delayed by any wait.

```
forall(j in JUNC, m in MINE, t in FTRIP)
      ( junction_wait[j,m,t] = 0 \/
        junction_wait[j,m,t] >= stop_allowance + start_allowance );
```

Siding capacity at junctions: We constrain trains waiting at a junction j to be no more than the number of sidings at the junction, `sidings[j]`.

```
forall(j in JUNC)
  ( cumulative([junction_time[j,m,t] | m in MINE, t in TRIP],
               [junction_wait[j,m,t] | m in MINE, t in TRIP],
               [ 1 | m in MINE, t in TRIP], sidings[j]) );
```

Headway constraints at junctions: Rather than using a `disjunctive` constraint to model that no two trains pass a junction in the same direction within headway time, since all the "durations" of these tasks are the same we simply use `alldifferent`. This is slightly stronger constraint than the `disjunctive` constraint but accurate enough for capacity planning.

```
forall(j in JUNC)
      (alldifferent([ junction_time[j,m,t] div headway_time
                    | m in mines_for_junction[j], t in FTRIP]) /\
       alldifferent([ junction_time[j,m,t] div headway_time
                    | m in mines_for_junction[j], t in ETRIP]));
```

4.2 Single track constraints

When there is only a single track between two locations we must ensure no two trains try to use the track while traveling in opposite directions. Though there are complex ways of modelling this using variable set up times we adopt a simpler approach where each train reserves the track for the entire time it is using it. By varying the granularity of the model (adding new junctions) we can limit the inaccuracy that derives from this overly restrictive constraint. This approach requires us to introduce the notion of *track segments* into the model.

A track segment `s in SEG` has: a start junction, `start_junc`, which may be `dummy` if the segment is a leaf; an end junction, `end_junc`; an (optional) set of mines that sit on that segment (usually in unmodelled mine-specific balloon loops), `mines_on_segment`; and a set of mines that use the segment on their path to and from the port, `mines_using_segment`.

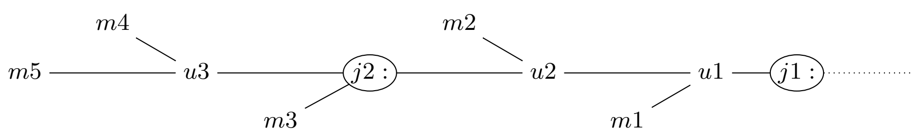

Fig. 2. Part of an (abstract) rail network.

Example 1. Consider the abstract rail network shown in Figure 2 which includes junctions $j1$ and $j2$, mines $m1, \ldots, m5$ and unmodelled intersections $u1$, $u2$ and $u3$. The rail network consists of 2 segments: a leaf segment ending at $j2$ which includes the mines $m3$, $m4$ and $m5$, and a non-leaf segment from $j2$ to $j1$ which includes the mines $m1$ and $m2$. There are no (additional) mines that use the first segment on their path to the port, while the mines $m3$, $m4$ and $m5$ all use the second segment on their path to the port. □

Leaf segments: Leaf segments connect mines to the rest of the network. We make sure that no train going to or from a mine in that segment overlap in time by using the travel time to/from the mine to the end junction of the segment.

```
array[SEG] of set of MINE: mines_on_segment;
forall(s in SEG where start_junc[s] = dummy)
  ({ let { JUNC: j = end_junc[s]; LOC: l = junc_loc[j]; } in
    disjunctive([ if t in FTRIP then               % start time
                    mine_time[m,t]
                  else junction_time[j,m,t] + junction_wait[j,m,t] endif
                | m in mines_on_segment[j], t in TRIP ],
                [ if t in FTRIP then               % duration
                    travel_time[m,l]
                  else travel_time[l,s] endif
                | m in mines_on_segment[s], t in TRIP ] ) );
```

Non-leaf segments: Non-leaf segments are used to handle trains traveling between the start and end junctions of the segment. They also handle trains that travel from either of these junctions to a mine that sits on the segment. Notice that this constraint always uses the start-to-end travel time. There is an implicit assumption here that this duration is always less than the travel time to (or from) a mine that sits on the segment. For our data sets this is always the case, but the model would need adjustment if it were not the case.

```
array[SEG] of set of MINE: mines_using_segment;
forall(s in SEG where start_junc[s] != dummy)
  ( let { JUNC: sj = start_junc[s]; LOC: sl = junc_loc[sj];
          JUNC: ej = end_junc[s]; LOC: el = junc_loc[ej];
          set of MINE: M = mines_on_segment[s] union
                           mines_using_segment[s]; } in
    disjunctive([ if t in FTRIP then               % start time
                     junction_time[sj,m,t] + junction_wait[sj,m,t]
                  else junction_time[ej,m,t] + junction_wait[ej,m,t] endif
                | m in M, t in TRIP ],
                [ if t in FTRIP then               % duration
                     travel_time[sl,el]
                  else travel_time[el,sl] endif
                | m in M, t in TRIP ] ) );
```

4.3 Search Strategy

We use the Gecode [14] solver to tackle our models. The default autonomous search does not perform well so we employ the following simple hybrid which does: we use a dom/wdeg variable selection heuristic [11] but order the variables carefully so that tie-breaking in dom/wdeg chooses the variables in a sensible order. We have found the following simple ordering to be particularly effective: (i) decision variables that determine arrival and departure times from mine load-points appear first; (ii) decision variables that determine arrival and departure times from port unload points appear next; (iii) all other decision variables follow, in any order. Given decision variables that are ordered in a "good" way, we have found that Gecode can often identify near-optimal solutions very quickly.

5 Experiments

We use our optimisation model to explore a range of infrastructural scenarios, many of which are difficult to evaluate analytically. These scenarios are:

- Capacity of the current infrastructure.
- Capacity under the assumption of increased payloads per train.
- Capacity assuming the addition of new below-rail infrastructure[6]; e.g. additional signalling and duplicated rail lines.

Where possible we will compare our computational approach against the industry-standard analytic techniques discussed in Section 3. Recall that these simplified models are used to compute the *maximum theoretical capacity* of infrastructure. We will compare against these optimistic upper-bounds in order to evaluate the quality of solutions computed with our CP model. Capacity figures are always given in Mtpa: Millions of tonnes (of coal) per annum.

5.1 Infrastructural Capacity With Analytic Modelling

Recall that the analytic model from Section 3 focuses on different aspects of the network to the exclusion of all other factors. To mitigate this myopic bias we will compute analytic capacity from three points of view: ports, mines and the physical rail lines. Table 2 presents our results. We assume loading, unloading and travel all proceed without delay and that infrastructure is always available and always operates at maximum throughput. When modeling trains we use a range of established operational parameters including real-world headway times and industry maximums for train length and payload size in each rail system. The full set of all such parameters are given in Table 1 while results from this analysis are given in Table 2. We make several observations:

[6] In industry terminology, *below-rail* refers to infrastructure controlled by the network owner, such as the physical track and signals. By comparison *above-rail* refers to infrastructure such as trains, wagons and other so-called rolling stock.

Network	Theoretical Capacity Model			Additional Parameters	
	A_{Mine}	A_{Port}	A_{Line}	Infrastructure Availability	Line Type
Blackwater	329.3	162.2	278.6	100%	Single Track
Moura	131.7		61.9	100%	Single Track
Goonyella	658.7	221.4	460.4	100%	Single Track
Newlands	98.7	81.0	127.02	100%	Single Track

Table 2. Analytic evaluation of the theoretical capacity of each rail system in the CQCN. Each of the three models take as input operational parameters from Table 1.

Network	Parameters		Network Performance				
	LPM	TPM	Trains	Avg. Cycle Time	Total Wait	Port Util.	Capacity
B/Moura	15	2	13,464	17.4 hrs	0	87.8%	142.7
Goonyella	15	2	15,130	19.8 hrs	0	89.2%	198.9
Newlands	35	4	8,860	10.0 hrs	0	95.5%	77.9*

Table 3. CP-based rail capacity. We assume current CQCN operational parameters, as described in Table 1. Columns LPM and TPM respectively indicate the number of loads per mine (i.e. the size of the freight task) and the number of (dedicated) trains per mine. Figures denoted with * are provably optimal.

– The data suggests that water-front unload points (and not the rail network) is the most likely bottleneck in each rail system.
– The port bottleneck observation holds despite our (pessimistic) assumption of single-line track for every A_{Line} model. Note that while this assumption is true for Moura and Newlands there exist large portions of Blackwater and Goonyella that are duplicated. We continue to use the single-line assumption in these cases as the the majority of mines are on spurs[7] that connect to the network via single-track branch lines.

5.2 Infrastructural Capacity With Optimisation

Next, we evaluate capacity in the CQCN using our scheduling-based optimisation model and the Gecode solver. As in the analytic case we employ the full range of real-world parameters from Table 1 and assume that infrastructure is always available and operates at maximum throughput. The first solution is typically found in seconds and we allow the solver to run for up to a minute thereafter.

We evaluate the capacity of each rail system by measuring its steady-state performance and extrapolating out to a full year. To avoid warm-up and cool-down effects we ignore loading and unloading operations at the beginning and toward the end of the schedule. In particular we consider only port arrivals between the first and third quartiles of our planning horizon. Results are given in Table 3. We make several observations:

[7] In industry terminology, a spur is a short branch usually leading to a private siding.

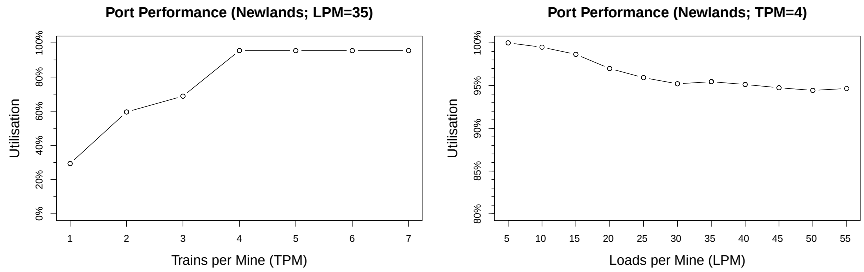

Fig. 3. Tuning LPM and TPM parameters for the Newlands System model. We isolate each parameter and vary its value. We measure the impact of each change by computing the percentage utilisation of port unloaders in each resultant scenario.

– In the case of the Newlands system we find that our optimisation approach is able to compute an exact figure for the maximum infrastructural capacity of rail. The figure (77.9 Mtpa) is within 5% of the optimistic upper-bound established by the analytic model A_{Port}.
– In the case of Blackwater/Moura and Goonyella we compute approximate capacities which are within 10.8% and 10.2% of the upper-bound A_{Port}.
– In all three cases port utilisation is very close to or above 90%. These figures suggest that the rail network is not the primary limiting factor for increased coal export volumes in the future. Rather, each system appears constrained by the infrastructural capacity of their respective ports.

For the experiments at hand the parameters LPM and TPM were hand-tuned on a per-model basis. If LPM is too small, the freight task can be finished quickly and before the system can reach a steady state. Alternatively, if LPM is too large the problem may grow to a size where our optimisation solver cannot compute a good solution in reasonable time. Similar observations are true for the parameter TPM. Given too few trains the port infrastructure can remain idle for long periods and its performance will not be indicative of potential capacity. On the other hand a TPM value that is too large can explode the search space, again making any solution difficult to find in a reasonable amount of time.

With LPM=15 and TPM=2 the size of the planning horizon is 7.3 days for Goonyella and 5.9 days fro Blackwater/Moura. We found these values sufficient to take reliable readings of network performance. In the case of the Newlands System the planning horizon with these parameters is too small to be useful (<3 days). Figure 3 gives results from a range experiments in which we empirically identified appropriate values for Newlands. Notice that: (i) setting TPM > 4 does not make any difference to port utilisation but smaller values have a large impact; (ii) setting LPM < 30 is insufficient to reach the system's steady-state.

5.3 Case Study: Increased Payloads

One of the case studies asked for by our industry partner is to determine rail capacity under the assumption that all trains have fixed payloads. The proposed

	10Kt Scenario		12Kt Scenario		14Kt Scenario		Current Max Scenario	
	Capacity	T. Len	Capacity	T. Len	Capacity	T. Len	Capacity	T. Len
B/Moura	139.6	1587m	**145.6**	1904m	138.4	2205m	142.7	1670m
Goonyella	197.5	1578m	198.6	1904m	197.8	2205m	**198.9**	2071m
Newlands	69.3	1578m	61.9	1904m	64.1	2205m	**77.9**	1369m

Table 4. Experiments using a range of alternative payload sizes. We measure capacity in three scenarios where all trains carry uniform payloads of 10, 12 and 14Kt (kilotonnes) of coal. For context, we also give results from the current capacity scenario which considers fully-loaded trains of the maximal size currently permitted in each rail system (see Table 1). Figures in bold indicate best results (highest capacity) found.

volumes are 10Kt, 12Kt and 14Kt. Increased payload scenarios involve modeling trains which are longer or which comprise wagons that are more densely packed. Lacking data regarding alternative wagon configurations we opt to model longer trains. Note that both options may require additional below-rail infrastructure; either in the form of longer balloon loops (to support longer trains) or new load and unload equipment (configured to support densely packed trains).

To model trains with alternative payload configurations we simply modify a single value in the associated data file for each network and run the solver anew. No change to the optimisation model is needed. A similar data-driven change would also be sufficient to model the densely-packed scenario (in this case we would need to modify wagon length and wagon capacity parameters in addition to payload size). All other parameters remain as in Section 5.2. Results from this experiment are given in Table 4. We observe that with few exceptions each increased/uniform payload scenario appears to make little difference to rail capacity beyond what can be achieved by running trains with the maximum currently permissible payload size. One exception is the Blackwater/Moura system where a small gain of 3Mtpa can be achieved by running 12Kt trains instead of the current maximum payload size of 10.6Kt.

5.4 Case Study: Decreased Headway

Another possibility for increasing the capacity of a rail system is to decrease the cycle time (i.e. round-trip time) per train. Such scenarios could involve deploying additional infrastructure or technology to allow decreased headway (i.e. a smaller temporal separation) between trains or the introduction of new rolling stock that can travel at faster speeds. We model the decreased headway scenario here though new rolling stock is equally simple to analyse. In both cases we make changes only to parameter values. The optimisation model remains unchanged. Results are given in Table 5.

In a range of experiments we observe that the total throughput of each rail system is largely invariant, even with reduced headway times. In the case of Blackwater/Moura system an increase of 3% (vs. the Current Capacity scenario) appears achievable if we fix the headway time of all trains to 26 minutes. This

Headway (mins)	Blackwater/Moura		Goonyella		Newlands	
	Capacity	Port Util	Capacity	Port Util	Capacity	Port Util
6	145.8	89.6%	195.9	87.9%	60.1	73.7%
16	144.7	89.0%	198.9	89.2%	64.8	79.4%
26	**147.6**	90.7%	**202.6**	90.9%	66.7	81.7%
30	142.2	87.4%	195.5	87.7%	**80.5**	98.6%
Current	142.7	87.8%	198.9	89.2%	77.9	95.5%

Table 5. Experiments using a range of fixed headway times. We evaluate their effectiveness in terms of capacity and port utilisation. For context, we compare these results against the capacity figures computed in Section 5.2 (row "Current"). Figures in bold indicate best results (highest capacity) found.

value is larger than the 20 minutes currently used for junctions in the Blackwater system but much smaller than the 90 minutes used in Moura. A similar gain can be achieved in Newlands when headways are reduced to 30 minutes (cf. 36 currently). It is interesting to note that for the Goonyella system the best result is for 26 minutes (cf. 16 currently). We interpret this as suggestive that small amounts of extra waiting can help when there is a high degree of contention for rail resources.

5.5 Case Study: Track Duplication

For a final case study we consider the impact on rail capacity through the duplication of key sections of rail track. Introducing new line capacity into the system reduces waiting and track contention and allows parallel travel in both directions (i.e. simultaneously to and from the port). There are two aspects to such an analysis: (i) we must identify which sections of track are most likely to yield the greatest benefit; (ii) we must evaluate the effect of the proposed simulation. We begin with an analysis of the Blackwater system.

Figure 4 shows the arrival frequency of trains at the most visited junctions in the Blackwater system. A junction is a reasonable candidate for duplication if the arrival frequency of trains traveling in the same direction is close to or less than the minimum headway time. We observe that while the busiest single-line junctions (Dingo, Walton, Umolo and Bluff) have trains arriving every 27-28 minutes, the frequency in any single direction is almost twice that at 50 minutes. As there is no contention we may thus infer that track duplication at these points will not increase the infrastructural throughput of the system. We confirmed this hypothesis empirically. Similar results hold for each of the other rail systems under consideration.

It is important to note that track duplication e.g. between Dingo and Bluff may still make sense operationally. With only 30 minutes of idle time between arrivals, and round-trip times of over 17 hours (see Table 3), it is entirely possible that unforeseen delays during loading, unloading or during travel on the network could result in contention for track resources at these locations.

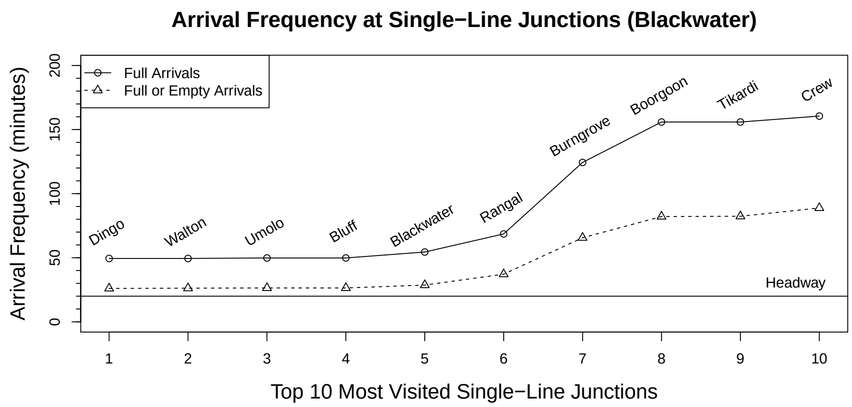

Fig. 4. Most visited single-line junctions in the Blackwater system. We give the average time difference between arrival times for full and empty trains at each junction. Measurements are in minutes and reflect system performance during its steady state.

6 Conclusion

We evaluate the infrastructural capacity of four rail systems which together comprise the Central Queensland Coal Network. Similar capacity evaluation problems appear in a range of industrial settings but especially cases where bulk goods and freight containers must be railed between inland terminals and the waterfront. Effective models that capture the dynamics of a such a system are prized tools of industry planners.

We propose a new approach for rail capacity estimation using constraint programming with MiniZinc. Written in the form of a scheduling problem, our model is simple to develop, easy to extend and can be used to compute fast and accurate capacity estimates for large rail networks. Because it is data-driven the model makes it especially easy to evaluate a wide range of "what-if" scenarios of interest to industry planners. We give particular examples involving alternative train payloads, alternative headway times and track duplication scenarios.

There are many other scenarios of practical interest such as mixed train lengths and grade easing. We could extend our model to investigate these. We can also extend our model to capture further dynamics of the system like: scheduled downtime, different train speeds, refuelling operations and crew changeover. Most of these extensions appear quite straightforward to achieve.

We believe the principal lesson of this paper is that optimisation technology has matured to the point where we can quickly undertake detailed infrastructure modelling and analysis. Such capability is essential to inform long-term infrastructural investment decisions made by governments and large corporations.

7 Acknowledgements

We thank Eric Nettleton for useful discussions during the development of this work. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

1. Abdekhodaee, A., Dunstall, S., Ernst, A.T., Lam, L.: Integration Of Stockyard And Rail Network: A Scheduling Case Study. In: Proceedings Of The Fifth Asia Pacific Industrial Engineering And Management Systems Conference, Gold Coast, Australia (2004)
2. Abril, M., Salido, M.A., Barber, F., Ingolotti, L., Lova, A., Tormos, P.: A Heuristic Technique for the Capacity Assessment of Periodic Trains. In: Proceedings of the 2005 Conference on Artificial Intelligence Research and Development. pp. 339–346. IOS Press, Amsterdam, The Netherlands, The Netherlands (2005)
3. Aurizon Network Pty Ltd: 2014 Network Development Plan (2014), http://www.aurizon.com.au/Downloads/Aurizon Network Development Plan 2014.pdf, last accessed 2015-09-29
4. Aurizon Network Pty Ltd: Review of rail infrastructure and line diagrams for central queensland coal region – 30 december 2014 (2014), http://www.aurizon.com.au/network/central-queensland-coal-network/goonyella-system, last accessed 2015-09-29
5. Aurizon Network Pty Ltd: Blackwater system information pack (issue 5.6) (2015), http://www.aurizon.com.au/network/central-queensland-coal-network/blackwater-system, last accessed 2015-09-29
6. Aurizon Network Pty Ltd: Goonyella system information pack (issue 6.4) (2015), http://www.aurizon.com.au/network/central-queensland-coal-network/goonyella-system, last accessed 2015-09-29
7. Aurizon Network Pty Ltd: Moura system information pack (issue 6.0) (2015), http://www.aurizon.com.au/network/central-queensland-coal-network/moura-system, last accessed 2015-09-29
8. Aurizon Network Pty Ltd: Newlands system information pack (issue 6.4) (2015), http://www.aurizon.com.au/network/central-queensland-coal-network/newlands-system, last accessed 2015-09-29
9. Barber, F., Abril, M., Salido, M., Ingolotti, L., Tormos, P., Lova, A.: Survey Of Automated Systems For Railway Management. Tech. Rep. Technical Report DSIC-II/01/07, Department of Computer Systems and Computation, Technical University of Valencia (2007)
10. Boland, N.L., Savelsbergh, M.W.: Optimizing the hunter valley coal chain. In: Gurnani, H., Mehrotra, A., Ray, S. (eds.) Supply Chain Disruptions, pp. 275–302. Springer London (2012)
11. Boussemart, F., Hemery, F., Lecoutre, C., Sais, L.: Boosting systematic search by weighting constraints. In: Procs. of ECAI04. pp. 146–150 (2004)
12. Burdett, R., Kozan, E.: Techniques for absolute capacity determination in railways. Transportation Research Part B: Methodological 40(8), 616 – 632 (2006)
13. Fukumori, K., Sano, H., Hasegawa, T., Sakai, T.: Fundamental algorithm for train scheduling based on artificial intelligence. Systems and Computers in Japan 18(3), 52–64 (1987)

14. Gecode: generic constraint development environment, www.gecode.org
15. Nethercote, N., Stuckey, P., Becket, R., Brand, S., Duck, G., Tack, G.: MiniZinc: Towards a Standard CP Modelling Language. In: Procs. of CP2007. LNCS, vol. 4741, pp. 529–543. Springer-Verlag (2007)
16. Oliveira, E., Smith, B.M.: A job-shop scheduling model for the single-track railway scheduling problem. RESEARCH REPORT SERIES-UNIVERSITY OF LEEDS SCHOOL OF COMPUTER STUDIES LU SCS RR (21) (2000)
17. OliverWyman: MultiRail Planning Suite. http://oliverwyman.com (2012), last accessed 2015-11-24
18. OpenTrack: Simulation of rail networks. http://www.opentrack.ch/, last accessed 2015-11-24