A Mixed Integer Programming Model For Long Term Capacity Expansion Planning: A Case Study From The Hunter Valley Coal Chain

Gaurav Singh, David Sier, Andreas T. Ernst, Olena Gavriliouk, Rob Oyston, Tracey Giles, Palitha Welgama

CSIRO Mathematics, Informatics and Statistics
Private Bag 33, Clayton South MDC, 3168, Victoria Australia
Email addresses: Firstname.Lastname@csiro.au
URL: www.csiro.au/org/CMIS.html

Hunter Valley Coal Chain Coordinator
PO Box 101, Carrington, NSW 2294

Rio Tinto,
Operations Centre, 11 George Wiencke Drive,
Perth Domestic Airport, Western Australia 6104
URL: http://www.riotintoironore.com

Abstract

The Hunter Valley Coal Chain is the largest coal export operation in the world with a throughput in excess of 100 Mtpa. Coal is delivered to the shipping terminal from 40 mines using 27 coal load points spread across the Hunter Valley region. This paper describes an MILP model for determining the capacity requirements, and the most cost effective capacity improvement initiatives, to meet demand while minimising the total cost of infrastructure and demurrage. We present results from computational experiments on the model’s performance along with a comparison of the model’s output with detailed analyses by the coal chain analysts and planners.

Key words: Facilities planning and design, Integer programming, Heuristics, Coal supply chain operations.

1. Introduction

The Hunter Valley Coal Chain is a joint venture between the mining, rail, terminal and port organisations that coordinate the export of coal from the area around the Australian town of Newcastle. Port Waratah Coal Services Limited (PWCS) operates the coal terminals at The Port of Newcastle, which is the world’s largest coal handling operation with total coal exports exceeding 100 Mtpa. The coal chain comprises 40 coal mines owned by 13 individual coal producers. There are 27 mine loading points served by approximately 40 trains making around 16,000 trips per year. The system exports more than 80 different blended coal brands through

*Corresponding author. Tel: +61 3 9545 8043; fax: +61 3 9545 8080.
five berths at three coal terminals which serve approximately 1,200 vessels, with an average size of 90,000 tonnes, per year. There is a total stockpile capacity of 3.4 million tonnes at the port with approximately 1.8 million tonnes of workable stockpile space for port operations.

The coal chain operations are coordinated by the Hunter Valley Coal Chain Coordinator (HVCCC) group whose primary goal is to maximise coal export volumes on a daily basis, and to coordinate planning for the provision of future coal chain infrastructure. The HVCCC has developed a simulation model of the coal chain that considers mines, load points, rail operation and port operations, and uses this simulation model to support capacity planning decisions. This paper describes an advanced optimisation tool commissioned by the HVCCC and designed to complement and/or integrate with the current simulation tool.

2. Background

Much of the supply chain literature focusses on Supply Chain Management (SCM), the integration, coordination and management of the different key business processes, often operating within individual silos, that deliver products, services and information from suppliers to customers across the chain. This area of the literature provides a background to the current paper, in the sense that it deals with factors such as setting up, operating, and improving systems like coal supply chains. Romano (2003) provides an overview of developments in supply chain management noting that it has evolved from encompassing mainly logistics activities (inventory management, transportation, warehousing, and order processing) to include other processes such as customer relationship management, product development and commercialisation, and quality management. Complementary reviews of SCM and supply chain design are given by Grossmann (2004), Power (2005), Meixell & Gargeya (2005), Petersen et al. (2005), Stadtler (2005), Trkman et al. (2010).

Various optimisation approaches are considered under the SCM umbrella. Melo et al. (2009) review facility location models in the context of supply chain management and in an earlier paper (Melo et al., 2005) propose a mathematical modelling framework for the strategic design of supply chain networks. Other authors who have considered the optimal design and configuration of deterministic and stochastic supply chains include Korpela et al. (2002), Gupta & Maranas (2003), Narasimhan & Mahapatra (2004), Guillena et al. (2005) and Santoso et al. (2005).

The focus in this paper is not the (optimal) design of a coal supply chain but rather a consideration of how we might maximise the throughput of products in such a supply chain by scheduling the use of specified sets of facilities and machines whose operations are constrained by capacities and operational rules relating to their use. In particular, we are interested in determining the optimal use of additional facilities and equipment proposed as part of a capacity expansion program for a supply chain, and to see whether the ‘new’ supply chain can provide the required increase in throughput. We expect that the application of SCM principles, or methodologies such as collaborative planning, forecasting and replenishment approaches to integrating supply chain members (Fliedner, 2003) would be used as part of the planning process to the expansion program.

With respect to supply chain scheduling, Hall & Potts (2003) noted that while there is an extensive supply chain literature, coordinated decision making in tactical supply chain scheduling models had not, at the time of their paper, been studied. They develop a job scheduling model for a tree structured supply chain in which a supplier makes deliveries to several customers who in turn make deliveries to customers, the benefit of cooperation between suppliers and customers is also considered. The objective is either to minimise the number of late jobs or to minimise the
maximum lateness. They use a polynomial time dynamic programming based algorithm to solve a number of tractable problems within a set of supply chain examples. In a later paper, Chen & Hall (2007) extend these scheduling models to consider the effects of conflicts between suppliers and a manufacturer arising from the requirement that all parts needed for a job must be received before the manufacturing stage of the job can commence.

Peng et al. (2009) outline an optimisation model that considers raw coal production, washing and processing, transportation and sales for an integrated coal supply chain. The model includes logistics, capital flows, and information flows between process nodes. The objective is to maximise overall profit and customer satisfaction. The authors describe an application of the model to coal production at the Xuzhou coal mines in China. Conradie et al. (2008) address the problem of the operational scheduling of coal extraction, stacking and reclaiming processes across a coal supply chain. The authors consider a multiple objective system. The goals are to: maximize the sum total of all coal stacked on all the yards from all the mines in all time periods, to minimise the amount of excess tons produced, which is not moved to the stockpiles and for which there is no space at the mines, to distribute stacked coal evenly across the yards, and, to meet blend requirements relating to ash and fines content. The objective also considers uncertainty in the demand for coal products. The focus of the modelling is on operational scheduling and a simulated annealing method is used to provide good solutions to the resulting large scale problems in reasonable time.

With respect to rail scheduling, Dorfman & Medanic (2004) consider scheduling trains in a railway network using a local feedback-based travel advance strategy based on a discrete event model of trains moving along rail lines in the network. A greedy “travel advance strategy” heuristic is used to evaluate the times at which trains enter and leave nodes, representing stations, sidings and passing loops in the network. A capacity check algorithm is used to prevent deadlock and increase the computational efficiency of the method. Burdett & Kozan (2010) describe a hybrid job shop approach to representing and constructing train timetables. A constructive algorithm is used to implement the model together with simulated annealing and local search meta-heuristic improvement algorithms. The objective is to minimise the makespan, which is a proxy for railing throughput, of the system.

Ahmed & Sahinidis (2003) describe a stochastic integer programming method for capacity expansion planning. Their model is structured to determine future expansion times, sizes and locations to meet anticipated growth in demand. Specifically, the model addresses the problem of determining the timing and level of capacity expansions for a set of generalised production facilities, along with a policy for allocating available capacity to a set of products, while minimising expected investment and capacity allocation costs. Product demand, capacity expansion and capacity allocation costs are assumed to be stochastic. The model is constructed using a multistage stochastic programming formulation over a planning horizon of $n$ periods. The authors note that the resulting model is NP hard and, at a practical level, cannot be solved using standard integer programming techniques. A heuristic method based on relaxed linear program and decomposition methods is used to solve a number of test problems in substantially less time than required by the integer program models.

Huh et al. (2006) consider capacity planning under uncertainty using a network optimisation based heuristic to determine the sequence and timing for purchasing and retiring machines required to manufacture different products. The authors also consider variance reduction strategies that can be used with certain stochastic demand forecast distributions.

Birge (2000) adopts option pricing methods from the finance sector to address the risk associated with operations management. The paper considers the incorporation of risk into capacity
planning models in order to assess the allocation of additional capacity in environments characterised by limited resources and uncertainty. A capacity expansion example, involving a decision about whether to install additional capacity at a plant based on possible increases in demand for different products, is used to demonstrate the risk inclusion principles. Essentially, the problem is to trade off the costs of additional capacity against potential revenue from additional sales.

3. Characteristics of the Hunter Valley coal chain

In this section we discuss a number of characteristics of the coal supply chain that influence the development of models of its operation. We present the model formulation in the next section.

As background, Bayer et al. (2009) provide an overview of the Australian coal mining industry and coal supply chains with a focus on potential increases in exports based on expansions in mine, rail and port capacities. The authors consider planned development of new mines and expansion of transport and port facilities for the New South Wales coal chains feeding the Newcastle terminal considered in this paper, and the Queensland coal chains feeding the Gladstone, Dalrymple Bay/Hay Point, and Abbot Point terminals.

Figure 1 shows a simplified diagram of the coal supply chain and the three coal terminals, Port1, Port2, Port3.

Further information on the coal chain system is given in the handbook published by Port Waratah Coal Services Limited (2007), and on the HVCCC¹, NCIG², and PWCS³ web sites.

The activities in the coal chain, and hence the model, are driven by the need to satisfy a coal demand pattern generated by ship arrivals.

Stockpiles are built to service the arriving ships, with each ship having its own set of stockpiles. The stockpiles are built from coal railed from the mines. Coal arriving at a terminal is stacked into a stockyard then reclaimed and sent to a ship-loader serving the berth at which the ship waits.

3.1. Ship arrivals

Contracts for coal result in ship arrivals at the terminals and these represent a nominated throughput, in Mtpa, for the coal chain. The model uses historical ship arrival and cargo data to specify a throughput scenario. The data for an arriving ship specifies an arrival time, and provides information on the number of different coal brands to be loaded, together with the sub-tonnages of products from different mines that make up the blend for each brand. The coal brand data are used to specify the number of stockpiles of each brand needed for the ship, and the tonnages that need to be railed from different mines to build each stockpile.

3.2. Estimating ship load times

The goal of the optimisation model is to minimise the overall cost of running the terminal at a specified annual throughput. It balances the costs of additional equipment against the demurrage costs associated with loading the ships. Demurrage is a penalty payment incurred by the terminal operator when a ship is not loaded within a contracted time following its arrival. In practice, the calculation of the demurrage cost for a ship is complicated, it is subject to an agreement associated with the loading contract for the ship and varies with the size of the ship.

The optimisation model uses a simplified demurrage calculation based only on the delay in loading a ship. We therefore need to know when, under an efficient use of the facilities in a given modelling scenario, we would expect a ship to be loaded given its arrival time relative to the arrivals of the other ships at the terminals.

We use a berthing simulation to determine when the ships for each terminal are berthed and when they are expected to begin and end loading. The simulation assumes that all the coal stockpiles needed for a ship are available in the stockyard when the ship berths, that is, that the rail and stockyard unloading systems can always meet the demand. The machines used to reclaim coal and load it onto the ships are run at their nominal rates adjusted by historical efficiency factors with maintenance outages taken into account. These assumptions are used as a proxy for the system operating in such a way that all contractual agreements can be met, there are no delays to ship loading, and no demurrage costs are incurred.

The simulation takes ships from a queue ordered on arrival times and puts them, on a first-come first-served basis, into the earliest available berth. The ship is loaded by the earliest available shiploader that can service the berth.

![Ship Loading Activities](image)

Figure 2: Ship berthing simulation for a terminal

Figure 2 shows the procedure used to determine the expected berth time and the times for start and completion of loading for each ship. The time when a berth becomes available after loading a ship is shown as ‘Berth Available’ in the diagram. The berthing process for a ship begins at the later of the time when a berth becomes available and the arrival time of the ship.
The hatch change, brand and deballast delays, and the times to prepare a ship once a berth is available and to sail it once loading is completed, are specified as data.

The optimisation model uses the expected completion of loading times from the simulation as reference points for calculating demurrage costs for the ships. If $F_{v,\text{load}}$ is the expected completion time for loading vessel $v$ calculated from the simulation, and $f_{v,\text{load}}$ is the actual completion of loading time for $v$ in the model, then the delay for vessel $v$ is (see also the discussion around constraints 22–31)

$$\delta_v = \max\left\{0, f_{v,\text{load}} - F_{v,\text{load}}\right\} \text{ hours.}$$

The total demurrage cost for vessel $v$ is then calculated from $\delta_v$ using a per hour cost taken from a specified daily demurrage penalty (typically around $20,000 per day).

3.3. Managing stockpiles for vessels

Each vessel is loaded with a number of coal brands and the brands must be loaded in a specified order. Figure 3 shows an example with 3 brands loaded in the order $b_1 \rightarrow b_2 \rightarrow b_3$. There is some changeover delay between loading the brands.

Each brand on a given vessel has its own stockpile in the stockyard. A stockpile is opened and stacked on one of the pads in the stockyard so that all the coal will be ready in time to start loading the brand onto the vessel when it arrives. The stockpile is then reclaimed and closed once all the coal is loaded.

If, as in Figure 3, $L_v$ is the time to begin reclaiming coal and outloading it to vessel $v$ then all the coal for all the stockpiles for that vessel must have been inloaded from the trains and stacked by $L_v$. Using $F_{s,\text{stack}}$ as the latest finish time for stacking stockpile $s$ gives

$$F_{s,\text{stack}} = L_v$$

and all the rail jobs for building stockpile $s$ must be completed before $F_{s,\text{stack}}$.

3.3.1. Railling coal to stockpiles

A set of train trips, or jobs, is needed to build each stockpile. We use the simple assumption that the last of these jobs should be completed just before we want to start reclaiming, that is at $F_{s,\text{stack}}$ for stockpile $s$, and that the finish time for the jobs, $f_{1,\text{job}}, f_{2,\text{job}},$ etc, are separated by at least the unloading time for each train. The coal brand in each stockpile is blended from fractions of...

---

**Figure 3:** Stacking and reclaiming stockpiles for a ship
different products. Each product comes from a different mine, \( m_1, m_2, \) etc. Figure 4 shows how a stockpile is built from a number of different jobs.

We use this information to construct a series of jobs to deliver coal to the stockpiles.

The number of jobs for a given mine is calculated from the required tonnage and the preferred consist size for serving the mine.

We calculate the total trip time for these jobs based on the loading rates and pre and post load delays for the mine load point, the travel times between the load point and its associated junction, and the up and down travel times from the junction to the port.

The due time for the jobs is calculated from the time at which the coal must be stacked and the load rates of the dump stations.

We also set up mappings of the junctions through which each train must pass on its way from and to the port for each job. These mappings are used to deal with different branches on the rail network.

### 3.3.2. Rail factors

Trains are run by a number of different rail operators. The rail operators can supply a number of consists.

In practice, train journeys are scheduled on the rail network according to a set of train paths. These are times at which trains travelling to the mines (down paths) and to the terminals (up paths) can enter rail junctions on the main line.

Trains are loaded at mine load points which are accessed via the rail junctions. The trains for a mine leave the main line at a specified junction, travel to the load point, are loaded, and then re-enter the main line at that junction. Load points have preferred and allowed train sizes and these must be observed when allocating consists to jobs.

### 4. Coal chain modelling considerations

The characteristics described above limit the possible operations of the facilities and equipment used to deliver coal from the mines to ships. These limits are modelled as constraints that guide the behaviour of the optimisation model.
The model works at a daily level, all capacities are specified for each day in the planning horizon, with maintenance and other factors used to adjust the capacities on particular days.

The discretisation of the planning horizon into days of 24 hour duration provides, without loss of generality, an appropriate level of granularity for the system considered in the model. All current activities in the coal chain can be carried out within a day, and activities are taken to occur on the day of their completion. However, the model formulation given below could also be used with the planning horizon divided into intervals of 12 hours, 36 hours, or some other duration that suitably accommodated the coal chain activities. Treating the basic time interval in the model with some flexibility means that we can use the current formulation to deal with systems with operational durations greater, or much less, than 24 hours.

While the model uses daily capacities to set constraints on the railing and terminal operations, the data for train trip times and machine rates at the mines and terminals are specified in terms of hours. We calculate the completion times for different activities in hours and then use that time to determine, as noted above, the day on which the activity is taken to occur. The associated daily capacity constraints are then used to constrain the activities.

4.1. Railing constraints

We impose the following constraints on the trains to ensure that they obey appropriate capacity limits.

1. The number of trains per day from a given rail operator cannot exceed the number of consists available from the operator.
2. The number of trains per day on each track section cannot exceed the number of paths available on each day of the week.

We do not explicitly use train path times for scheduling because the particular rail schedule configurations for different expansion scenarios are not known in advance. Instead, we use the total number of trains that can pass through each junction on each day to specify the railing capacity across the network. The daily junction train capacities are given as data for the model.
3. The maximum number of trains per day to a load point must observe the daily loading capacity (tonnes per day) and number of trains (trains per day) for the mine load point.
4. All the jobs for the stockpiles associated with a given ship should be completed so that the trains can be unloaded and the stockpiles stacked by the time we expect to start loading the ship. We cannot start loading a ship until all the coal for the ship is on the ground in the stockyard. That is until after the actual completion of stacking in the model for the last job associated with the stockpile.
5. Preferred train sizes at mine load points are used when assigning jobs to a mine.

4.2. Terminal operations constraints

As with the railing system, we impose a series of constraints to ensure that appropriate capacity constraints are obeyed. The current constraints require that:

1. The total dumping (train unloading) capacity at a terminal per day is given by the sum of the daily capacities of each dump station at the terminal. The capacity of a dump station is given by its nominal dump rate multiplied by the effective number of hours it operates per day. The effective operating hours per day reflects breakdowns and other unplanned stoppages.
We also use planned maintenance data to reduce the capacities on corresponding days across the planning horizon.

2. The total stacking capacity per day is similarly calculated from the number of stackers and their stacking rates.

3. The total daily receiveal capacity of the terminal is the minimum of the dumping and stacking capacities.

4. The total number of trains that can be unloaded at a terminal each day is determined by the receiveal capacity. The time taken to unload a train is calculated from its size and the average receiveal rate. A post unloading delay is added to the unload time for each train to allow for clearance operations.

5. The total daily outloading capacity of the terminal is similarly determined from the minimum of the ship-loading capacity and the reclaiming capacity.

6. The amount of coal that can be loaded per day to a ship from a stockpile is determined from the daily outloading capacity of its berth. The total amount of coal that can be outloaded to ships at berth is limited by the total daily outloading capacity of the terminal.

7. The loading start time for a ship is taken as the maximum of the estimated start load time, calculated from the shipping and the latest completion time for stacking the ship’s stockpiles. The loading completion time for a ship is then calculated from the loading start time and the average outloading rate. We allow for a delay between finishing stacking/reclaiming one stockpile and starting to stack/reclaim the next, and also between finishing reclaiming and starting stacking.

The day on which ship loading is complete is determined from its loading start time, its loading duration and the daily outloading capacity at the terminal.

8. The amount of coal in a terminal on day \( d \) is the amount on day \( (d - 1) \) less the amount unloaded on day \( d \) plus the amount received on day \( d \).

The amount of coal in a terminal on day \( d \) must be greater than or equal to zero and less than or equal to the maximum capacity of the stockyard.

These constraints ensure that we cannot overload the stockyard or take out more than is available.

4.3. The objective function

The constraints in the model allow us to determine, among other things, completion times for stacking the stockpiles and reclaiming them to the ships at berth. We use these variables to penalise late operations in the objective function.

We impose a cost penalty if the completion of reclaiming for a stockpile is later than the expected time to complete loading, as calculated from the berthing simulation.

As described above, there are constraints relating capacity limits for train loading, railing, dumping, stacking, reclaiming and ship-loading to the numbers of load points, train paths, dump stations, stackers, reclaimers and ship-loaders in the system.

These constraints have been written so that, for example, the stacking capacity is determined by the current number of stackers plus additional stackers running at the average stacking rate (based on the rates of the current stackers) for the terminal.

That is, the model allows for additional machines or facilities but, if used, they incur a fixed capital cost and a variable running cost, based on the type of machine or facility, in the objective.

The objective is to minimise the cost of the sum of the delays to ships, plus the costs of using: additional load stations at a mine, additional paths on different sections of track, and additional dump stations, stackers, reclaimers and ship-loaders at a terminal.
5. The Capacity Expansion Planning Model (CEPM) formulation

Ships arrive at a port which has a number of terminals, each of which has a number of berths. We use $V$ to denote the set of ships arriving at the port during the time horizon $H$ set up for a particular run of the model. Each vessel has an expected time to start loading, $F_{ETL}$, calculated from the shipping simulation.

The vessels are divided into ordered sets (based on arrival times) at each berth. That is, $V = \bigcup_{b \in B} V^b$, where vessel $v^b_i$ directly precedes $v^b_{i+1}$ at berth $b \in B$ and $B$ is the set of berths (at each terminal).

$S_v \in S$ is the set of the stockpiles that are built for vessel $v \in V$, with $S$ denoting the set of all stockpiles. Each stockpile $s \in S_v$ of mass $M_{SP}(s)$ is made up from one or more products $p$. $M_{SP}(s)$ is the mass of product $p$ in stockpile $s$ and $M_{SP} = \sum_{p \in S} M_{SP}(p)$.

Trains deliver each product from a mine to a stockyard and each train trip is called a job. Stockpile $s$ is assembled from a set of jobs $J_s$ and $J$ is the set of all jobs that must be done over the time horizon $H$.

A number of different train types $\tau$ are used to perform the jobs in the CEPM. In practice, only certain train types can be sent to a mine, because of loading and track restrictions, and each mine has a preferred train type which is specified in the input data. The model uses only the preferred train type $\tau_j$, with capacity $C_{\tau_j}$, for job $j$ carrying coal from a given mine to the stockyard. We use $d_j$ to denote the trip time of job $j$.

$E$ is the effective work hours per day.

5.1. Train jobs

The mass of job $j \in J$ is $m_{\text{job}}^j$ and for the CEPM we have

$$m_{\text{job}}^j = C_{\tau_j} \quad (2)$$

and the number of jobs needed to assemble stockpile $s$ is

$$\sum_{p \in S_j} \sum_{s \in S_j} \left\lceil \frac{M_{SP}(s)}{C_{\tau_j}} \right\rceil \quad (3)$$

The completion time for job $j$ is $f_{\text{job}}^j$ and defining the day on which job $j$ finishes using

$$u_{\text{job}}^{jd} = \begin{cases} 1 & \text{if job } j \text{ finishes on day } d \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

we can write

$$f_{\text{job}}^j \geq u_{\text{job}}^{jd} F_{\text{day}}^{d-1} \quad \forall d \in H, \forall j \in J \quad (5)$$

$$f_{\text{job}}^j \leq F_{\text{day}}^d + \Omega(1 - u_{\text{job}}^{jd}) \quad \forall d \in H, \forall j \in J, \quad (6)$$

where $F_{\text{day}}^d$ denotes the time (in hours) from the beginning of $H$ to the end of the day $d$.

We also have

$$\sum_{d \in H} u_{\text{job}}^{jd} = 1 \quad \forall j \in J. \quad (7)$$
We also need to define binary variables for assigning each job $j$ to a pad $p$ in the stockyard on day $d$ as follows
\[
 u_{jpd}^{\text{job}} = \begin{cases} 
 1 & \text{if job } j \text{ is assigned pad to } p \text{ and finishes on day } d \\
 0 & \text{otherwise,}
\end{cases}
\] (8)

and then use the following constraints to ensure that a job is assigned to only one pad
\[
 u_{j}^{\text{job}} = \sum_{p \in P_y} u_{jpd}^{\text{job}},
\] (9)

where $y \in Y$ is the terminal stockyard to which job $j$ was assigned and $P_y$ is the set of pads in this stockyard.

Jobs are done using trains assembled from particular wagon types and there is a limit on the total number of wagons of each type available per day. So if $w_j$ is the wagon type assigned to job $j$ we have:
\[
 \sum_{j \in J} w_j \sum_{d \in H} N_{wj} \leq N_w + s_w \forall d \in H
\] (10)

where $N_{wj}$ is the number of wagons of type $w$ assembled for the train needed for job $j$, $N_w$ is the number of wagons of type $w$ available per day, and we use $s_w$ to represent the extra $w$-type wagons per day that might be needed to satisfy the jobs on day $d$. The term $s_w$ is assigned a capacity expansion cost in the objective function (39).

The railroad track sections have limited capacity. We use the junction capacities associated with the train path data to limit the number of jobs that can run on each section on each day of the week.

If $J_t$ is the set of jobs using track section $t \in T$, where $T$ is the set of all track sections, then
\[
 \sum_{j \in J_t} u_{j}^{\text{job}} \leq N_{th} + s_t \forall d \in H \forall t \in T
\] (11)

where $N_{th}$ is the number of train paths available on section $t$ on day $h$ of the week ($h = d \% 7$) and $s_t$ is the extra train paths that could be used, at a cost given in the objective function (39), on section $t$ on day $d$.

5.2. Loadpoint capacities

Each mine is serviced by a load point $l \in L$ where $L$ is the set of all load points. Each load point has a daily capacity $C_l \forall l \in L$ representing the maximum amount of coal that can be loaded onto trains per day. If $J_l$ is the set of jobs serviced at load point $l$, then
\[
 \sum_{j \in J_l} u_{j}^{\text{job}} \leq C_l + s_l^{\text{LPTones}} \forall l \in L, \forall d \in H
\] (12)

where $s_l^{\text{LPTones}}$ is the additional daily capacity (in tonnes) that could be used, at a cost defined in the objective function (39), at load point $l$ on day $d$.

We can also write the load point capacities in terms of the number of trains that can be serviced each day
\[
 \sum_{j \in J_l} u_{j}^{\text{job}} \leq N_l + s_l^{\text{LPTrains}} \forall l \in L, \forall d \in H
\] (13)
where \( N_l \) is the number of trains that can enter load point \( l \) per day and \( s^{TP}_{l, \text{Trains}} \) is the additional number of trains on day \( d \) that could be loaded, again at a cost given in the objective function (39), at load point \( l \) to meet the demand.

### 5.3. Receival capacities

The amount of coal that can be received at a terminal on each day is limited by the dump station capacity at the terminal. If \( J_y \) is the set of jobs arriving at stockyard \( y \in Y \), where \( Y \) is the set of all stockyards, then

\[
\sum_{j \in J_y} n^\text{job}_j u^\text{job}_{jd} \leq C^\text{dump}_{yd} + \rho^\text{dump}_y E^\text{dump}_y \quad \forall y \in Y, \forall d \in \mathcal{H} \tag{14}
\]

where \( C^\text{dump}_{yd} \) is the receival capacity in tonnes, allowing for maintenance downtime of the dump stations, at stockyard \( y \) on day \( d \). The term \( \rho^\text{dump}_y E \) is the average amount of coal that can be unloaded per day by a dump station at the terminal. The average dumping rate per station, \( \rho^\text{dump}_y \), is calculated from daily capacities averaged over a year. The additional dump stations that could be used, at a cost specified in the objective function (39), on day \( d \) at stockyard \( y \) is \( s^\text{dump}_y \).

If the model gives a nonzero value for \( s^\text{dump}_y \), then we need additional capacity which could be obtained by either increasing the number of dump stations or increasing the rates of the existing dump stations.

The receival capacity can additionally be defined in terms of the hours needed to unload some number of trains per day. If \( D^\text{dump}_y \) is the turnaround time of a train at stockyard \( y \), then we can write

\[
\sum_{j \in J_y} u^\text{job}_{jd} \left( D^\text{dump}_y + \frac{n^\text{job}_j}{\rho^\text{dump}_y} \right) \leq C^\text{dump}_{yd} + \frac{E^\text{dump}_y}{\rho^\text{dump}_y} + s^\text{dump}_y \quad \forall y \in Y, \forall d \in \mathcal{H} \tag{15}
\]

where constraints (15) say that the total time consumed in unloading trains at a stockyard on a given day must be less than the total daily hours available from the dump stations. Note that (15) gives (14) as a particular case when \( D^\text{dump}_y \) is zero.

### 5.4. Stacking capacities

The stacking capacities are modelled similarly to the receival capacities. The arrival at the terminal of the train for job \( j \) is given by

\[
f^\text{job}_j \geq d_j + \sum_{d \in \mathcal{H}} F^\text{day}_{d-1} u^\text{job}_{jd} \quad \forall j \in J \tag{16}
\]

making (5) redundant, and then, to ensure that stacking to a stockpile does not commence before the associated trains arrive, we need the finish time for stacking job \( j \) to be

\[
f^\text{stk}_j \geq f^\text{job}_j + \frac{n^\text{job}_j}{\min \{ \rho^\text{stk}_y, \rho^\text{dump}_y \}} \tag{17}
\]

where \( \rho^\text{stk}_y \) is the average stacking rate at terminal \( y \).
The amount stacked on each day and on each pad is limited by the stacking capacity of the pad in the stockyard and can be expressed as follows:

\[
\sum_{j \in J_y} y_{jpad}^p \left( D_{stk}^y + m^y_{job} \right) \leq C_{stk}^ypd \rho_{stk}^y + E_{stk}^p \forall p \in P_y \forall y \in Y, \forall d \in \mathcal{H} \tag{18}
\]

where \( C_{stk}^ypd \) is the total stacking capacity in tonnes for pad \( p \), allowing for maintenance of the machines, at stockyard \( y \) on day \( d \), \( D_{stk}^y \) is the turnaround time at terminal \( y \) and \( s_{stk}^y \) is the number of costed additional stackers that could be used on pad \( p \) of stockyard \( y \) on day \( d \).

In practice a stacker can service more than one pad. If \( p_1 \) and \( p_2 \) are two pads in stockyard \( y \) which have at least one stacker in common and \( C_{yp1:p2:d} \) is the total stacking capacity common to pads \( p_1 \) and \( p_2 \) on day \( d \), then we use the following constraint to ensure that the overall capacity is not exceeded

\[
\sum_{j \in J_y} y_{jpad}^p \left( D_{stk}^y + m^y_{job} \right) + \sum_{j \in J_y} y_{jpad}^{p_2} \left( D_{stk}^y + m^y_{job} \right) \leq C_{yp1:p2:d} \rho_{stk}^y + E_{stk}^p \forall p \in P_y \forall y \in Y, \forall d \in \mathcal{H} \tag{19}
\]

If \( f_{stk}^s \) is the completion time for stacking stockpile \( s \) and \( f_{stk}^v \) is the time when stacking of all the stockpiles in \( S_v \) for vessel \( v \in V \) has been completed. Then

\[
f_{stk}^s = \max_{j \in J_s} f_{stk}^j, \tag{20}
\]

and

\[
f_{stk}^v = \max_{s \in S_v} f_{stk}^s. \tag{21}
\]

### 5.5. Loading constraints

Analogously to the \( y_{jpad}^p \) variables, we define

\[
u_{spd} = \begin{cases} 1 & \text{if SP } s \text{ finishes loading from pad } p \text{ onto a vessel on day } d \\ 0 & \text{else,} \end{cases} \tag{22}
\]

as the variables denoting the day on which loading of stockpile \( s \) onto its respective vessel is completed.

Then

\[
\sum_d \sum_p u_{spd} = 1 \quad \forall s \in S_v \forall v \in V. \tag{23}
\]

and similarly to constraints (5) and (6), we have

\[
f_{load}^s \geq \sum_p u_{spd} f_{load}^{p,day} \tag{24}
\]

\[
f_{load}^s \leq f_{day}^s + \Omega (1 - \sum_p u_{spd}) \tag{25}
\]

If \( F_{ETL}^v \) is the time to begin outloading to vessel \( v \) then all stockpiles in \( S_v \) must have been stacked by that time.
The shipping simulation determines $F_v^{\text{load}}$ as the completion time for loading vessel $v$. We use $f_v^{\text{load}}$ to represent the actual time when $v$ is loaded. Hence the delay for vessel $v$ is

$$\delta_v = \max \left\{ 0, f_v^{\text{load}} - F_v^{\text{load}} \right\}.$$  

(26)

Additionally, a vessel $v$ might be delayed due to the tardiness of the previous ship loading at the same berth. If we use $\Delta_v$ to represent the time interval between sailing vessel $v$ from a berth and bringing the next vessel into that berth (see Fig 5), and $\eta_v$ to represent the delay to vessel $v$ caused by the tardiness of the previous ship on that berth, then, if vessel $v$ is the $i^{th}$ ship to be loaded at berth $b$

$$\eta_v = \max \left\{ 0, \delta_{v_{b-1}} - \Delta_{v_{b-1}} \right\}$$  

(27)

where $v_{b-1}$ represents the previous ship loaded at berth $b$.

![Figure 5: Loading delay due to tardiness of the previous ship at the berth](image)

If $\rho_v^{\text{load}}$ is the ship loading rate calculated from the shipping simulation, then assuming that the order in which the stockpiles $S_v$ must be loaded is given by the predetermined sequence $s_1, s_2, \ldots, s_n$ with $n = |S_v|$, the loading of $s_1$ is restricted by

$$f_{s_1}^{\text{load}} \geq f_{s_1}^{\text{sink}} + \frac{M_{s_1}^{\text{SP}}}{\rho_v^{\text{load}}}$$  

(28)

and

$$f_{s_1}^{\text{load}} \geq F_v^{\text{ETL}} + \eta_v + \frac{M_{s_1}^{\text{SP}}}{\rho_v^{\text{load}}}.$$  

(29)

Each subsequent stockpile finishes loading at

$$f_{s_i}^{\text{load}} \geq f_{s_{i-1}}^{\text{load}} + \frac{M_{s_i}^{\text{SP}}}{\rho_v^{\text{load}}} i = 2, \ldots, |S_v|,$$  

(30)

and the actual time at which loading is completed for the ship is

$$f_v^{\text{load}} \geq f_{s_n}^{\text{load}}.$$  

(31)
5.6. Stockpile–Job–Pad Assignments

Every job in a stockpile must be assigned to the same pad as that for its associated stockpile. We use the following constraint

\[ u_{load}^p l s \leq d \sum_{d'=1}^{d+1} \sum_{j \in J_s} u_{jpad}^p \quad \forall p \in P_y, \forall s \in S \forall d \in \mathcal{H} \]  

(32)

where \( y_s \) is the stockyard in which stockpile \( s \) is stacked, to ensure this. Since for each \( j u_{jpad}^p \) can only be 1 for a single pad and day (9), once \( u_{load}^p l s = 1 \) it forces all the train jobs for this stockpile \( s \) to be allocated in pad \( p \) and not elsewhere.

5.7. Reclaiming capacities

We also need to consider the reclaiming capacity of a stock yard on each day. As for the receival constraints (14) and (15), we write

\[ \sum_{s \in S} u_{load}^p l s y_s \leq C_{rclm}^y p d + \rho_{rclm}^y E_{s y p d} \quad \forall p \in P_y \forall y \in Y, \forall d \in \mathcal{H} \]  

(33)

\[ \sum_{s \in S} u_{load}^p l s y_s \leq C_{rclm}^y p d + \rho_{rclm}^y E_{s y p d} \quad \forall p \in P_y \forall y \in Y, \forall d \in \mathcal{H} \]  

(34)

where \( C_{rclm}^y p d \) is the reclaim capacity in tonnes at pad \( p \), allowing for maintenance, \( D_{rclm}^y y p d \) is the turnaround of a reclaimer between two stock piles, \( \rho_{rclm}^y \) is the average reclaiming rate at \( y \), and \( s_{rclm}^y p d \) is the additional number of reclaimers needed on day \( d \) at pad \( p \) to satisfy the demand with minimum cost.

Again, a reclaimer may service more than one pad. Let \( p_1 \) and \( p_2 \) be two pads in stockyard \( y \) which have at least one reclaimer in common and \( C_{rclm}^y p d \) be the total reclaiming capacity common to pads \( p_1 \) and \( p_2 \) on day \( d \). We use the following constraint to ensure that the overall capacity is not exceeded

\[ \sum_{s \in S} u_{load}^p l s y_s \leq C_{rclm}^y p d - C_{rclm}^y p_{12} d + \rho_{rclm}^y E_{s y p d} + s_{rclm}^y \quad \forall p \in P_y \forall y \in Y, \forall d \in \mathcal{H} \]  

(35)

5.8. Stacker-Reclaimer Capacity

Some pads may be served by stacker-reclaimers that can either stack or reclaim, but not both, at any given time. If \( CSR_{rclm}^y p d \) (\( CSR_{sp}^y p d \)) are the stacking (reclaiming) capacities of a stacker-reclaimer on pad \( p \) of stockyard \( y \) on day \( d \), then we use the following constraint to restrict the machine’s overall operational capacity.

\[ \sum_{s \in S} u_{load}^p l s y_s \leq C_{rclm}^y p d - C_{rclm}^y p_{12} d + \rho_{rclm}^y E_{s y p d} + s_{rclm}^y \quad \forall p \in P_y \forall y \in Y, \forall d \in \mathcal{H} \]  

(36)
5.9. Stockyard inventories

If the capacity (in metres) of pad \( p \) in stockyard \( y \) is \( C_{py}^{SY} \), \( \theta_{ypd} \) is the metre-inventory level on pad \( p \) of stockyard \( y \) at the end of day \( d \), and \( G_{job}^{sp}(G_{a}^{sp}) \) is the size, in metres, of job \( j \) (stockpile \( s \) in the stockyard, then

\[
\theta_{ypd} = \theta_{ypd-1} + \sum_{j \in J} u_{jpd}^{pad} G_{job}^{sp} - \sum_{s \in S} u_{spd}^{load} G_{a}^{sp} \quad \forall p \in P_{y}, y \in Y, d = 2, \ldots, |H| \quad (37)
\]

where \( \theta_{yp0} = 0 \) \( \forall p \in P_{y}, y \in Y \), and \( P_{y} \) is the set of all the pads in stockyard \( y \). We assume all stockpiles are empty at the start of the planning horizon and use a ‘warmup’ period to allow coal to be delivered for the first ships at the berths.

In addition, we use the following constraint to limit the amount of material that is in stockyard pad \( p \) on day \( d \).

\[
\theta_{ypd} \leq C_{py}^{SY} + s_{py}^{SYMetres}, \quad (38)
\]

where \( s_{py}^{SYMetres} \) is the additional daily capacity in metres that could be used, at a cost given in the objective function (39), on pad \( p \) of stockyard \( y \).

5.10. The objective function

The model minimizes the overall cost of meeting a demand defined in terms of the throughput specified by a set of vessel arrivals. We balance the overall demurrage (delays to vessels) against the cost of installing additional machines such as stackers, reclaimers, dump stations, or trains, and providing additional capacity for facilities such as load points, stockpile pads, or train paths, across the coal chain.

The objective is

\[
\sum_{v \in V} \Pi_{v} \delta_{v} + \sum_{w} \alpha_{1} s_{w} + \sum_{d} \sum_{t \in T} \alpha_{2} s_{t} + \sum_{l \in L} (\alpha_{3} s_{l}^{LP} + \alpha_{4} s_{l}^{LP}) + \sum_{y \in Y} (\alpha_{5} s_{y}^{dump} + \alpha_{6} s_{y}^{dumpTime} + \alpha_{7} s_{y}^{stk} + \alpha_{8} s_{y}^{clm} + \alpha_{9} s_{y}^{SYMetres}) \quad (39)
\]

where \( \Pi \) is the per hour demurrage cost for delayed vessels and the \( \alpha_{i} \) are the costs of providing additional machines and operating capacities.

6. Solution methods

The model described above allows for potentially very long ship delays. In practice this represents an unsatisfactory key performance indicator for the terminal operator. As part of the implementation of the model we restrict the maximum delay for loading a ship to a specified number of days (typically set at 10 days in the input data). This is consistent with the performance guarantees often given as part of the contractual agreements for ship loading. It also creates a time window within which jobs must be carried out and this reduces the model size.

CPLEX is used as the base method to solve a CEPM model scenario and obtain an exact integer solution within specified CPLEX mip gap tolerances. However, a typical scenario looks at a 180 day planning horizon with a 1 day granularity and CPLEX cannot solve the resulting models within 24 hours, which is considered the upper level for getting an acceptable solution.
For practical applications of the model we select from a number of heuristics to obtain good solutions within a reasonable time. The best heuristic solution at the end of a heuristic search, is passed as a starting point for CPLEX to generate a final solution. This also ensures that the cost evaluation is consistent across the different heuristic methods. In this final evaluation the CPLEX model is restricted to keep the pad allocation fixed as per the heuristic solution and to limit the possible days on which a job or stockpile can be done to a time window of 2 days based on the heuristic solution day.

The user can specify a time limit, typically one to a few hours, for the heuristic and a separate limit for the subsequent CPLEX solution resolution. CPLEX can take anywhere from a few minutes to many hours to resolve the heuristic solution.

The different solution methods available to the user are:
1. CPLEX
2. Genetic Algorithm Heuristic
3. Squeaky Wheel Heuristic
4. Large Neighbourhood Search

6.1. Constructive heuristic

The constructive heuristic is a stand-alone subroutine that develops an operating schedule from a given list of job sequences and potential equipment capacity expansions. The objective function (39) is used to evaluate this schedule as a candidate solution for a suitable search method.

The constructive heuristic is essentially a greedy algorithm that calculates a tardiness cost for ship loading plus an expansion cost for the machines, and works as follows.

1. It reads two candidate lists: one is a sequence of the jobs required to service the arriving ships and the other is a vector of additional capacity expansions for the different machines in the system. An empty expansion vector represents current equipment capacity levels.
2. Each job has a time window representing the earliest time it can start, based on the completion of earlier jobs on which it is dependent, and the latest time it must finish as determined by the maximum ship loading delay discussed earlier.
3. The job sequence list gives the job priorities, where an earlier job in the list is scheduled first. Using machine capacities (including any proposed expansion) and rates, the jobs are scheduled successively. If we find a job that cannot be completed within its specified time window then we prepare a list of all machines that are currently operating at full capacity and are thus potential bottlenecks.
4. We expand the capacities of these machines which are at full capacities, so that the job can be completed within its time window. For example, if a job requires eight wagons but only six are free then we add an extra two wagons to the system; if a train is available but there is no path then we add a path for that job; if all dump stations are occupied then we add a new dump station.
5. The additional facilities, if any are required, are added to the expansion vector and we rebuild the schedule starting with the first job and using the extra facilities.
6. This process is repeated until, for the specified job sequence, all the ships can be loaded within their assigned time windows. The schedule is then evaluated using the objective function and returned to a search meta-heuristic as a candidate solution.

We use two methods, a Genetic Algorithm (GA) heuristic and a Squeaky Wheel (SW) heuristic, to generate the lists used by the constructive heuristic.
6.2. Genetic Algorithm heuristic

In this section we describe a Genetic Algorithm based heuristic that was implemented for the above-mentioned problem. Unlike the traditional GA, the chromosome structure for the implemented GA contains two lists: a job sequence and an expansion list. The job sequence is made up from the rail jobs (which imply stacking jobs) and stockpile jobs (which imply reclaiming and shiploading jobs). The expansion list holds a list of all the distinct resources/facilities.

The GA starts with an initial population of 100 chromosomes containing a mix of randomly due-date and release-date sorted jobs, and a set of randomly generated expansion levels that specify each resource either at or above its capacity.

The fitness of a chromosome is based on its objective value which is calculated from the costs of any lateness arising from its job sequence and from the expansion levels it specifies for the resources.

The GA evolves through successive generations until it finds a zero cost solution, or a specified time limit is reached. When the GA terminates it passes its solution to CPLEX to generate a final solution.

6.3. Squeaky Wheel heuristic

One of the reasons why the Genetic Algorithm struggles to find good heuristic solutions is that the search space is vast: some of the data sets discussed below involve over 1000 stockpiles and around 10,000 train jobs. Searching the space of permutations with a sequence of over 10,000 items is challenging. Hence we also tried a Squeaky Wheel algorithm that focuses the search in promising areas.

The term “Squeaky Wheel” optimisation was introduced by Joslin & Clements (1999), for an algorithmic approach that involves a cycle of three steps: (1) constructing a solution, (2) analysing the solution to determine which elements are to “blame” for poor quality (3) reprioritising based on the blame. This method tends to find good solutions relatively rapidly by using some problem specific knowledge on the influence of elements in the solution on solution quality to speed up the search.

The Squeaky Wheel (SW) method uses the same solution representation as the GA consisting of a sequence of jobs and expansion levels. It starts with a single random solution and performs a randomised descent. The heuristic uses a local search method on the job and expansion lists, focussed on the days on which extra capacity is indicated to improve the objective. That is the additional capacity requirements are the “squeaky wheels” that need “oilng” in this context. This means trying to move some of the jobs using this additional capacity to a randomly chosen earlier or later point in the job list. In addition any job that is only completed after the due date may be moved to an earlier point. In addition the expansion vector is perturbed in a randomised manner, reducing any expansion that was not actually required or making available additional expansion at the start of the horizon where this expansion proved to be necessary at some later date.

We have also included here a random mutation component to the search which has also been found in Aickelin et al. (2009) to be beneficial for improving the quality of the solutions found. Whenever the solution does not improve for a number of iterations (8 in our numerical tests), the solution is given a larger random perturbation. This is to allow the search to escape local minima in the manner of iterated local search (see Lourenço et al. (2003)).

The search continues until a zero cost solution is found or the time limit specified in the parameter file is reached. When the SW method terminates it passes its solution to CPLEX to generate a final solution.
6.4. Large Neighbourhood Search method

The Large Neighbourhood Search (LNS) method solves a series of CPLEX models constructed for operations over selected intervals within the overall time horizon.

The method first runs the Squeaky Wheel algorithm for 600 CPU seconds to obtain a “good” starting solution.

It then selects the earliest set of 5 days for which either vessels were delayed or resource expansions were required. If \( d \) is the last day in the selected set of 5 days then

a) all jobs before the 5 day interval are fixed (they can be completed with no delays or expansion requirements),

b) all jobs in the current solution with a completion time later than \( d + 5 \) days are fixed,

c) all jobs within the 5 days are given bounds of 2 days before and after their completion times in the current solution,

and the resulting MIP model is solved to find the optimal values for the (relatively small number of) bounded job variables. That is, we solve the original model over the full planning horizon with all but a selected set of variables fixed to values found from a previous solution.

The method then selects the next set of 5 days in which either vessels were delayed or resource expansions were required and solves for this new group of (later) bounded job variables.

This process is repeated until a zero cost solution is found or the time limit specified in the parameter file is reached. If the algorithm reaches the end of the planning horizon before the time limit is reached then it takes the current solution and starts again from the first day.

Given that a large series of CPLEX MIPs are solved using the LNS method, an overall time limit of 5 hours or more is needed to get good solutions for reasonably large models.

7. Model validation

The model was validated using historical data to run actual operating scenarios over a nominated six month operating horizon. The outputs from the model were then compared with the recorded performance of the coal chain. Figures 6–8 show some results from a validation run. The horizontal axes show day 20 to day 170, representing ‘steady state’ operations in the model.

When analysing the results we remove two sets of edge effects. The model has a warmup interval that allows the system to initiate railing and prepare stockpiles for the first of the arriving ships. The warmup duration is set by a parameter that offsets all ship arrivals with the first ship arriving at the end of the specified interval. There is no reclaiming or ship loading in the warmup period. A second parameter is used to specify the last operating day in the model. This parameter specifies the number of days for which railing can occur after the estimated time for the completion of loading of the last ship. No ships can be delayed past this point. Railing and stockyard activities will start winding down in this period.

Figure 6 shows the actual ship queue and the queue from the model run. The opening value of the actual queue has been adjusted to correspond to the number of ships arriving in the offshore queue on the first fully operational day, after the railing warmup, for the model run.

The results in Figure 6 are from a run using 2006 data in which the terminal capacities were sufficient to give reasonable queue lengths. So the model minimised demurrage costs without providing additional capacity, reflecting the current operations at the port. The results show that the model has balanced the ship queue while achieving the required coal throughput. That is, the
operational behaviour generated by the model results in a more orderly growth in the queue size, reflecting less variability in the system work flows over the planning period.

Figure 7 shows the actual and model output for the daily trains unloaded at Port1.

The actual operations used an average of 23 trains per day while the model met the required throughput with an average of 22 trains per day.

Figure 8 shows the actual and model output for the daily tonnage dumped at Port1.

The model output is somewhat smoother than the actual. The actual operations resulted in an average 162,631 tonnes per day railed to Port1 while the model railed an average of 158,764 tonnes per day.
8. Computational results

Data for the model are read from a series of csv files prepared by the user. These files specify ship arrivals and products, rail network resources and capacities, terminal resources and capacities, and operational parameters.

Tables 1 and 2 show some of the coal chain operational characteristics. These are based on the current infrastructure in the Hunter Valley system. Additional data specifies the daily capacities of the loadpoints, train paths, terminal machines, and so on.

Table 1: Terminal characteristics

<table>
<thead>
<tr>
<th></th>
<th>Dump Stations</th>
<th>Stackers</th>
<th>Reclaimers</th>
<th>Stacker Reclaimers</th>
<th>Ship Loaders</th>
<th>Pads</th>
<th>Berths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port1</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Port2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td></td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Port3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Rail Network characteristics

<table>
<thead>
<tr>
<th></th>
<th>Mines</th>
<th>Loadpoints</th>
<th>Junctions</th>
<th>Consists</th>
<th>Wagon Type A</th>
<th>Wagon Type B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>47</td>
<td>34</td>
<td>14</td>
<td>14</td>
<td>1766</td>
<td>376</td>
</tr>
</tbody>
</table>

Table 3 shows ship arrival data for two scenarios evaluated by the model. The first, Scenario A, is taken from historical shipping data and represents a throughput of 100 Mt/ta with 580 ship arrivals over a 180 day period. The shipping demand requires 865 separate stockpiles to be built and cleared across the three terminals, and 7,473 train trips to deliver the coal to the terminals from the mines.
Scenario B, represents a planned future throughput level of 140 Mtpa for the port. In this case the ship arrivals were generated by compressing7 a set of ship arrivals taken from historical data. For this scenario we have 795 ships arriving over 180 days, with a resulting 1181 stockpiles and 10,340 train trips.

<table>
<thead>
<tr>
<th>Throughput (Mtpa)</th>
<th>Scenario A</th>
<th>Scenario B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ships</td>
<td>580</td>
<td>795</td>
</tr>
<tr>
<td>Port1</td>
<td>318</td>
<td>433</td>
</tr>
<tr>
<td>Port2</td>
<td>156</td>
<td>214</td>
</tr>
<tr>
<td>Port3</td>
<td>106</td>
<td>148</td>
</tr>
<tr>
<td>Stockpiles</td>
<td>865</td>
<td>1181</td>
</tr>
<tr>
<td>Rail jobs</td>
<td>7473</td>
<td>10340</td>
</tr>
</tbody>
</table>

Table 3: Input data for two 180 day scenarios

8.1. Results for the 100 Mtpa throughput scenario

Table 4 shows the results for scenario A, with a throughput of 100 Mtpa. See Table 3 for information about the problem characteristics.

We have run the GA and SW heuristics for half an hour and one hour, as well as a longer run of 20,000 seconds (over 5 hours) to match the LNS method which cannot produce useful results in much less time than this. Interestingly the SW method, because of additional computation required in determining how to modify the current solution, carries out somewhat fewer iterations (calls to the underlying constructive heuristic) than the GA. Nevertheless it already finds a better solution in half an hour than the GA does with more than 10 times as much CPU time. This is because the very large search space (a sequence of over 8000 jobs plus expansion decisions), means that the GA takes a very long time to converge. A non-trivial amount of effort has been expended by the authors trying to improve the GA result through better tuning of parameters but we were unable to close the gap in performance to SW. Interestingly the LNS does return a better solution than SW though in practice the long run time means that the SW method would be favoured. It also interesting to note that the SW method tends to favour solutions with smaller expansion costs though higher demurrage than the other two methods. This is not entirely surprising given that the way the heuristic is constructed it focusses primarily on reducing expansion costs.

The local neighbourhood search method completes just 13 iterations in the available time. The time required by CPLEX to find a solution to the sub-problem is highly variable ranging from 17 seconds to 6455 seconds. The longer times correspond to later iterations where the search method has expanded the neighbourhood in order to try to find better solutions. Interestingly the biggest improvement comes in the first step which is already sufficient to obtain a solution that is slightly better than the best SW solution.

8.2. Results for the 140 Mtpa throughput scenario

Table 5 shows the results for scenario B, where the target throughput has been increased to a rate of 140 Mtpa.

---

7Reducing the arrival times between ships.
The pattern of results is similar to the first example except that of course now significantly more infrastructure has to be purchased to achieve the desired throughput. Where the 100Mtpa throughput could be achieved through a relatively small increase in the number of trains. At about 80 wagons per train the solutions in Table 4 represent a requirement of no more than two new trains. This contrasts with over 10 new trains required according to the solutions in Table 5 as well as increased train loading capacities at some of the loadpoints and an additional dump station required for unloading of trains at the terminal.

Comparing the different methods we again see that the GA method is not competitive. The SW method outperforms the LNS even though it actually consumed somewhat less CPU time when run with the 20,000 second limit. The LNS method in this case only completed 11 iterations with the last and longest iteration taking just over 2 hours of CPU time. However given the parallel nature of CPLEX 12.1 and the 16 cores available on our machine this iteration took “only” a bit over an hour of real time. As for the previous example the SW solution shows a bias towards lower expansion options even if this comes at the price of additional shipping delays and hence higher demurrage.

<table>
<thead>
<tr>
<th>Heuristic solution</th>
<th>GA</th>
<th>SW</th>
<th>LNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution time (sec)</td>
<td>1800</td>
<td>3600</td>
<td>20000</td>
</tr>
<tr>
<td></td>
<td>24200</td>
<td>47400</td>
<td>247400</td>
</tr>
<tr>
<td></td>
<td>1800</td>
<td>3600</td>
<td>20000</td>
</tr>
<tr>
<td>Iterations</td>
<td>24200</td>
<td>47400</td>
<td>247400</td>
</tr>
<tr>
<td></td>
<td>17250</td>
<td>34831</td>
<td>201206</td>
</tr>
<tr>
<td>Costs ($×10^6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>22.83</td>
<td>18.82</td>
<td>15.96</td>
</tr>
<tr>
<td></td>
<td>14.71</td>
<td>14.71</td>
<td>14.45</td>
</tr>
<tr>
<td>Expansion</td>
<td>16.47</td>
<td>12.22</td>
<td>9.63</td>
</tr>
<tr>
<td></td>
<td>7.15</td>
<td>7.15</td>
<td>6.45</td>
</tr>
<tr>
<td>Demurrage</td>
<td>6.36</td>
<td>6.60</td>
<td>6.34</td>
</tr>
<tr>
<td></td>
<td>7.56</td>
<td>7.56</td>
<td>8.09</td>
</tr>
</tbody>
</table>

| CPLEX resolution   |          |          |          |
| Solution time (sec)| 15.92    | 15.85    | 15.86    |
|                    | 16.64    | 16.65    | 16.81    |
|                    | 19968.53 |          |          |
| Costs ($×10^6)     |          |          |          |
| Total              | 17.41    | 13.34    | 10.99    |
|                    | 9.52     | 9.52     | 9.26     |
|                    | 9.06     |          |          |
| Expansion          | 16.47    | 12.22    | 9.63     |
|                    | 7.15     | 7.15     | 6.45     |
|                    | 6.9      |          |          |
| Demurrage          | 0.93     | 1.13     | 1.36     |
|                    | 2.37     | 2.37     | 2.81     |
|                    | 2.16     |          |          |

| Additional Infrastructure |          |          |          |
| Terminal Machines         |          |          |          |
| Type A Wagons             | 243      | 184      | 179      |
| Type B Wagons             | 143      | 143      | 129      |
| Train Paths               | 138      |          |          |
| Loadpoint trains/day      | 48066.36 | 35041.66 | 6750.38  |
| Loadpoint tonnes/day      |          |          |          |

| Shipping Delays |          |          |          |
| Ships delayed   | 92       | 99       | 99       |
| Total Delay (hrs)| 936.9  | 1126     | 1360.2   |
| Average Delay (all ships) | 1.61   | 1.94     | 2.34     |

Table 4: Solutions for Scenario A of expanding the supply chain to 100 Mtpa throughput.
8.3. A note on the lower bound for the problem

We can obtain a lower bound for the solution by relaxing the integer and binary variables and solving the resulting continuous model using CPLEX. This solution does not provide a proper physical solution because it allows for ‘fractional’ expansion, for example, adding an extra 0.4 of a stacker. The lower bound is therefore not particularly good.

Using CPLEX to solve the relaxed model for the 100 Mtpa scenario gave a lower bound of $5.499 \times 10^6$, with a solution time of 43998.7 seconds. The best integer solution from Table 4 is, for the LNS method, $9.06 \times 10^6$.

For 140 Mtpa scenario CPLEX could not solve the relaxed model even after 180,000 seconds. The last solution from CPLEX was $3.8994 \times 10^7$. The best integer solution from Table 5 is, for the Squeaky Wheel method, $8.3410 \times 10^7$.

9. Conclusion

We have outlined the constraints and approaches used for the development of a large scale capacity planning model for a coal supply chain. The model is currently being used by the Hunter
Valley Coal Chain Logistics Team to evaluate least cost expansion options as part of the Hunter Valley master planning process.

The brief numerical results show that this is a challenging problem that cannot be solved easily with currently available commercial MILP solvers or straight application of general meta-heuristics like genetic algorithms. The customised squeaky wheel and large neighbourhood search heuristics perform somewhat better but there is undoubtedly room to produce better optimisation methods, particularly by exploring alternative or more closely coupled hybridisations between the MILP approaches and heuristic search methods.

References


