CP and MIP Methods for Ship Scheduling with Time-Varying Draft

Elena Kelareva[1] **Sebastian Brand**[2] **Philip Kilby**[1] **Sylvie Thiébaux**[1] **Mark Wallace**[3]
(1) ANU and NICTA, Canberra, Australia, elena.kelareva@nicta.com.au
(2) University of Melbourne and NICTA, Melbourne, Australia
(3) Monash University and NICTA, Melbourne, Australia

Abstract

Existing ship scheduling approaches either ignore constraints on ship draft (distance between the waterline and the keel), or model these in very simple ways, such as a constant draft limit that does not change with time. However, in most ports the draft restriction changes over time due to variation in environmental conditions. More accurate consideration of draft constraints would allow more cargo to be scheduled for transport on the same set of ships.

We present constraint programming (CP) and mixed integer programming (MIP) models for the problem of scheduling ships at a port with time-varying draft constraints so as to optimise cargo throughput at the port. We also investigate the effect of several variations to the CP model, including a model containing sequence variables, and a model with ordered inputs. Our model allows us to solve realistic instances of the problem to optimality in a very short time, and produces better schedules than both scheduling with constant draft, and manual scheduling approaches used in practice at ports.

Introduction

Ship scheduling deals with assigning arrival and departure times to a fleet of ships, as well as the amount and sometimes type of cargo that is carried on each ship. Ship scheduling is a problem with significant real-world impact, as the majority of the world's international trade is transported by sea, so even a small improvement in schedule efficiency can have significant benefits to industry (Christiansen, Fagerholt, and Ronen 2004).

One consideration in ship scheduling which does not occur in other transportation problems is that most ports have restrictions on the draft of ships that may safely enter the port. Draft is the distance between the waterline and the ship's keel, and is a function of the amount of cargo loaded onto the ship. Ships with a deep draft risk running aground when entering or leaving the port; therefore most ports restrict the draft of ships allowed to transit through the port. In existing ship scheduling algorithms, only constant maximum draft constraints have been considered (Christiansen et al. 2011) (Song and Furman 2010).

In practice, most ports restrict ship sailing drafts using safety rules that estimate the under-keel clearance (UKC) – the depth of water under a ship's keel. These safety rules are dependent on environmental conditions such as tide, and therefore in practice, the allowable sailing draft at most ports varies with time. Existing ship scheduling algorithms do not consider time-varying draft restrictions at ports, and therefore may produce suboptimal schedules, where ships sail with less cargo than they could have carried if the schedule had allowed for the extra draft allowable at high tide.

Our work aims to develop ship scheduling algorithms that can take environmentally-dependent time-varying draft constraints into account. In this paper, we consider the problem of scheduling ships with time-varying drafts to maximise cargo throughput over a single high tide, at a single bulk export port. In practice, our models can be used to schedule ships for any time range, including multiple tides; however, a problem with multiple tides would likely be solvable by a decomposition approach, as discussed in future work.

We present Constraint Programming and Mixed Integer Programming models for the deterministic single-port ship scheduling problem with time-varying environmentally-dependent draft constraints. These models have been implemented in the MiniZinc constrained optimisation modelling language, and solved using the G12 finite domain CP solver and MIP OSI CBC solver (Nethercote et al. 2007). The problem without tug constraints, as described in Section , has been implemented in a prototype application, and has undergone user testing at Port Hedland, Australia's largest iron ore export port (Kelareva 2011). We then compare the performance of the CP and MIP models, and investigate variations on our CP model. The final two sections present our conclusions and discuss future work.

Draft and Under-Keel Clearance

Draft is the distance between the waterline and the ship's keel, and most ports have safety restrictions on the draft of ships allowed to transit through the channel to reduce the risk of deep-draft ships running aground in shallow water. At draft-restricted ports, accurate modelling of draft constraints allows more cargo to be loaded onto each ship in good environmental conditions without compromising safety, which increases profit for shipping companies. In practice, draft constraints at ports are usually calculated by estimating the under-keel clearance of a ship – the amount of water under the ship's keel.

Under-keel clearance rules vary between ports, but may include the following components (O'Brien 2002):

- the *depth* of water at each point along the channel.
- the predicted *tide height* at the time the ship will be transiting through the channel.
- the *draft* of the ship.
- *squat* – a phenomenon caused by the Bernoulli effect, which causes a ship travelling faster in shallow water to sit lower in the water.
- *heel* – the effect of a ship leaning to one side under the effect of centripetal force due to turning, or due to the force of wind. Heel causes one side of the ship to sit lower in the water, thus decreasing under-keel clearance.
- *wave response* – the vertical component of a ship's motion in response to waves.

Equation (1) shows an example under-keel clearance constraint for a port. A vessel v will be allowed sail at time t if the constraint expressed in Equation (1) is met, ie. if the sum of the positive UKC factors (depth D and tide height T), minus the sum of the negative UKC factors (draft d, squat s, heel h, wave response w, etc) exceeds some safety factor F.

$$D(t) + T(t) - d(v) - s(v,t) - h(v,t) - w(v,t) \geq F \quad (1)$$

The simplest UKC requirement for ship scheduling is that schedules must satisfy safety rules at the port. However, scheduling approaches that use overly simple approximations to under-keel clearance and draft constraints may miss the opportunity to load a ship to a deeper draft at high tide, or when waves are low, leading to suboptimal schedules.

Constraint Programming Model

Key Features

There are several key features of the ship schedule optimisation problem which make it computationally difficult.

Optimality: At a draft-constrained port, such as Port Hedland in Western Australia, even a single centimetre of extra draft can allow 130 extra tonnes of cargo to be carried on an average-sized bulk carrier (Port Hedland Port Authority 2011b). At a large bulk export port, with around 1300 bulk carriers departing from the port in a year (Port Hedland Port Authority 2011a), even a 1cm increase in the average draft would result in up to 170,000 tonnes of extra cargo. Even a small reduction in schedule quality is undesirable, so in this paper we always aim to find an optimal schedule.

Oversubscribed Problems: The problem is undersubscribed in general, as all ships can sail on the tide, but it is oversubscribed at the peak of the high tide, since not all ships will be able to carry the maximum amount of cargo.

Time-indexed Formulation: The allowable draft changes rapidly with the tide. We therefore use a time-indexed formulation with five-minute time increments, to model the draft with sufficient accuracy. This high time resolution results in large variable domain sizes, and therefore longer solution times.

Sequence-Dependent Constraints: Both the basic constraints, and the constraints on availability of tugs (discussed in Section) depend on the order in which ships sail. Constraints are thus highly interdependent, which makes it hard to find optimal solutions.

Variables and Parameters

The CP model of our problem involves the following variables and parameters.

Parameters

V is the set of all ships to be scheduled.

$[1, T_{max}]$ is the range of allowable time indices.

$E(v)$ is the earliest time when vessel v can sail.

$ST(v_i, v_j)$ defines the minimum separation time required between ships $v_i, v_j \in V$.

$D(v,t)$ defines the maximum allowable sailing draft for the vessel v at time slot t, accounting for all safety rules at this port, including the effects of tide, waves, squat, etc. The maximum draft may also be limited by the ship structure, or by the amount of cargo available for transport.

B is the number of pairs of incoming and outgoing ships which share the same berth as their origin and destination.

$B_i(b)$ and $B_o(b)$ are the incoming and outgoing ships in pair b which share a common berth.

$d(b)$ is the minimum allowable delay (in time slot increments) between the sailing time for the outgoing ship $B_o(b)$ departing from its berth, and the sailing time for the incoming ship $B_i(b)$ due to arrive at the same berth.

$C(v)$ is the tonnage per centimetre of draft for vessel v.

Explanation

Each ship $v \in V$ – incoming or outgoing – has an associated berth. For each pair of ships there is a minimum separation $ST(v_i, v_j)$ between their sailing slots, which depends on their associated berths. If v_i is an incoming ship whose berth is the same as that of outgoing ship v_o, in addition to their minimum separation, v_i can only be scheduled after v_o. If v_o is not included in the schedule, then nor can v_i be.

Decision variables

$s(v) \in [0,1]$ specifies whether the ship v is included in the schedule, since it is possible that some ships cannot be scheduled at all, if, for example, a scheduler tries to schedule too many ships on one tide.

$T(v) \in [1, T_{max}]$ is the time slot when vessel v is scheduled to sail.

Constraints

Earliest Departure Time Constraints

$$s(v) = 1 \Rightarrow T(v) \geq E(v), \ \ \forall\, v \in V \quad (2)$$

Every vessel v that is included in the schedule ($s(v) = 1$) sails no earlier than its earliest possible departure time $E(v)$.

Berth Availability Constraints

$$\begin{gathered} s(B_i(b)) = 1 \Rightarrow \\ s(B_o(b)) = 1 \wedge T(B_o(b)) \leq T(B_i(b)) - d(b), \\ \forall\, b \in B \end{gathered} \quad (3)$$

For every pair of incoming and outgoing ships $B_i(b)$ and $B_o(b)$ that share the same berth as their destination and origin, if the incoming ship is included in the schedule, then the outgoing ship must also be included in the schedule, and the scheduled sailing time $T(B_i(b))$ of the incoming ship must be later than the sailing time $T(B_o(b))$ of the outgoing ship by a delay of at least $d(b)$, which gives the outgoing ship enough time to clear the berth.

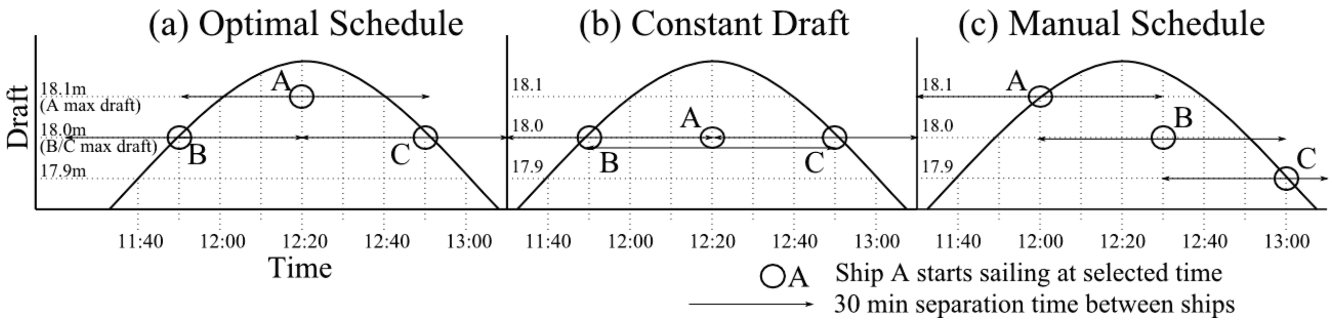

Figure 1: Optimal schedule vs. constant draft and manual schedule.

Separation Time Constraints

$$s(v_i) = 1 \ \wedge s(v_j) = 1 \Rightarrow \qquad (4)$$
$$T(v_j) - T(v_i) \geq ST(v_i, v_j) \vee$$
$$T(v_i) - T(v_j) \geq ST(v_j, v_i),$$
$$\forall \ v_i, \ v_j \in V$$

Each pair of ships v_i, v_j has a minimum separation time $ST(v_i, v_j)$ or $ST(v_j, v_i)$ between their scheduled sailing times, depending on the order in which they sail.

Objective Function

The objective function for the ship scheduling problem at a single port varies per port. Some ports may have an objective function that purely optimises throughput; other ports may need to prioritise fairness to competing clients above optimising the total throughput for the port. In this paper, we only consider a simple generic objective function that optimises total throughput at the port, shown by equation (5).

$$\sum_{v \in V} s(v) \ . \ C(v) \ . \ D(v, T(v)) \qquad (5)$$

Equation (5) shows an objective function that optimises the total cargo throughput at the port by maximising the sum of the drafts $D(v, T(v))$ at the scheduled sailing time $T(v)$ for each vessel v, weighted by the tonnage per centimetre of draft $C(v)$ for each ship, since the amount of extra cargo allowed by an increase in draft varies depending on the size and shape of the ship. The objective function is also weighted by the binary variable $s(v)$ that specifies whether the ship was included in the schedule.

Comparison Against Existing Approaches

Constant Draft: In existing ship scheduling problems with multiple ports such as (Fisher and Rosenwein 1989), only constant draft restrictions are considered. Constant draft restrictions produce good schedules for problems with small (ie non-draft-restricted) ships, or for non-tidal ports. However, the majority of the world's sea ports are affected by tides, which will cause the draft restrictions at the port to vary with time. For ports that have time-varying draft restrictions, scheduling draft-limited ships using constant draft restrictions can result in sub-optimal schedules.

Figure 1 shows one example of a schedule with three ships sailing on a tide, with time-varying or constant draft restrictions. This problem has three outgoing ships A to C, with maximum drafts of 18.1m for ship A, and 18.0m for ships B and C. For simplicity, we assume that the separation time between each pair of ships is 30 minutes, regardless of order.

Time-varying draft restrictions will allow ship A to sail with 18.1m draft, as shown in Figure 1(a). Constant draft restrictions, on the other hand, require the draft restriction to be set low enough to allow all ships to sail with the constant draft. This results in all ships sailing with 18.0m draft, as shown in Figure 1(b).

Manual Scheduling: Schedules produced by our time-varying draft model can be compared against schedules produced by naive manual optimisation algorithms. Two algorithms that can be used for manual scheduling are: schedule the ship with the highest objective function component first – ie. the ship with the highest tonnage per cm draft – or schedule the ship with the highest allowable draft first. In both cases, each ship is scheduled at the earliest time it can sail with its highest possible draft.

Both these algorithms produce suboptimal schedules for some problems, such as the example shown in Figure 1. This example is simpler than a real-world scenario; however, the drafts, amounts of cargo, and sailing windows for the maximum drafts used in this example are realistic. Constraints at real ports are more complex, and less likely to be solved to optimality by these simple manual algorithms. For the larger, more realistic example problems presented in Section (ONEWAY_NARROW problems without tugs) the optimal schedules allowed an average of 120cm more draft for the set of ships compared to scheduling with constant draft constraints, and 15.8cm more draft per set of ships compared with manually scheduling the biggest ships first.

Impact of Suboptimal Schedules: Assuming that all ships can carry 130 tonnes of cargo per centimetre of draft – the average for iron ore bulk carriers at Port Hedland (Port Hedland Port Authority 2011b) – the fixed-draft and manual schedules above result in 1300 tonnes less iron ore being carried on these three ships. This is around US$221,000 less iron ore, at the January – October 2011 average iron ore price of around US$170/tonne (Index Mundi 2011).

Around 1300 ships sailed from Port Hedland in the 2009-10 financial year (Port Hedland Port Authority 2011a). A 10cm reduction in draft of every 3rd ship will result in 563Kt less iron ore being shipped on the same set of ships over a year, or around US$96 million less iron ore per year.

Constant draft restrictions may reduce draft even more for some ships, as the height of each tide varies with the springneap tidal cycle by up to several metres. Any scheduling approach which uses constant draft restrictions would have to set the draft restriction low enough to allow ships to sail even at neap tides, thus reducing the draft of ships sailing at higher tides even further.

This shows that problems involving large draft-restricted ships sailing through tidally-affected ports have a high potential for improvement in schedule quality by incorporating time-varying draft restrictions. As shown by this example, even a small difference from the optimal schedule has a high cost, which makes this problem worth solving to optimality even for a single port.

Tug Constraints

Many ports require tugs – small boats – to guide large cargo ships in and out of the port. Tugs attach to outgoing ships at berth, and detach from the ship after it clears the most constrained part of the channel. Tugs attach to incoming ships while the ship is at sea, and detach as the ship arrives at berth. At some ports, tugs may also need to push incoming ships onto the berth. Tug availability may constrain ship schedules, as found in user testing of a prototype of our CP model at Port Hedland (Kelareva 2011).

Modelling Approaches

Tug constraints depend on the number of tugs available at the port, the number of tugs required for each ship, and how long the tugs are in use for on each job. Tug job durations depend on the origin and destination of the ship, the tug's travel time between jobs, whether the tug is required to assist in berthing, and port operational rules, such as the locations where tugs attach to and detach from ships. Tug job durations are therefore highly sequence dependent.

There has been some prior research on tug scheduling, such as (Yan et al. 2009). However, in our problem, tug availability only needs to be considered as a constraint in the larger ship schedule optimisation problem.

Our first attempt at modelling tug constraints assigned individual tugs to ships. However, this was too slow to find optimal solutions, since the large number of interactions caused by sequence-dependent waiting times between ships were compounded by the highly sequence-dependent waiting times between successive tug jobs.

Our second tug model only tracked the number of tugs busy at each point in time, rather than allocating tugs to ships. However, as the delay between successive jobs for a tug depends on the sequence of jobs it performs, this new model still required tracking origins and destinations for tugs, and was still too slow to find an optimal solution.

Successful CP Model

Our third attempt at modelling tug constraints successfully used features of the problem to simplify the constraints, enabling realistic-sized ship schedules to be solved to optimality within a few minutes.

There are two problem features, or simplifying assumptions, that were critical to simplifying the tug constraints.

1. The port we considered in our model has a single channel, which can only be used in one direction at a time.
2. The berths are close enough together that the travel time for tugs moving from a berth to sea or vice versa can be considered independent of the berth location. This is the case at Port Hedland, and is likely to apply at the majority of ports. However, at ports where berths are spaced far apart compared to the length of the channel, a model based on this assumption would need extension to avoid reducing schedule quality.

Assumption 1 implies that a schedule of ships can be split up into component scenarios of four possible types:

1. A sequence of outgoing ships
2. A sequence of incoming ships
3. An outgoing ship followed by an incoming ship
4. An incoming ship followed by an outgoing ship

The tug availability constraints can be considered separately for each scenario, thus making the combined ship schedule optimisation problem much simpler.

Scenarios 1 and 2: For a sequence of outgoing ships or a sequence of incoming ships, the turnaround time between successive jobs for any tug is independent of berth location (Assumption 2), and therefore also of the sequence of jobs, as long as both jobs are in the same direction.

Scenario 3: At the port we modelled in our problem, the tugs transfer from the outgoing ship to the incoming ship while the ships are in transit, so no additional delay is required. For ports where there is a delay for tug transfer from an outgoing ship to an incoming ship, Scenario 3 can be modelled similarly to Scenario 4 below.

Scenario 4: Tugs moving from an incoming ship followed by an outgoing ship require a delay between the end of the first job and the start of the second job. This can be modelled by introducing an extra variable for outgoing ships, to specify how many tugs are still busy prior to that ship's departure time due to having recently completed an incoming job. The duration of the extra delay for tugs moving from an incoming to an outgoing ship may vary based on the locations of the destination and origin berths.

Variables and Parameters

Adding tug availability constraints to the CP model requires the following additional parameters and variables.

Parameters

U_{max} is the total number of tugs available at the port.

$G(v)$ is the number of tug groups required for vessel v, where a tug group is a set of tugs that spend the same length of time working on that ship.

$H(v, g)$ is the number of tugs in group g for vessel v.

G_{max} is the maximum number of tug groups for any ship.

I is the set of incoming ships.

O is the set of outgoing ships.

$r(v, g)$ (the "turnaround time") specifies the time taken for the tugs in group g of vessel v to become available for another job in the same direction (incoming vs outgoing).

$X(v_i, v_j)$ specifies the extra delay required for tugs moving from an incoming vessel to an outgoing vessel, compared to the usual maximum turnaround time $\max_{g \in G} r(v_i, g)$. In this paper, $X(v_i, v_j)$ is 0 for tugs moving from an outgoing ship to an incoming ship; however, this may not be the case for other ports.

Dependent variables
$U(v,t,g)$ is the number of tugs busy for tug group g of vessel v at time t, assuming the next job for these tugs is in the same direction (incoming/outgoing).

$x(v,t)$ defines the number of extra tugs that are busy at time t for an outgoing vessel v, due to still being in transit from the destination of an earlier incoming job.

$L(v,t)$ is an "overlap" flag. $L(v,t)$ is true iff vessel v has its extra tug delay time $x(v,t)$ overlapping with the transit start time for another vessel travelling in the opposite direction.

Constraints

Scenarios 1 and 2: One-Directional Sequence of Ships

$$s(v) = 1 \wedge t \geq T(v) \wedge t < T(v) + r(v,g) \Rightarrow U(v,t,g) = H(v,g), \quad \forall\, v \in V,\ t \in [1, T_{max}],\ g \in [1, G_{max}] \tag{6}$$

$$s(v) = 0 \vee t < T(v) \vee t \geq T(v) + r(v,g) \Rightarrow U(v,t,g) = 0, \quad \forall\, v \in V,\ t \in [1, T_{max}],\ g \in [1, G_{max}] \tag{7}$$

For a one-directional sequence of ships, at each time t, the number of tugs busy $U(v,t,g)$ in group g of vessel v is equal to the total number of tugs in that tug group, $H(v,g)$, if and only if the vessel has already sailed at time t, but the turnaround time $r(v,g)$ has not yet passed.

Scenario 4: Incoming Followed By Outgoing

$$L(v_i,t) \iff \exists\, v_o \in O \ \ s.t. \ \ t = T(v_o) \wedge T(v_i) \leq T(v_o) \wedge T(v_i) + \max_{g \in [1, G(v_i)]} r(v_i,g) + X(v_i,v_o) > T(v_o), \quad \forall\, v_i \in I,\ t \in [1, T_{max}] \tag{8}$$

$$x(v_i,t) = \text{bool2int}(L(v_i,t)). \sum_{g \in [1, G(v_i)]} H(v_i,g), \quad \forall\, v_i \in I,\ t \in [1, T_{max}] \tag{9}$$

The constraints in Equation (8) specify that the "overlap" flag, $L(v_i,t)$ is true iff the extra tug delay time $X(v_i,v_o)$ overlaps with the scheduled sailing time $T(v_o)$ of at least one incoming vessel, v_o.

The constraints in Equation (9) express the requirement that for any incoming vessel v_i, the tugs from that vessel are still considered busy for vessels travelling in the opposite direction if the "overlap" flag $L(v_i,t)$ is true.

Scenario 3: Outgoing Followed By Incoming

$$x(v_o,t) = 0, \ \ \forall\, v_o \in O,\ t \in [1, T_{max}] \tag{10}$$

For our port, there is no additional delay required for tugs to transfer from an outgoing ship to an incoming ship. For ports where this is not the case, the additional delay for tugs to transfer from an outgoing ship to an incoming ship can be modelled similarly to the Scenario 4 constraints above.

Tug Availability Constraints

$$\sum_{v \in I} \sum_{g \in G(v)} U(v,t,g) \leq U_{max}, \ \ \forall\, t \in [1, T_{max}] \tag{11}$$

$$\sum_{v_o \in O} \sum_{g \in G(v_o)} U(v_o,t,g) + \sum_{v_i \in I} x(v_i,t) \leq U_{max}, \quad \forall\, t \in [1, T_{max}] \tag{12}$$

At each time t, the total number of tugs in use, $U(v,t,g)$ over all tug groups g, for all incoming vessels $v \in I$ is no greater than the total number of tugs available at the port, U_{max}. Equation (11) ensures that the schedule satisfies the tug availability constraints for all sequences of incoming vessels (Scenario 2).

Equation (12) represents the same requirement for outgoing vessels – Scenario 1. However, the total number of busy tugs also needs to include any tugs that were still busy at time t due to having recently completed an incoming job and not yet having had time to transfer to the outgoing ship $(X(v_i).(t = T(v))$ – Scenario 4.

Mixed Integer Programming Model

We modelled the ship scheduling problem with time-varying draft as a Mixed Integer Programming model, as MIP has been effective for solving other maritime scheduling problems (Christiansen, Fagerholt, and Ronen 2004). Our MIP model is similar to the CP model, but with non-linear constraints converted to linear forms.

Variables and Parameters

The MIP model uses the same parameters as the CP model presented in Section , but adds some new variables.

$s(v,t) \in [0,1]$ is a binary variable which specifies whether the ship v is scheduled to sail at time t.

$T(v) \in [0, T_{max}]$ is a dependent variable specifying the time slot when vessel v is scheduled to sail:

$$T(v) = \sum_{t \in [1, T_{max}]} s(v,t)\ .\ t, \ \ \forall\, v \in V \tag{13}$$

Constraints

Some CP constraints need to be converted to linear form for the MIP model. Modified constraints are shown below.

Ship Uniqueness Constraints

$$\sum_{t \in [1, T_{max}]} s(v,t) \leq 1, \ \ \forall\, v \in V \tag{14}$$

Earliest Departure Time Constraints

$$T(v) \geq E(v), \ \ \forall\, v \in V \tag{15}$$

Berth Availability Constraints

$$T(B_o(b)) \leq T(B_i(b)) - d(b) \tag{16}$$

$$\sum_{t \in [1, T_{max}]} s(B_o(b),t) \geq \sum_{t \in [1, T_{max}]} s(B_i(b),t), \ \ \forall\, b \in B$$

Separation Time Constraints

$$s(v_i,t) + \sum_{t' \in [t, \min(T_{max}, t+ST(v_i,v_j)-1)]} s(v_j,t') \leq 1 \quad (17)$$

$$s(v_j,t) + \sum_{t' \in [t, \min(T_{max}, t+ST(v_j,v_i)-1)]} s(v_i,t') \leq 1$$

$$\forall\ v_i,\ v_j \in V,\ t \in [1, T_{max}]$$

Scenarios 1 and 2: One-Directional Sequence of Ships

$$U(v,t,g) = H(v,g). \sum_{t' \in [\min(1, t-r(v,g)+1), t]} s(v,t') \quad (18)$$

$$\forall\ v \in V,\ t \in [1, T_{max}],\ g \in [1, G_{max}]$$

Scenario 4: Incoming Followed By Outgoing

$$x(v_i,t) = L(v_i,t). \sum_{g \in [1, G(v)]} H(v_i,g) \quad (19)$$

$$\forall\ v_i \in I,\ t \in [1, T_{max}]$$

$$L(v_i,t) \geq s(v_o,t) + \sum_{t' \in [\max(1, t-t_{range}), t]]} s(v_i,t) - 1 \quad (20)$$

$$\forall\ v_i \in I,\ v_o \in O,\ t \in [1, T_{max}], \text{where}$$

$$t_{range} = t - X(v_i,v_o) - \max_{g \in [1, G(v_i)]} (r(v_i,g)) + 1$$

Experimental Results

The models described in Sections , and were formulated in MiniZinc 1.4, and solved with the finite domain CP solver and MIP OSI CBC solver included in G12 (Nethercote et al. 2007) (Nethercote et al. 2010). The G12 finite domain CP solver uses standard backtracking search, and allows a choice of variable selection and domain reduction strategies to be used for solving the problem. For MIP, the search strategy is set by the solver. The choice of solver was constrained by commercial requirements, as this model was going to be used in a commercial system.

The CP calculation time is highly dependent on the search strategy used by the solver. We analysed the effectiveness of several variable selection and domain reduction strategies, such as searching on time vs draft first. In this paper, we use the fastest search strategy for all CP model comparisons, searching first on the dependent variable – draft, $D(v, T(v))$ – and searching on time as a second step.

Problem Instances

The CP and MIP models were tested on four different problem types that varied in how tightly constrained they were.

1. ONEWAY_NARROW (ON): all ships sail in the same direction (outbound), and have high maximum drafts, leading to narrow windows at the peak of the tide, and the problem being oversubscribed at high tide.
2. MIXED_NARROW (MN): ships are split evenly between inbound and outbound, and outbound ships have high maximum drafts with narrow peak draft windows.
3. ONEWAY_WIDE (OW): all ships are outbound, but with lower maximum drafts, leading to wider windows and a less constrained schedule.

Problem	CP	MIP
MW	**10 (3.17)**	8 (39.1)
OW	**10 (0.45)**	9 (41.8)
MN	**10 (7.99)**	8 (11.4)
ON	**8 (42.5)**	7 (11.4)
MWT	**10 (115)**	6 (180)
OWT	**8 (1.76)**	8 (273)
MNT	**8 (4.06)**	6 (126)
ONT	**6 (10.3)**	5 (7.66)

Table 1: Comparison of MIP vs CP.

4. MIXED_WIDE (MW): ships are evenly split between inbound and outbound, and with low maximum drafts and wide sailing windows.

WIDE problems are less constrained than NARROW problems, and MIXED problems are less constrained than ONEWAY problems, though MIXED problems also result in more complex tug constraints coming into effect.

Each problem type was solved with 4–10 ships sailing on a single tide. These are realistic sized problems – Port Hedland, Australia's biggest iron ore port, set a record of five draft-constrained ships sailing on a single tide in 2009 (OMC International 2009). The problems are based on a fictional but realistic port, similar to the ship_scheduling data set used for the 2011 MiniZinc challenge (University of Melbourne 2011). Each set of problems was solved both with and without tugs (eg. MWT vs MW respectively).

Computational Results

All tests were run on a Windows 7 machine with an Intel i7-930 quad-core 2.80 GHz processor and 12.0 GB RAM, and with a 5-minute (300-second) cutoff time.

Table 1 presents the largest number of ships for each problem type that could be solved to optimality with each model within the 300-second cutoff time. (Suboptimal heuristics are left for future work, so we do not present the best solution obtained for problems where an optimal solution was not found.) Numbers in brackets indicate the time taken to find the optimal solution for the largest problem where an optimal solution was found. Bold font indicates the model that was fastest to solve the largest problem. The 300-second cutoff time was chosen to allow schedulers time to try different inputs, and to allow rescheduling in response to delays, equipment breakdowns and weather.

Discussion

Tugs vs No Tugs

The results in Table 1 show that, as expected, the addition of tug constraints significantly increases the calculation time required to solve the problems.

ONEWAY problems are more severely affected, probably because ONEWAY problems are more tightly constrained than MIXED problems, due to the incoming ships in the MIXED problems having low draft and therefore wide sailing windows. This results in tug constraints causing more disruption to ONEWAY problems than to MIXED problems.

CP vs MIP

Table 1 shows that CP with a good choice of search strategy was able to solve larger problem sizes to optimality within the cutoff time for almost all problem types, and was the fastest to solve all problems. MIP was particularly slow for MIXED problems, possibly indicating that the tug constraints for incoming ships followed by an outgoing ships, which occur only for MIXED problem types, were particularly inefficient in the MIP model.

One possible reason for CP being faster than MIP for this set of problems is that searching on draft allows large areas of the search space to be eliminated quickly by the solver. The search strategy chosen by the MIP solver is ignorant of this aspect of the problem structure.

While our MIP model resulted in slower solution times than CP, the use of MIP for this scheduling problem may be worth investigating further. There may be ways to improve the MIP constraints to make them more efficient, and other MIP solvers may also be faster at solving this problem. Further investigation of the MIP model is left for future work.

We briefly explored other solvers (Gecode, CPLEX and Gurobi). Though calculation times varied, the overall picture stayed the same, namely that CP was significantly faster.

Improving the CP Model

After the initial investigation of the CP and MIP model calculation times, we also experimented with modifying the model itself to make it faster to solve.

We implemented a modified Constraint Programming model for the ship scheduling problem with additional variables specifying the ordering between every pair of vessels in the schedule, to investigate whether setting the order in which ships sail prior to choosing the exact sailing times would reduce the search, and thus speed up the time required to find an optimal solution. This approach produced little improvement on its own, but was much more effective when combined with other improvements.

One variation to the CP model which made the sequence variables significantly faster to solve was to convert multi-dimensional array lookups with a variable index to use one-dimensional arrays instead. The objective function uses the term $D(v, T(v))$, where v is a constant and $T(v)$ is a variable. Constraints on variables which index into arrays are inefficiently handled in MiniZinc; a better model is achieved by replacing $D(v, T(v))$ with $D'_v(T(v))$ where D'_v is the projection of D on v.

Another modification that improved calculation time was sorting ships into ascending order of maximum objective function component $\max_{t \in [1..T_{max}]} D(v,t).C(v)$, where $D(v,t)$ is the maximum allowable draft for vessel v at time t, and $C(v)$ is the tonnage per centimetre of draft for vessel v. This improved the efficiency of the search slightly, as it allowed sailing times to be searched in order of ship size.

Sequence Variable Model

Adding sequence variables to the CP model required the following modifications to variables and constraints.

Dependent variables

$sb(v_i, v_j) \in [0, 1]$ – SailsBefore(v_i, v_j) – is a binary variable which is set to 1 if the vessel v_i sails earlier than the vessel v_j, ie. if $T(v_i) < T(v_j)$, and 0 otherwise.

Problem	OLD	SEQVARS	ONEDIM	SORT	ONEDIMSORT
MW	10 (3.17)	10 (2.09)	**10 (0.22)**	10 (1.00)	10 (0.56)
OW	10 (0.45)	10 (2.09)	10 (2.62)	10 (0.56)	**10 (0.34)**
MN	10 (7.99)	9 (298)	**10 (0.22)**	10 (8.30)	10 (0.56)
ON	8 (42.5)	8 (57.9)	10 (190)	8 (37.1)	**10 (157)**
MWT	10 (115)	**10 (14.1)**	10 (14.3)	10 (124)	10 (19.4)
OWT	8 (1.76)	8 (14.4)	9 (124)	8 (2.09)	**9 (100)**
MNT	8 (4.06)	8 (137)	10 (166)	8 (4.16)	**10 (121)**
ONT	6 (10.3)	6(13.0)	9 (228)	6 (7.75)	**9 (211)**

Table 2: Comparison of modified CP models.

Sequence Variable Constraints

$$sb(v_i, v_j) = 1 \leftrightarrow T(v_i) < T(v_j), \forall v_i, v_j \in V; v_i \neq v_j \quad (21)$$
$$sb(v_i, v_j) = 1 \leftrightarrow sb(v_j, v_i) = 0, \forall v_i, v_j \in V; v_i \neq v_j \quad (22)$$
$$sb(v_i, v_i) = 0, \forall v_i \in V \quad (23)$$

The constraints in Equations (21), (22) and (23) define the values of the sequence variables $sb(v_i, v_j)$ introduced above.

Equation (21) specifies that the vessel v_i "sails before" v_j if the scheduled sailing time $T(v_i)$ for v_i is earlier than the scheduled sailing time $T(v_j)$ for v_j.

Equation (22) specifies that if vessel v_i "sails before" v_j, then v_j cannot sail before v_i, and Equation (23) specifies that no vessel can sail before itself.

Separation Time Constraints

$$\begin{aligned} s(v_i) = 1 \ \wedge s(v_j) = 1 \Rightarrow \quad (24) \\ (sb(v_i, v_j) = 1 \Rightarrow T(v_j) - T(v_i) \geq ST(v_i, v_j)), \\ \forall \ v_i, \ v_j \in V \end{aligned}$$

The separation time constraints originally introduced in Section are modified to depend on the "sails before" sequence variables $sb(v_i, v_j)$. The modified constraints above represent the requirement that for each pair of ships v_i, v_j, with v_i sailing first, v_i's sailing time must predate v_j's by at least the minimum separation time $ST(v_j, v_i)$.

Calculation Results

The modified CP models were compared against the model introduced in Sections and , with the same set of example problems as used for earlier tests. The original CP model was tested with the fastest search strategy – searching on draft first, followed by time. The improved CP models were tested with the search strategies that were fastest for each model, as discussed in Section below.

Table 2 shows that sequence variables on their own were not effective in speeding up calculation time for most problems. However, the CP model with one-dimensional arrays performed significantly better than the basic CP model for all problem types, solving problems with an average of 2 more ships than the original CP model. The CP model with sorted ships only gave a very small improvement on the basic CP model, and in some cases resulted in slower solution time. However, when combined with one-dimensional arrays, sorted inputs had faster calculation times for the most difficult problems – ONEWAYNARROW with and without tugs, and MIXEDNARROW and ONEWAYWIDE with tugs.

	OneDimSort		OneDim		Sort	
Problem	Draft	SeqVars	Draft	SeqVars	Draft	SeqVars
MW	**10 (0.45)**	10 (0.56)	**10 (0.22)**	**10 (0.22)**	**10 (1.00)**	8 (179)
OW	**10 (0.33)**	**10 (0.34)**	**10 (2.40)**	10 (2.62)	**10 (0.56)**	8 (136)
MN	**10 (0.33)**	10 (0.56)	**10 (0.22)**	**10 (0.22)**	**10 (8.30)**	8 (54.5)
ON	8 (22.8)	**10 (157)**	8 (18.4)	**10 (190)**	**8 (37.1)**	9 (204)
MWT	10 (50.7)	**10 (19.4)**	10 (48.8)	**10 (14.3)**	**10 (124)**	7 (9.83)
OWT	8 (0.89)	**9 (100)**	8 (1.43)	**9 (124)**	**8 (2.09)**	8 (257)
MNT	9 (40.1)	**10 (121)**	9 (27.8)	**10 (166)**	**8 (4.16)**	7 (21.9)
ONT	7 (119)	**9 (211)**	7 (98.3)	**9 (228)**	**6 (7.75)**	8 (282)

Table 3: Searching on draft vs sequence variables for three modified CP models.

Search Strategies

Table 3 compares searching on draft or sequence variables first for the modified models. Bold font indicates the fastest search strategy for each model. The calculation times for the improved CP model with one-dimensional arrays (with or without sorted inputs) were faster when searching on sequence variables compared to searching on draft. However, for the CP model with sorted inputs only, searching on sequence variables is much slower than searching on draft. This is also the case for the basic CP model with sequence variables only. This implies that sequence variable constraints in particular propagate better when expressed with one-dimensional rather than multi-dimensional arrays.

Discussion

Improvements to the model significantly improved calculation time. Converting constraints on variables which index into multi-dimensional arrays to use one-dimensional arrays instead led to the largest improvements, particularly when combined with sequence variable search.

Whereas other improvements presented in this paper were highly problem-dependent, conversion of multi-dimensional arrays to more efficient one-dimensional arrays could be built into a CP solver or modelling language. This issue is worth considering in the design of CP solvers, and may be worth investigating as a potential improvement to MiniZinc.

Conclusions

In this paper, we presented CP and MIP models for the problem of scheduling ships at a port with time-varying draft constraints. We compared these models against both fixed-draft schedules of the sort produced by existing ship scheduling algorithms, and against manual scheduling approaches used in practice at ports. Our models produced schedules that allowed more cargo throughput for some problems than the fixed-draft and manual scheduling approaches, and were able to solve problems of realistic size.

Our CP and MIP models included constraints on the availability of tugs, which were highly sequence dependent and made the problem computationally difficult. We were able to solve this problem by splitting the tug constraints into several scenarios which could be handled separately.

We found that the CP model with a good choice of search strategy was significantly faster and was able to solve larger problems than the MIP model. We also compared several variations on our original CP model, and investigated their effects on solution time. We found that converting CP constraints that used multi-dimensional array lookups with a variable index to use one-dimensional arrays significantly improved calculation time, particularly when searching first on additional sequence variables specifying the order in which ships sail. Sorting the input data prior to passing it into the CP model also improved calculation time slightly.

Future Work

As this is the first paper considering a novel problem in maritime logistics, there are several avenues for future research.

Other CP solvers may achieve faster calculation times for this problem. In the 2011 MiniZinc Challenge, the Chuffed and Gecode solvers achieved the fastest performance on the closely related ship_scheduling problem (University of Melbourne 2011). However, it would be worth investigating a wider array of solvers, including some that incorporate recent advancements in global scheduling constraints, such as constraints on optional interval variables (Laborie and Rogerie 2008) and reservoir resource constraints (Laborie 2003).

In this paper, we only looked at optimising throughput on a single tide. The multi-tide scheduling problem would likely be suited to a Logic-Based Benders Decomposition approach, similar to the manual scheduling approach used in practice at some ports, where ships are first allocated to tides, and then the schedule for each tide is optimised. This is similar to other allocation and scheduling problems where Logic-Based Benders Decomposition has proved effective (Hooker 2007) (Bajestani and Beck 2011).

Larger ship routing or mining supply chain optimisation problems may also benefit from time-varying draft constraints. As shown in this paper, finding optimal schedules with time-varying draft is a non-trivial problem even for a single port. Multi-port problems may be solvable with a decomposition approach, or using heuristic search.

Finally, this problem involves uncertainty due to variation in environmental conditions, loading delays and equipment breakdowns. Uncertainty may be a rich area for future work.

Acknowledgements

The authors would like to acknowledge the support of OMC International, where Elena Kelareva was employed while conducting this research. The authors would also like to acknowledge the support of ANU and NICTA at which Elena Kelareva is a PhD student. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

Bajestani, M. A., and Beck, C. 2011. Scheduling an aircraft repair shop. In *Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling (ICAPS'11)*, 10–17.

Christiansen, M.; Fagerholt, K.; Flatberg, T.; Haugen, Ø.; Kloster, O.; and Lunda, E. 2011. Maritime inventory routing with multiple products: A case study from the cement industry. *Euro. J. of Operational Research* 208(1):86–94.

Christiansen, M.; Fagerholt, K.; and Ronen, D. 2004. Ship routing and scheduling: status and perspectives. *Transportation Science* 38(1):1–18.

Fisher, M., and Rosenwein, M. 1989. An interactive optimization system for bulk-cargo ship scheduling. *Naval Research Logistics* 36(1):27–42.

Hooker, J. N. 2007. Planning and scheduling by logic-based benders decomposition. *Operations Research* 55:588–602.

Index Mundi. 2011. Iron ore monthly price. http://www.indexmundi.com/commodities/?commodity=iron-ore.

Kelareva, E. 2011. The "DUKC Optimiser" ship scheduling system. In *2011 International Conference on Automated Planning and Scheduling System Demonstrations*.

Laborie, P., and Rogerie, J. 2008. Reasoning with conditional time-intervals. In *Proceedings of the 21st International FLAIRS Conference*.

Laborie, P. 2003. Algorithms for propagating resource constraints in ai planning and scheduling: Existing approaches and new results. *Artificial Intelligence* 143(2):151–188.

Nethercote, N.; Stuckey, P.; Becket, R.; Brand, S.; Duck, G.; and Tack, G. 2007. Minizinc: Towards a standard cp modelling language. In Bessière, C., ed., *Principles and Practice of Constraint Programming - CP 2007*, volume 4741 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 529–543.

Nethercote, N.; Marriott, K.; Rafeh, R.; Wallace, M.; and de la Banda, M. 2010. Specification of zinc and minizinc.

O'Brien, T. 2002. Experience using dynamic underkeel clearance systems. In *Proceedings of the PIANC 30th International Navigational Congress*, 1793–1804.

OMC International. 2009. Dukc helps port hedland set ship loading record. http://www.omc-international.com/images/stories/press/omc-20090810-news-in-wa.pdf.

Port Hedland Port Authority. 2011a. 2009/10 cargo statistics and port information. http://www.phpa.com.au/docs/CargoStatisticsReport.pdf.

Port Hedland Port Authority. 2011b. Dynamic under keel clearance system. http://www.phpa.com.au/dukc_information.asp.

Song, J.-H., and Furman, K. 2010. A maritime inventory routing problem: Practical approach. *Computers & Operations Research*.

University of Melbourne. 2011. Minizinc challenge 2011. http://www.g12.cs.mu.oz.au/minizinc/challenge2011/challenge.html.

Yan, W.; Bian, Z.; Chang, D.; and Huang, Y. 2009. An improved particle swarm optimization algorithm for solving dynamic tugboat scheduling problem. In *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, 433 – 437.