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

Elena Kelareva¹
Sebastian Brand²
Philip Kilby¹
Sylvie Thiébaux¹
Mark Wallace³

1. NICTA / ANU
2. NICTA / University of Melbourne
3. NICTA / Monash University

Photograph: Marj Kibby
Outline

Problem Introduction

- Draft Restrictions in Ship Scheduling
- Maximising Cargo Throughput at a Bulk Export Port

Modelling the Problem

- Constraint Programming Model
- Tug Constraints
- CP vs MIP

Improving the CP Model

- Sequence Variables
- 2D arrays
- Sorted Inputs
Ship Scheduling Background

What is ship scheduling?

- Choose sailing times and amounts of cargo
- Maximise profit / minimise cost
- Obey port safety rules (eg. Draft)
Time-Varying Draft Restrictions

- Ports have safety restrictions on draft
- Draft increases as more cargo loaded
- Existing approaches use constant draft restrictions
- Time-varying draft → increased accuracy → more cargo → more profit

Can sail if:
Tide + Depth − Draft − Squat − Heel − WR ≥ Safety Margin

[O'Brien 2002]
Maximising Cargo Throughput at a Bulk Export Port

Outbound ships full; inbound empty
Narrow channel $\rightarrow$ sequence-dependent separation times
Objective: maximise cargo for a set of ships
## Commercial System: Input Ship Parameters

<table>
<thead>
<tr>
<th>Vessel Name</th>
<th>CAMELLIA 50m 300m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gmf (m)</td>
<td></td>
</tr>
<tr>
<td>KG (m)</td>
<td></td>
</tr>
<tr>
<td>Earliest Departure</td>
<td>25May2012 0908</td>
</tr>
<tr>
<td>Requested Draft (m)</td>
<td>19</td>
</tr>
<tr>
<td>Min Draft (m)</td>
<td>18</td>
</tr>
<tr>
<td>Max Draft (m)</td>
<td>19</td>
</tr>
<tr>
<td>Lock Draft</td>
<td></td>
</tr>
<tr>
<td>Lock Time</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>3 EARLY</td>
</tr>
<tr>
<td>Sail</td>
<td>NPB</td>
</tr>
<tr>
<td>Berth</td>
<td>B2</td>
</tr>
<tr>
<td>Destination</td>
<td>SEA</td>
</tr>
<tr>
<td>Pilotage</td>
<td></td>
</tr>
<tr>
<td>Tugs</td>
<td></td>
</tr>
<tr>
<td>NTugs Early Turnaround</td>
<td></td>
</tr>
</tbody>
</table>
## Commercial System: Output

<table>
<thead>
<tr>
<th>Vessel</th>
<th>Sailing Draft</th>
<th>Sailing Slot Open Time</th>
<th>Sailing Slot Close Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) AMIRA (32.26m, 217m)</td>
<td>16.00m</td>
<td>25May2012 0910</td>
<td>25May2012 0925</td>
</tr>
<tr>
<td>(B) BEGONIA (45m, 280.8m)</td>
<td>18.00m</td>
<td>25May2012 1235</td>
<td>25May2012 1250</td>
</tr>
<tr>
<td>(C) CAMELIA (50m, 300m)</td>
<td>18.75m</td>
<td>25May2012 1145</td>
<td>25May2012 1200</td>
</tr>
<tr>
<td>(D) DIONE (45m, 283m)</td>
<td>18.00m</td>
<td>25May2012 1020</td>
<td>25May2012 1035</td>
</tr>
<tr>
<td>(E) EURYDICE D (42.5m, 267.6m)</td>
<td>18.00m</td>
<td>25May2012 1050</td>
<td>25May2012 1105</td>
</tr>
<tr>
<td>(F) FIRST JUPITER (45m, 277m)</td>
<td>17.45m</td>
<td>25May2012 1255</td>
<td>25May2012 1310</td>
</tr>
</tbody>
</table>
Optimality

- 1cm extra draft = 130t extra cargo
- Even a small reduction in quality is undesirable
- Aim to find an optimal schedule

Oversubscribed problem

- Undersubscribed in general; oversubscribed at high tide

Time-indexed formulation

- Draft changes over time
- Modelled with 5-minute time increments

Sequence-dependent constraints

- Separation times between ships depend on ship order
- Tug constraints depend on ship order
Basic CP Model

Earliest Departure Time:
\[ s(v) = 1 \rightarrow T(v) \geq E(v) \quad \forall v \in V \]

Berth Availability:
\[ s(B_i(b)) = 1 \rightarrow \\
\left( s(B_o(b)) = 1 \land T(B_o(b)) \leq T(B_i(b)) - d(b) \right) \forall b \in B \]

Separation Time:
\[ s(v_i) = 1 \land s(v_j) = 1 \rightarrow \\
T(v_j) - T(v_i) \geq ST(v_i, v_j) \lor T(v_i) - T(v_j) \geq ST(v_j, v_i) \quad \forall v_i, v_j \in V \]

Objective Function:
maximise \[ \sum_{v \in V} s(v) \cdot C(v) \cdot D(v, T(v)) \]
Comparison Against Existing Approaches

Constant draft restrictions and manual schedules

- 10cm less total draft
- 1300 tonnes less cargo
- $221,000 less iron ore

Larger problems in paper

- 120cm more draft per set of ships than constant draft
- 15.8cm more draft than manual scheduling
Sequence-dependent turnaround times
- Tug constraints scale badly

Allocating tugs to ships: too slow!

Counting busy tugs
- First ensure feasibility, then assign tugs
- Faster than allocating tugs
- Turnaround times are still sequence-dependent
- Still need to track tug origins and destinations: slow
Tug Constraints: Faster Model

2 Key Assumptions:

- Ships cannot pass each other in the channel
- Berths are close enough together that travel time from berth to sea or vice versa is independent of berth location

Incoming and outgoing tugs can be tracked separately
Tug Constraints: 4 Scenarios

Scenario 1: outgoing ships
- out
- out
- out

Scenario 2: incoming ships
- in
- in

Scenario 4: incoming followed by outgoing

Scenario 3: outgoing followed by incoming
Constraint Programming Model: Tugs

Scenarios 1 and 2: One-Directional Sequence of Ships

\[ \begin{align*}
    s(v) &= 1 \land t \geq T(v) \land t < T(v) + r(v, g) \rightarrow \\
    U(v, t, g) &= H(v, g), \forall v \in V, t \in [1, T_{\text{max}}], g \in [1, G_{\text{max}}] \\
    s(v) &= 0 \lor t < T(v) \lor t \geq T(v) + r(v, g) \rightarrow \\
    U(v, t, g) &= 0, \forall v \in V, t \in [1, T_{\text{max}}], g \in [1, G_{\text{max}}]
\end{align*} \]

Scenario 4: Incoming Followed by Outgoing

\[ \begin{align*}
    &L(v_i, t) \Leftrightarrow \exists v_o \in O \text{ s.t.} \\
    &t = T(v_o) \land T(v_o) \leq T(v_i) \land T(v_i) + \max(r(v_i, g), g \in [1, G(v_i)]) + X(v_i, v_o) > T(v_o), \\
    &\forall v_i \in I, t \in [1, T_{\text{max}}] \\
    &x(v_i, t) = \text{bool2int}(L(v_i, t)). \sum_{g \in [1, G(v_i)]} H(v_i, g), \forall v_i \in I, t \in [1, T_{\text{max}}]
\end{align*} \]

Scenario 3: Outgoing Followed by Incoming

\[ \begin{align*}
    &x(v_o, t) = 0, \forall v_o \in O, t \in [1, T_{\text{max}}]
\end{align*} \]

Tug Availability Constraints

\[ \begin{align*}
    &\sum_{v \in I} \sum_{g \in G(v)} U(v, t, g) \leq U_{\text{max}}, \forall t \in [1, T_{\text{max}}] \\
    &\sum_{v_o \in O} \sum_{g \in G(v_o)} U(v_o, t, g) + \sum_{v_i \in I} X(v_i, t) \leq U_{\text{max}}, \forall t \in [1, T_{\text{max}}]
\end{align*} \]
Mixed Integer Programming Model

Time Slot Definition:
\[ T(v) = \sum_{t \in [1,T_{\text{max}}]} s(v, t) \cdot t \quad \forall v \in V \]

Ship Uniqueness Constraints:
\[ \sum_{t \in [1,T_{\text{max}}]} s(v, t) \leq 1 \quad \forall v \in V \]

Earliest Departure Time:
\[ T(v) \geq E(v) \quad \forall v \in V \]

Berth Availability:
\[ T(B_o(b)) \leq T(B_i(b)) - d(b) \land \sum_{t \in [1,T_{\text{max}}]} s(B_o(b), t) \geq \sum_{t \in [1,T_{\text{max}}]} s(B_i(b), t) \quad \forall b \in B \]

Separation Time:
\[ s(v_i, t) + \sum_{t' \in [t, \min(T_{\text{max}}, t + ST(v_i, v_j))] \ldots} s(v_j, t') \leq 1 \land s(v_j, t) + \sum_{t' \in [t, \min(T_{\text{max}}, t + ST(v_j, v_i))] \ldots} s(v_i, t') \leq 1 \]
\[ \forall v_i, v_j \in V, t \in [1,T_{\text{max}}] \]
Scenarios 1 and 2: One-Directional Sequence of Ships

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

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

Scenario 4: Incoming Followed by Outgoing

\[ x(v_i, t) = L(v, t) \sum_{g \in [1, G(v_i)]} H(v_i, g), \forall v_i \in I, t \in [1, T_{\text{max}}] \]

\[ L(v_i, t) \geq s(v_o, t) + \sum_{t' \in [\max(1, t-t_{\text{range}}), t]} s(v_i, t) - 1, \forall v_i \in I, v_o \in O, t \in [1, T_{\text{max}}] \]

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

Scenario 3: Outgoing Followed by Incoming (Identical to CP model)

Tug Availability Constraints (Identical to CP model)
Computational Results

Model implemented in MiniZinc
- G12 finite domain CP solver
- OSI CBC MIP solver

Problems of varying size and constrainedness
- Problem sizes: 4-10 ships (realistic for bulk export port)
- “one-way” vs “mixed” ship direction
- “narrow” vs “wide” windows at maximum draft
  - Schedule is more or less constrained at peak of tide
- 5-minute cutoff time

Search strategies
- MIP: solver chooses search strategy
Computational Results

Largest problem solved within 5 min cutoff (calc time in brackets)

<table>
<thead>
<tr>
<th>Problem</th>
<th>CP</th>
<th>MIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW</td>
<td>10 (3.17)</td>
<td>8 (39.1)</td>
</tr>
<tr>
<td>OW</td>
<td>10 (0.45)</td>
<td>9 (41.8)</td>
</tr>
<tr>
<td>MN</td>
<td>10 (7.99)</td>
<td>8 (11.4)</td>
</tr>
<tr>
<td>ON</td>
<td>8 (42.5)</td>
<td>7 (11.4)</td>
</tr>
<tr>
<td>MWT</td>
<td>10 (115)</td>
<td>6 (180)</td>
</tr>
<tr>
<td>OWT</td>
<td>8 (1.76)</td>
<td>8 (273)</td>
</tr>
<tr>
<td>MNT</td>
<td>8 (4.06)</td>
<td>6 (126)</td>
</tr>
<tr>
<td>ONT</td>
<td>6 (10.3)</td>
<td>5 (7.66)</td>
</tr>
</tbody>
</table>

Table 1: Comparison of MIP vs CP.

CP faster than MIP
MIP may do better with other solvers or search strategies
Sequence variables
  • Additional variables specifying ordering between ships
  • Search on sequence variables first

Converting multidimensional array lookups to 1D arrays
  • Objective: \( \Sigma D(v, T(v)) \)
  • Replace with \( D'_v (T(v)) \) where \( D'_v \) is the projection of \( D \) on \( v \)
  • More efficiently implemented in G12 FD solver

Sorting ships for improved search efficiency
  • Order by decreasing max draft
  • Assign times to biggest ships first
OneDim gives a larger improvement than Sort
SeqVars alone: slightly slower in general
1D array conversion could be automated: MiniZinc update?
CP Model Improvements: Search Strategies

- Original CP model: draft >> time
- Sequence variables alone / sorted inputs: draft > seq. vars
- With 1D Arrays: seq. vars > draft
- Seq. var. constraints propagate better with 1D arrays?
Conclusions

Introduced time-varying draft to ship scheduling
  • Novel problem: optimising cargo throughput at bulk export port
  • More cargo than fixed-draft and manual approaches
  • Tug constraints solved by splitting into scenarios

CP faster than MIP

3 improvements to CP model
  • Sequence variables
  • 1D arrays
  • Sorted inputs

1D arrays may have implications for other models and MiniZinc improvements
Future Work

Other CP solvers (Chuffed, Gecode, others)

Extension to larger ship routing or mining supply chain problems
  • Multiple ports
  • Longer time horizon
  • Decomposition approach may be needed

MIP model may do better with other solvers/search strategies
Thank you!