

The “DUKC Optimiser” Ship Scheduling System

Elena Kelareva*

OMC International / ANU / NICTA
Melbourne, Australia
elena@omc-international.com.au

Abstract

We present an automated ship scheduling system – DUKC[®] Optimiser – which selects sailing times for a set of cargo ships at a port, so as to maximise total cargo throughput while meeting port operational and safety guidelines, as well as producing schedules that are fair to all companies using the port. The system has been developed by maritime engineering company OMC International, incorporating elements of the author’s PhD research at the Australian National University and NICTA. A prototype of the system has undergone user testing in late 2010, and is planned to undergo further development in order to include additional functionality and incorporate results into a web-based ship management system.

DUKC[®] Optimiser is the first ship scheduling system that accounts for environmentally-dependent constraints on the times when ships can enter or leave a port. The system uses OMC’s existing Dynamic Under-Keel Clearance (DUKC[®]) software to calculate sailing windows for each ship. The results of the DUKC[®] calculations are then converted into a Mixed-Integer Programming model, formulated in the MiniZinc modelling language, and solved using the G12 constraint optimisation solver.

Ship Scheduling Background

Ship scheduling deals with assigning sailing times to a fleet of ships, as well as optionally the amount and type of cargo that each ship carries. 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 is that most ports have restrictions on the draft of ships that are able to 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.

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In existing ship scheduling algorithms, draft constraints have only been considered in trivial ways, for example, assuming that a given port will always allow ships with a draft of 13 metres or less, and never allow ships with deeper drafts to enter (Fisher and Rosenwein 1989). Other ship scheduling algorithms leave draft constraints entirely up to human schedulers (Fagerholt 2004).

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. In recent years, OMC International has developed algorithms to accurately calculate under-keel clearance using real-time environmental conditions. OMC’s Dynamic Under-Keel Clearance (DUKC[®]) software allows significantly more cargo to be loaded safely onto each vessel compared to the static UKC rules previously used by most ports, which don’t take real-time environmental data into account (OMC 2011). However, ship scheduling has not been able to take advantage of these recent improvements in UKC estimation, due to not considering complex time-varying draft constraints.

In this presentation, we demonstrate the DUKC[®] Optimiser software, which is the first ship scheduling system that can take environmentally-dependent time-varying draft constraints into account.

Dynamic Under-Keel Clearance

Figure 1 illustrates all aspects of ship motion taken into account by the Dynamic Under-Keel Clearance (DUKC[®]) software in calculating under-keel clearance. Components of ship motion taken into account by the DUKC[®] software include:

Draft: the distance from the waterline to the bottom of the ship’s keel.

Squat: a phenomenon caused by the Bernoulli effect which causes a ship travelling fast through shallow water to sink deeper into the water than a ship travelling slowly.

Heel: the effect of a ship leaning towards one side, caused by the centripetal force of turning, or the force of wind on the side of the ship.

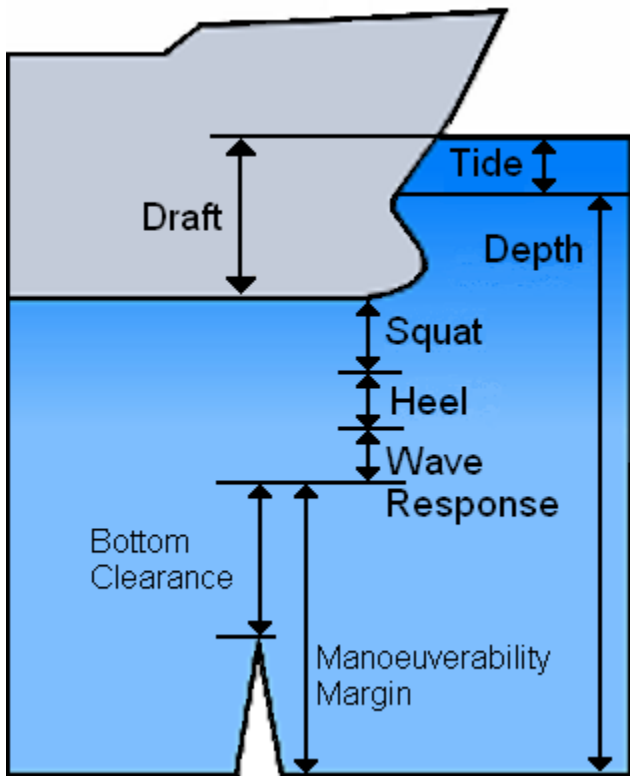


Figure 1: Dynamic Under-Keel Clearance Components

Wave Response: the motion resulting from the action of waves on the ship. Only the vertical component of this motion affects under-keel clearance.

Under-keel clearance is computed as follows:

$$\text{UKC} = \text{Tide} + \text{Depth} - \text{Draft} - \text{Squat} - \text{Heel} - \text{Wave Response}$$

The Bottom Clearance and Manoeuvrability Margin shown in Figure 1 are safety factors that ensure the ship has sufficient distance from the highest points on the channel bottom (Bottom Clearance) and that there is sufficient water around the ship to maintain good manoeuvrability (Manoeuvrability Margin). If the under-keel clearance is below either the Bottom Clearance or Manoeuvrability Margin safety limits, then the DUKC[®] software will advise the operator not to sail. However, the DUKC[®] software only provides navigational advice; the final decision always rests with the ship's pilot or captain.

For a more detailed analysis of Dynamic Under-Keel Clearance methodology, see (O'Brien 2002).

System Architecture

The initial prototype of DUKC[®] Optimiser is a command line application which uses Microsoft Excel input and output files as a simple "GUI". Excel was used in place of a customised GUI in order to deliver a prototype for user test-

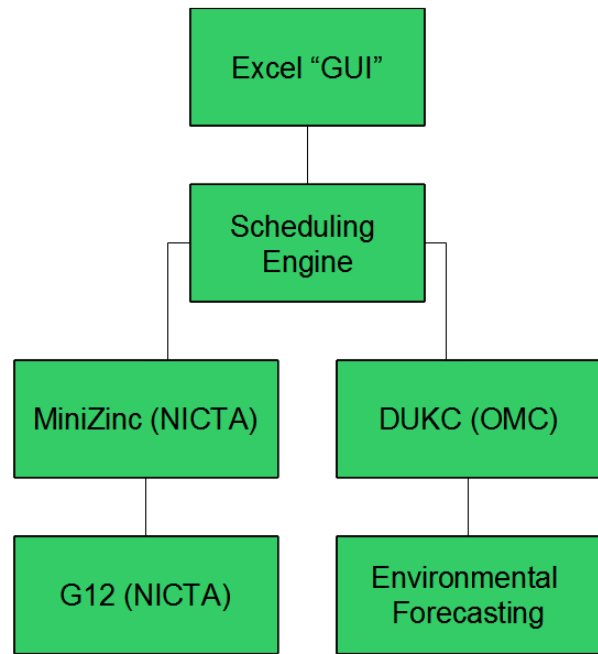


Figure 2: DUKC[®] Optimiser System Architecture

ing as quickly as possible; the system is planned to be incorporated into a web-based under-keel clearance management system in future development. See Figure 3 for a mockup of what a future GUI may look like.

The scheduler inputs data about each ship into an Excel spreadsheet, which is then read by DUKC[®] Optimiser and converted into a set of queries to OMC's DUKC[®] software. The DUKC[®] software reads real-time environmental forecasts and measurements from databases, and uses these to analyse each ship's motion in response to the predicted tide, wave and current conditions. The results of this analysis is used to calculate under-keel clearance – the amount of water under the ship at each point in the transit – and thus to determine sailing windows for a range of drafts for each ship.

DUKC[®] Optimiser then converts the user inputs and the results of the DUKC[®] calculations into a Mixed Integer Programming (MIP) model, implemented in the MiniZinc optimisation language (Nethercote et al. 2007). This model is then solved using the G12 constraint optimisation platform (Stuckey et al. 2005).

MIP Formalisation

The MIP formalisation of the ship scheduling problem includes constraints on the valid range of drafts for each ship, as well as on the sailing draft allowed at each time for each ship by the port's safety rules. Each ship has a minimum and maximum draft range, determined by physical limitations such as the size and shape of the ship, as well as an

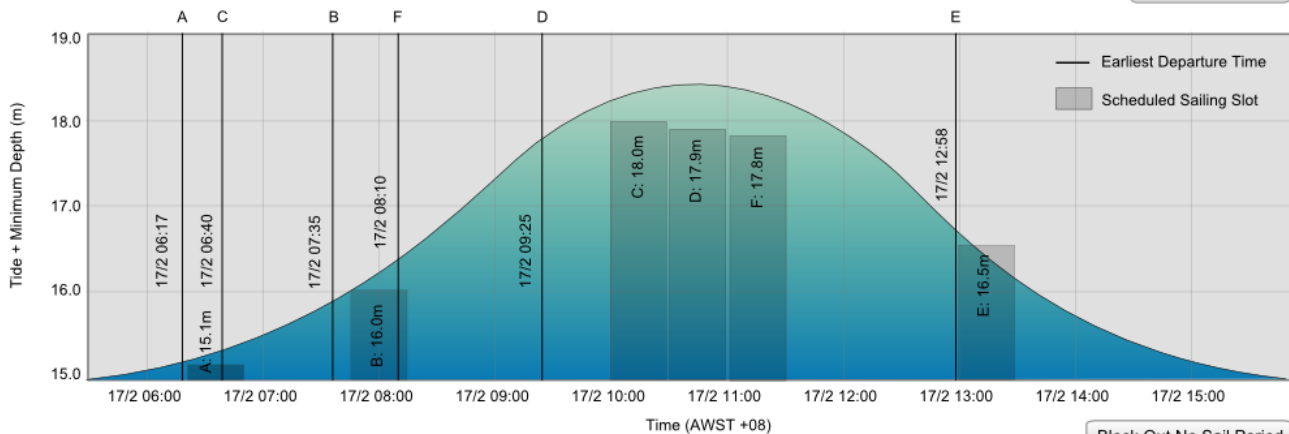
DUKC Optimiser - Schedule for High Water 17/2 10:51

High Water: 2011-02-17 10:51:00 +08

Tugs Available: 10 (maximum)

Vessel Name	Beam	Length	Priority	Request Draft	Sailing Draft	Origin	Destination	Sail Early	GMf	No. of Tugs
A	45	240	5	15.1	15.1	Berth1	Sea	True		3
B	45	260	1	16	16	Berth2	Sea	True		3
C	50	300	2	18.3	18.0	Berth3	Sea	True		4
D	50	305	3	17.9	17.9	Berth4	Sea	True		4
E	45	240	6	16.5	16.5	Berth5	Sea	False		3
F	50	290	4	18	17.8	Berth6	Sea	True		4

Add More Vessels



Block Out No Sail Period

Cancel

Save Inputs

Calculate

Figure 3: Example GUI: Output

earliest sailing time, which depends on when the ship finishes loading.

The model also incorporates other parameters representing the port's geometry and operational procedures, including locations of significant shallow or narrow points along the channel, travel times between waypoints, and minimum required separation times between ships passing a given waypoint. Constraints on these parameters ensure that ships do not pass through each other, and stay far enough apart to meet the port's safety guidelines.

Objective Function

The objective function for the ship scheduling problem 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.

One example of an objective function optimises the total cargo throughput at the port by maximising the sum of the drafts, weighted by the tonnage per centimetre of draft, since

the amount of extra cargo allowed by an increase in draft varies depending on the size and shape of the ship.

An alternative objective function, for a port with more complex operating procedures, allows shipping agents to request minimum drafts for each ship, for example to meet contractual obligations. Ships are allocated sailing slots based on priority, which is determined by the port's fairness rules.

Future Development

Future development of the system will include incorporating additional resource constraints to account for tugs, which are used to assist ships entering or leaving the port. Initial user testing conducted in late 2010 found that in some situations, the need to wait for tugs to return from a job constrains the schedule. Therefore tugs need to be incorporated before the system can be used in operation.

Another future development will be to incorporate the system into a web-based under-keel clearance management system, to improve the usability of DUKC[®] Optimiser for

operational use. In the demonstration, we will use mockup screenshots from the future GUI to demonstrate system behaviour, even though the GUI itself has not yet been developed.

A mockup GUI showing input data and an output schedule for a set of six ships sailing on one tide is shown in figure 3.

Acknowledgements

The DUKC[®] Optimiser system was developed as a project at OMC International. The author would like to acknowledge the contribution of other OMC engineers involved in the development of the DUKC[®] Optimiser system, particularly Giles Lesser, Gregory Hibbert and Kalvin Ananda. The author would also like to acknowledge the input from her PhD supervisors Philip Kilby, Sylvie Thiébaux and Mark Wallace, and the support of ANU and NICTA at which she 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

- Christiansen, M.; Fagerholt, K.; and Ronen, D. 2004. Ship routing and scheduling: status and perspectives. *Transportation Science* 38(1):1–18.
- Fagerholt, K. 2004. A computer-based decision support system for vessel fleet scheduling - experience and future research. *Decision Support Systems* 37(1):35–47.
- Fisher, M., and Rosenwein, M. 1989. An interactive optimization system for bulk-cargo ship scheduling. *Naval Research Logistics* 36(1):27–42.
- 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.
- O’Brien, T. 2002. Experience using dynamic underkeel clearance systems. In *Proceedings of the PIANC 30th International Navigational Congress*, 1793–1804.
2011. OMC International public website. <http://www.omc-international.com.au>.
- Stuckey, P.; de la Banda, M.; Maher, M.; Marriott, K.; Slaney, J.; Somogyi, Z.; Wallace, M.; and Walsh, T. 2005. The g12 project: Mapping solver independent models to efficient solutions. In Gabbrielli, M., and Gupta, G., eds., *Logic Programming*, volume 3668 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 9–13.