AN IMPROVED RECEDING HORIZON GENETIC ALGORITHM FOR THE TUG FLEET OPTIMISATION PROBLEM

Robin T. Bye and Hans G. Schaathun
Faculty of Engineering and Natural Sciences
Aalesund University College
Postboks 1517, NO-6025 Ålesund, Norway
Email: {roby,hasc}@hials.no

KEYWORDS

Dynamic resource allocation; genetic algorithm; cost functions; receding horizon control; tug fleet optimisation simulator.

ABSTRACT

A fleet of tugs along the northern Norwegian coast must be dynamically positioned to minimise the risk of oil tanker drifting accidents. We have previously presented a receding horizon genetic algorithm (RHGA) for solving this tug fleet optimisation (TFO) problem. In this paper, we begin by presenting an overview of the TFO problem and the details of the RHGA. Next, we identify and correct a flaw in the original cost function of the RHGA. In addition, we present several new cost functions that can be used for dynamic resource allocation by an algorithm such as the RHGA. In a preliminary simulation study, we correct and extend the simulation scenarios used in our previous work and examine the merit of each of the suggested cost functions. Finally, we discuss the potential for an objective evaluation method for comparing various TFO algorithms and briefly present our TFO simulator.

INTRODUCTION

Thousands of ships, including several hundred oil tankers, move along the northern Norwegian coastline every year, making it susceptible to the risk of drift grounding accidents and oil spill (Havforskningsinstituttet, 2010). Constantly attempting to reduce the risk of such accidents, the Norwegian Coastal Administration (NCA) runs a vessel traffic services (VTS) centre in the town of Vardø that administers a fleet of tugs patrolling the coastline. The role of the VTS centre is to continuously order the tugs to new positions in a manner such that if an oil tanker loses manoeuvrability, e.g., through steering or propulsion failure, there will be at least one tug sufficiently close that it can intercept the drifting oil tanker before it runs ashore (Eide et al., 2007a).

To aid the NCA with positioning their fleet of tugs, a set of risk-based decision support tools based on dynamical risk models have been developed (Eide et al., 2007a,b). The risk models are based on real-time information such as wind, waves, currents, geography, kind of oil tankers in transit, their crew, and the estimated oil spill size and potential impact, to mention some. Using the decision support tools aids the human operator at a VTS centre in directing tugs towards high-risk target areas.

The abovementioned decision support tools do not tell explicitly which tugs should move where; that is still an informed decision based on the operators experience and currently available information. Since the number of oil tanker transits are expected to rise significantly in coming years (Havforskningsinstituttet, 2010), the problem of positioning the tugs can quickly grow and become unmanageable for human operators. Consequently, there is a need of an algorithm able to calculate position trajectories that each tug should follow in order to reduce the overall risk of drifting accidents.

In our Dynamic Resource Allocation with Maritime Application (DRAMA) research group at the Aalesund University College (AAUC), we have solved this tug fleet optimisation (TFO) problem by means of a receding horizon genetic algorithm (RHGA) (Bye et al., 2010; Bye, 2012). This algorithm combines methods from control theory and computational intelligence to iteratively plan movement trajectories for each individual tug such that the net collective behaviour of the tugs as measured by a cost function is optimised by means of a genetic algorithm (GA). Subsequently, at last year's meeting of this conference, we presented a modified version of the RHGA called the receding horizon mixed integer programming algorithm (RHMIPA) in which we reformulated our choice of cost function such that it turned into a linear programming problem (Assimizele et al., 2013). Notably, the cost function is the same in the RHGA and the RHMIPA, it is only the mathematical formulation that differs. Whereas the RHGA typically will return a good, albeit inexact and suboptimal solution, at every run, the RHMIPA in contrast finds an exact, global minimum of the cost function.

Since both algorithms are identical except for the method used for minisiming the very same cost function, a simple measure for comparison is simply the accumulated cost, for which the exact MIP solver in the RHMIPA will cause it to outperform the RHGA, which uses a suboptimal, heuristic GA solver. Unsurprisingly, perhaps, this superiority of the RHMIPA comes at the cost of slower computational evaluation when compared to the heuristic RHGA (Assimizele et al., 2013).

In keeping with the real-world nature of the TFO problem, both algorithms were also compared with a realistic

Proceedings 28th European Conference on Modelling and Simulation ©ECMS Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani (Editors)
ISBN: 978-0-9564944-8-1 / ISBN: 978-0-9564944-9-8 (CD)

option that the NCA and Norwegian policymakers regularly consider, namely that of a static policy in which the tugs are uniformly positioned at base stations spread out along the coastline. Obviously, in terms of cost function minimisation, the active tug fleet patrol scheme of the RHGA and the RHMIPA both outperform the static policy in which tugs are waiting passively for an incident to occur (Bye et al., 2010; Bye, 2012; Assimizele et al., 2013).

The work presented in this paper was motivated by the desire to (1) continue our development of the RHGA with a particular focus on optional cost functions, and (2) rewrite and formalise the implementation of the algorithm and simulation scenarios in a TFO simulator.

With respect to (1), we recently identified what appears to be a flaw in the cost function used for the RHGA and RHMIPA. In addition to rectifying this error, we wanted to examine several other choices of cost functions. A particular challenge, then, is the comparison and evaluation of cost functions. Using a naive heuristic or a simple method such as the static policy above constitute an indirect measure for comparing algorithms, where each algorithm's performance versus the naive heuristic or static policy is compared instead of a direct comparison of the algorithms' ability to minimise some common cost function. However, when designing new algorithms for the TFO problem that employ new and different cost functions, the methods for comparing algorithms above are of limited value, since different cost functions by definition are not directly comparable. Hence, there is a need for some kind of common, objective method for evaluating the merit of a particular choice of cost function in a TFO algorithm.

With respect to (2), our previous work has shown us that the TFO problem is an excellent case study for the DRAMA research group, with still many aspects of the TFO problem yet to be examined. Specifically, we want to investigate how a TFO algorithm can be able to handle a variety of simulation scenarios, including oil tankers entering or leaving the patrol zone; changes in number of tugs and oil tankers; changing weather conditions, drift trajectories, and maximum tug speeds; and much more (see Discussion). To answer these questions, we have completely rewritten our code base using the advanced, purely-functional programming language Haskell.[1]. Part of the motivation for choosing a functional language like Haskell was to enable fast prototyping while keeping our code robust, concise, and not the least correct. Another reason was the potential for extensions into parallel programming, which may be required as the simulator grows more complex and more computational resources are needed.

In the following sections, we proceed by presenting the formulation of the TFO problem as defined in Bye et al. (2010); Bye (2012); Assimizele et al. (2013). Next, we point out what we believe is a flaw in the original cost function employed in the RHGA, and present a number of optional cost functions. We then propose a new and objective evaluation method that can be used for comparing various TFO algorithms. Finally, we present some simulation results and discuss the viability of our approach as well as future work.

TFO PROBLEM FORMULATION

For formulating the TFO problem, we adopt most of the assumption in our previous work (Bye et al., 2010; Bye, 2012; Assimizele et al., 2013). First, assume that N_o oil tankers move in one dimension only (north or south, say) along a line of motion z. This is a reasonable assumption considering that oil tankers by law follow predefined piecewise-linear corridors. Second, inside of z and closer to shore, assume that N_p tugs are patrolling along a line of motion y parallel to z. Although the coastline is rather rugged, with fjords, peninsulas, and islands, tugs should stop drifting ships before they reach land or danger zones, thus a straight patrol line some distance from the rugged coastline can be considered a conservative choice.

Next, we assume real-time access to simulation data from a set of accurate models able to predict future positions of oil tankers along z and the corresponding potential drift trajectories given current and predicted information about the tankers themselves and the environment they are operating in. Such models exist and are currently an active focus of research (e.g., see Hackett et al. (2006); Breivik and Allen (2008); Breivik et al. (2011)).

For example, consider an oil tanker currently positioned at $z(t)$. There is a small chance that the tanker may suffer from engine failure or some other incident and start drifting right now at $t = t_\mathrm{d}$. However, if not, it may also continue sailing along z. We may predict the future positions of the tanker some time T_h hours ahead in time, where T_h is called the prediction horizon. Employing a discrete-time model with a sampling period of $T_\mathrm{s} = 1$ hour, the estimated future tanker positions are given by $\{\hat{z}(t|t_\mathrm{d})\}$ for $t = t_\mathrm{d}+1, t_\mathrm{d}+2, \ldots, t_\mathrm{d}+T_\mathrm{h}$.

For each predicted point $\hat{z}(t|t_\mathrm{d})$, there is a corresponding predicted drift trajectory starting at $\hat{z}(t|t_\mathrm{d})$ that may or may not intersect the patrol line y after an estimated drift time $\hat{\Delta}$ into the future depending on ocean currents, wave heights, wind conditions, oil tanker shape and weight, and other factors.

In previous work, we either set the estimated drift time $\hat{\Delta}$ to be 8 hours for all oil tankers (Assimizele et al., 2013), or to be drawn randomly for each oil tanker from a uniform probability distribution in the interval $[8, \ldots, 12]$ hours (Bye et al., 2010; Bye, 2012). According to Eide et al. (2007a), these drift times correspond to situations of "fast drift" and not the typical, or average, case. On the other hand, it should be kept in mind that there will inevitably be a delay between when an oil tanker begins drifting and when the VTS centre actually is notified of the incident and can order tugs to the rescue.

Collecting all predicted drift trajectories for all oil tankers results in a distribution of *crosspoints* located at points where future drift trajectories will intersect the patrol line y. A crosspoint of the cth oil tanker's drift trajectory at time t can be defined as y_t^c. Taking the drift

[1]http://www.haskell.org

time $\hat{\Delta}$ into account, a drift trajectory starting on z at $t = t_{\mathrm{d}}$ will have a a cross point on y at $t = t_{\mathrm{d}} + \hat{\Delta}$. Assuming the same drift time for all drift trajectories and considering the prediction horizon T_{h}, there is a predicted set of crosspoints given by

$$\{y_t^c\} = \left\{y_{t_{\mathrm{d}}+\hat{\Delta}}^c, y_{t_{\mathrm{d}}+1+\hat{\Delta}}^c, \ldots, y_{t_{\mathrm{d}}+T_{\mathrm{h}}}^c\right\} \quad (1)$$

In addition to crosspoints, we define a *patrol point* (tug position on y) on the pth tug's patrol trajectory at time t as y_t^p.

Based on the predicted future distribution of crosspoints, the TFO problem is to calculate trajectories, or sequences of patrol points, along y for each of the patrolling tugs such that the risk of an oil tanker in drift not being reached and prevented from grounding is minimised.

Figure 1 shows a graphical representation of the problem description, exemplified by two patrolling tugs and three oil tankers.

COST FUNCTIONS

Original Cost Function f_1

Determining a suitable cost function for optimisation algorithms such as the RHGA and the RHMIPA is imperative for the algorithm to be able to find desirable solutions. The cost function we present firstly is the same as the one used in these algorithms (Bye et al., 2010; Bye, 2012; Assimizele et al., 2013) and is defined as the sum of the distances between all crosspoints and the *nearest* patrol points. The rationale behind this choice is that if an oil tanker in drift can/cannot be saved by a tug some distance away, it is not important that other tugs further away can/cannot save it at a later time.

For N_{o} oil tankers and N_{p} patrol tugs, the cost $f_1(t)$ is defined mathematically as

$$f_1(t) = \sum_{t=t_{\mathrm{d}}}^{t_{\mathrm{d}}+T_{\mathrm{h}}} \sum_{o \in O} \min_{p \in P} |y_t^c - y_t^p| \quad (2)$$

for each oil tanker $o \in O = \{1, 2 \ldots, N_{\mathrm{o}}\}$ and each patrol tug $p \in P = \{1, 2 \ldots, N_{\mathrm{p}}\}$.

Note that choosing distance as a cost measure is equivalent to minimum rescue time if one assumes that all tugs have the same maximum speed. For cases where tugs have different maximum speeds, one could define rescue time as distance divided by maximum tug speed and add up the minimum rescue times for each cross point.

An example scenario with six oil tankers and three tugs is shown in Figure 2 (adapted from (Assimizele et al., 2013)), where an optimal solution found by the RHMIPA (bottom) is compared with a static policy (top) where tugs simply remain at their individual base station. Employing the RHMIPA, the patrol tugs spread out and track different clusters of crosspoints, thus collectively reducing the overall risk of grounding.

Cost Function f_2

The cost function f_1 presented above adds up the absolute value of the distance between every cross point and its nearest patrol point. This means that in situations where a particular patrol point lies between two crosspoints, its exact position does not affect the cost, since being closer to one of the crosspoints means being further away from the other. This may or may not be what we want. If we prefer the patrol point to be positioned midway between the two crosspoints, we could use the square of the distance instead of the absolute value, such as in cost function f_2:

$$f_2(t) = \sum_{t=t_{\mathrm{d}}}^{t_{\mathrm{d}}+T_{\mathrm{h}}} \sum_{o \in O} \min_{p \in P} |y_t^c - y_t^p|^2 \quad (3)$$

The reason is that by using the square, we punish larger distances more than smaller distances.

A Flaw In The Original Cost Function

Let the alarm time t_{a} denote when the tugs are alarmed that an oil tanker is adrift, and, as before, let t_{d} be the time the oil tanker actually starts drifting. Using the cost functions f_1 and f_2 for planning the trajectories of the tugs implies the assumption that $t_{\mathrm{a}} = t_{\mathrm{d}}$, that is, the tugs are alarmed immediately, and that tugs will continue to execute their original plans even after receiving an alarm. In reality, however, these assumptions are unrealistic. First of all, oil tankers will typically have drifted for some time, 3 hours say, before the tugs are alarmed, and hence, in general, t_{d} will occur earlier than t_{a}. Consequently, we can define a new, and shorter, drift-from-alarm (DFA) time $\hat{\Delta}_{\mathrm{a}}$, which is the drift time from the tugs receive an alarm at t_{a} until the drifting tanker crosses the patrol line at a crosspoint. For example, let us consider Figure 2, and assume that all oil tankers will take an estimated $\hat{\Delta} = 11$ hours, say, to drift aground from current positions. Hence, the first crosspoints that appear at $t = 8$ correspond to oil tankers starting drifting at $t = t_{\mathrm{d}} = -3$, the NCA being alarmed 3 hours later at $t = t_{\mathrm{a}} = 0$, and the DFA time becomes $\hat{\Delta}_{\mathrm{a}} = 8$ hours. Likewise, the crosspoints that appear at $t = 9$ correspond to oil tankers starting drifting at $t = t_{\mathrm{d}} = -2$, the NCA being alarmed at $t = t_{\mathrm{a}} = 1$, the DFA time becomes $\hat{\Delta}_{\mathrm{a}} = 8$ hours, and so on.

Second, when alarmed, tugs should abandon their plans and make every effort to intercept a drifting tanker before it runs aground. More relevant, therefore, are the positions of the tugs when they receive the alarm at time t_{a}, and the hypothetical future positions where drifting tankers will cross the patrol line some $\hat{\Delta}_{\mathrm{a}}$ hours later, where $\hat{\Delta}_{\mathrm{a}}$ is the total drift time (8–12 hours) less the time it takes before the tugs are being alarmed (3 hours), thus $\hat{\Delta}_{\mathrm{a}}$ is in the range 5–9 hours.

Cost Function f_3

To address the issues raised above regarding the original cost function f_1, we propose a modified cost function f_3 as given below:

$$f_3(t) = \sum_{t=t_{\mathrm{a}}}^{t_{\mathrm{a}}+T_{\mathrm{h}}} \sum_{o \in O} \min_{p \in P} \left| y_{t+\hat{\Delta}_{\mathrm{a}}}^c - y_t^p \right| \quad (4)$$

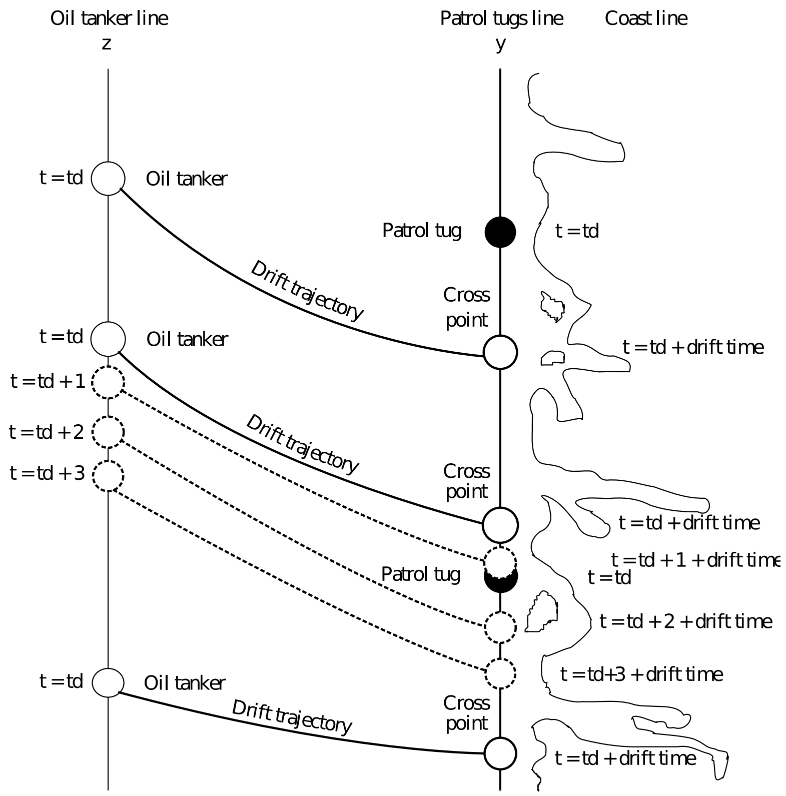


Figure 1: TFO Problem: Where Should Tugs Move?

Compared with f_1, f_3 is modified in two ways: (1) in the first sum, time t is upper-limited by $t_\mathrm{a} + T_h$ instead of $t_\mathrm{d} + T_h$ and lower-limited by t_a instead of t_d; and (2) in the absolute value term, we measure the distance between each cross point at some future cross time $t + \hat{\Delta}_\mathrm{a}$ and the position of the nearest tug at the alarm time t, and do this for the current and future potential alarm times.

Cost Function f_4

Again, combining the option of squaring the distances in f_2 with the modification in f_3, we also propose the cost function f_4 given by

$$f_4(t) = \sum_{t=t_\mathrm{a}}^{t_\mathrm{a}+T_\mathrm{h}} \sum_{o \in O} \min_{p \in P} \left| y^c_{t+\hat{\Delta}_\mathrm{a}} - y^p_t \right|^2 \tag{5}$$

Cost Function f_5

Yet another option for the choice of cost function is to categorise crosspoints within a certain safe range r as very likely to be reachable before grounding by a particular tug and therefore not to include these crosspoints in the cost function evaluation. Considering f_3 presented above, we simply subtract the safe range r from the distance and if the result is negative, we raise it to zero, as shown in f_5 below:

$$f_5(t) = \sum_{t=t_\mathrm{a}}^{t_\mathrm{a}+T_\mathrm{h}} \sum_{o \in O} \max \left\{ 0, \min_{p \in P} \left| y^c_{t+\hat{\Delta}_\mathrm{a}} - y^p_t \right| - r \right\} \tag{6}$$

A reasonable and conservative choice for r could for instance be half the expected distance a tug can travel from an alarm is received until the first hypothetical crosspoints occur.

In terms of minimising this cost function, one challenge will be that of flat cost surface regions for crosspoints within the safe range, which makes it more difficult to find an optimal solution.

Cost Function f_6

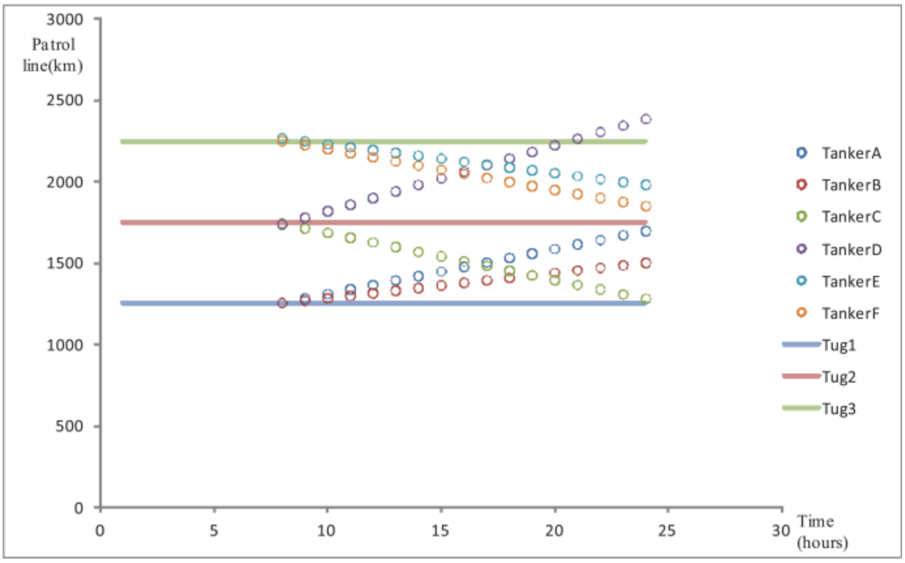
For the last cost function we will present, we continue to use the safe range r for measuring the number of unreachable crosspoints. If crosspoints are outside the safe range, we add 1 to the accumulated cost, otherwise 0. The cost function is given by f_6 below:

$$f_6(t) = \sum_{t=t_\mathrm{a}}^{t_\mathrm{a}+T_\mathrm{h}} \sum_{o \in O} g \left(\min_{p \in P} \left| y^c_{t+\hat{\Delta}_\mathrm{a}} - y^p_t \right| \right), \tag{7}$$

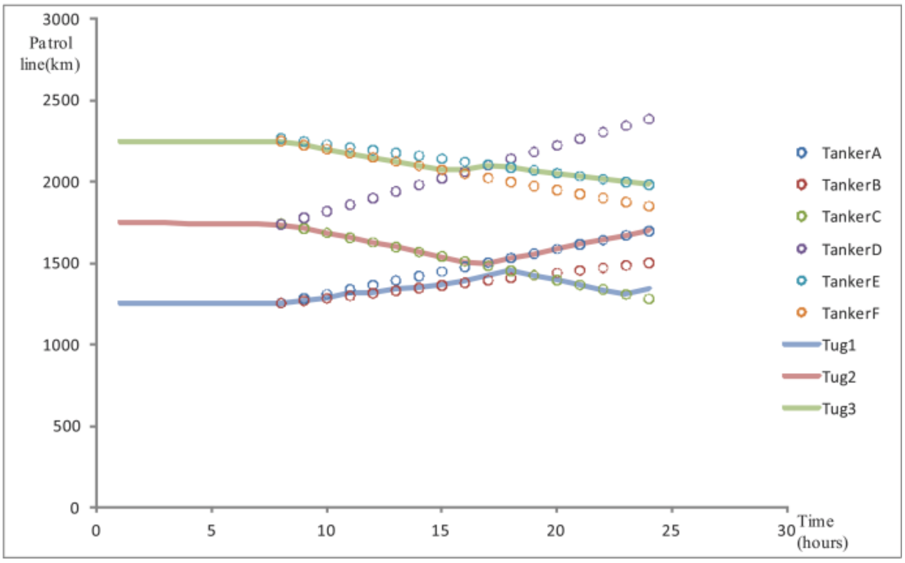
$$g(x) = \begin{cases} 1, & \text{if } x > r. \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Objective Evaluation Method For TFO Algorithms

By definition, the costs of different cost functions are not directly comparable, and we therefore need some kind of objective evaluation method for making comparisons. Here, we suggest that one such method is to (1) generate a deterministic and reproducible simulation scenario; (2) run the RHGA (or another TFO algorithm) for a given number of planning steps; (3) considering each oil tanker separately, assume each tanker begins drifting and count the number of salvageable tankers; (4) for the same simulation scenario, repeat (2) and (3) with a different



(a) Static Policy



(b) Optimal RHMIPA Solution

Figure 2: Example TFO Scenario

cost function in the RHGA (or a different TFO algorithm); and (5) repeat (1)–(4) for a number of different simulation scenarios and find the accumulated objective evaluation cost for each cost function (or TFO algorithm).

A simulation scenario in this case is simply a set of pre-determined oil tanker movements and the resulting hypothetical drift trajectories and crosspoints for a pre-specified duration. For testing purposes, we can generate a number of such scenarios offline and use them as input data for testing TFO algorithms. In a real-world application, the actual scenario is what that is happening right now, and future oil tanker positions, drift trajectories, and crosspoints would have to be predicted in real-time.

Other possible objective measures exist, e.g., we could sum up the total fuel consumption and use it as a component of an overall objective measure if that is of interest.

RECEDING HORIZON CONTROL

Above, we have presented a number of cost functions that can be minimised, with a GA or otherwise, to return a set of optimal tug trajectories. However, we need to present a method for handling the dynamic nature of the environment and parameters involved, and therefore adopt the principle of receding horizon control (RHC) that we have employed previously Bye et al. (2010); Bye (2012). Because neither oil tankers' speed and heading, nor wind, wave, and ocean current conditions are static, the resulting predicted future distribution of crosspoints will change, and patrol trajectories optimised by the GA will soon become outdated. One possibility for overcoming this problem is to run the GA at regular intervals, constantly incorporating updated *current* information about the state of the oil tankers and weather conditions as well as updated *predictions* of these factors. While tugs begin to move according to the solutions planned by the GA, new patrol trajectories can be calculated and replace the old ones. This feedback strategy is equivalent to a RHC scheme, which is interchangeably termed model predictive control (MPC) in the literature (e.g., see Maciejowski (2002); Rossiter (2004) for theoretical treatments).

In RHC, a control strategy that minimises some cost function is calculated a prespecified duration, namely the prediction horizon, into the future. However, only the first portion of this strategy is implemented before another control strategy is calculated based on new and predicted information available. The new solution replaces the old one but again only the first portion is implemented. This process repeats as a sequence of RHC planning steps.

A particular advantage of using RHC is that constraints can be handled in the design phase and not post hoc (e.g., see Goodwin et al. (2001); Maciejowski (2002)). For tugs, one such constraint is the inherent limitation of moving no faster than the maximum possible speed limited by the ship's engine, weather conditions, or even the wish to save fuel if one wants to take that into account. This maximum speed limits the number of reachable crosspoints. Using RHC it is possible to incorporate this constraint in the planning of tug patrol trajectories.

GA Optimisation Between Planning Steps

In the RHGA, a GA is used to solve an optimisation problem at every RHC planning step. A good choice of initial population allows the GA to find good solutions in fewer iterations than simply using a random population. It is possible to take the dynamics of the simulated scenario into account and, assuming that the scenario will not change significantly, a solution found at one planning step should also be a viable solution at the next planning step. This is achieved by an elitist strategy of keeping (a slightly modified version of) the best chromosome at one RHC step and inserting it into the initial population of the GA at the next RHC step. More details on the RHGA is outside the scope of this paper and has been presented previously (Bye et al., 2010; Bye, 2012).

SIMULATION RESULTS

Figure 3 shows an example simulation scenario where three tugs (black circles) are positioned at $y = [-500, 0, 500]$ at $t = 0$ and six oil tankers (not shown) are randomly positioned in open water along the oil tanker corridor z, limited to an observation zone of $[-750, 750]$ km. All plots depict time along the horizontal axis and position along the vertical axis. Using each of the cost functions presented previously, the plots show the first RHC planning step at $t = 0$ and the planned tug trajectories 24 hours ahead in time that collectively minimise the respective cost functions.

The tugs are limited to a maximum speed of 20 km/h, whereas the speeds of the oil tankers are randomly drawn from a uniform distribution on $[20, 30]$ km/h. Drift trajectories are perpendicular onto the patrol line, and crosspoints (red crosses) are generated from extrapolating the future positions of oil tankers and their resulting drift trajectories. Note that compared with previous work, we have reduced the maximum speed of the tugs from 30 km/h to 20 km/h, thus making it more difficult for tugs to cover potential crosspoints. The speeds used above are in line with the literature (e.g., Det Norske Veritas (2009), Eide et al. (2007a)).

Another difference when compared with our previous work is the more realistic scenario of oil tankers leaving or entering the observation zone. For simplicity, we have implemented this feature such that whenever an oil tanker leaves the zone to the north or south, another oil tanker enters at the opposite end.

We have drawn random drift durations for each oil tanker from a uniform distribution on [8,9,...,12], and set the alarm times used for f_3–f_6 to occur 3 hours after time of drift t_{d}. Hence, although we have shown crosspoints for the entire simulation interval, no crosspoints at time $t = 8-3 = 5$ or earlier will have an effect in the evaluation of cost functions f_3–f_6. Likewise, without the alarm time, no crosspoints at time $t = 8$ or earlier will have an effect in the evaluation of cost functions f_1 and f_2.

For f_5 and f_6, we set the safe region $r = 50$ km, using a conservative value corresponding to half the maximum speed (= 10 km/h) times the number of hours until the first crosspoints can occur, namely 5.

Effect Of Alarm Time

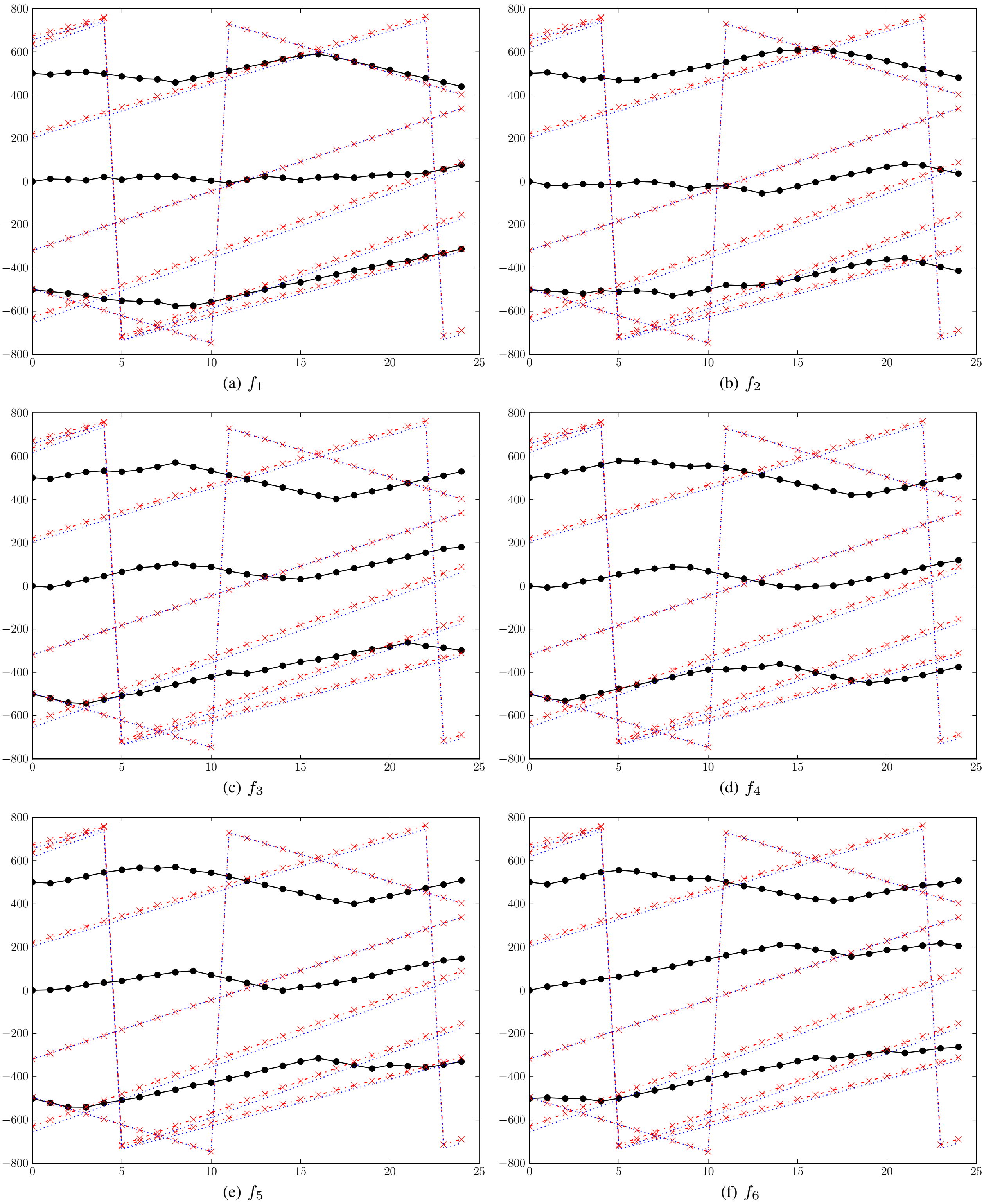
Let us first examine the effect of our newly introduced alarm time t_{a} in cost functions f_3–f_6, which means using the distances between crosspoints and the corresponding tug positions at the time of an alarm. Compared with f_1 and f_2, it seems evident that including the alarm time causes the trajectories to better anticipate future crosspoints. For example, the planned trajectory for the bottom tug of f_3–f_6 turns north already around $t = 2$ to $t = 4$, whereas this turnaround does not occur until $t = 8$ or $t = 9$ for f_1 and f_2.

Similarly, planned trajectories using f_3–f_6 clearly takes into account some future cost for the latter half of the simulation period, where many tugs turn the opposite direction of the tug trajectories planned using f_1 and f_2.

In short, it appears that for any point in the tug trajectories, the algorithm asks itself "where should the tugs be some hours ahead in time when the first crosspoints can occur?" and directs the tugs accordingly.

Effect Of Squaring

The effect of squaring can be seen for f_2 vs. f_1 and for f_4 vs. f_3. For f_2 (squared) vs. f_1 (non-squared), the planned trajectories for tugs are more likely to be positioned in-between crosspoints when using the squared cost function. For f_4 (squared) vs. f_3 (non-squared), another, related effect is visible for the bottom tug, which turns south to cover more of the southernmost cross points. Indeed, since squaring means punishing larger distances more, examination of several simulation scenarios not reproduced here shows that squaring leads to tugs spreading out more and covering larger areas.



(a) f_1 (b) f_2

(c) f_3 (d) f_4

(e) f_5 (f) f_6

Figure 3: RHGA Planning Using Cost Functions f_1–f_6

Effect Of Safe Region r

The effect of employing a safe region $r = 50$ km is shown for f_5 and f_6, which can be compared with f_3 that does not have a safe region. Comparing f_5 with f_3, the two topmost tug trajectories are more or less the same, whereas using a safe region causes the bottom tug to travel south towards the end of the simulation scenario for f_5. If we employ f_6, which accumulates all non-reachable crosspoints and weights them all equally as a value of unity, the top tug trajectory is quite equal to that of f_3, the middle tug trajectory seeems a bit like a lagged version of the middle tug trajectory of f_3 and f_5, and finally, the bottom tug trajectory of f_6 is quite stationary compared with f_3 and f_5.

Objective Evaluation Of TFO Algorithms

Above, we suggested counting the number of unsalvageable oil tankers at every point in time as an objective evaluation method for comparing TFO algorithms. Here, we used this measure to compare each of the cost functions but there was little difference between the cost functions (not shown graphically). Examination of a number of simulation scenarios does seem to indicate that f_1 and f_2 perform worse than the others, but this needs a quantitative analysis to be confirmed.

Conclusions

In the above, we have made some qualitative interpretations of a particular simulation scenario with respect to our proposed cost functions. Similar qualitative interpretations have been made for a number of simulation scenarios not presented in this paper. Our study is largely work-in-progress and needs to be expanded upon to include quantitative analyses for comparing the properties, pros and cons, of different cost functions and TFO algorithms. Nevertheless, we think our observations are worthwhile, and in particular, we believe the simulation results add to our proposition that the original cost function f_1 (and f_2, which is just a squared version of f_1) has a flaw by not including an alarm time and using the distances between tug positions at the time of an alarm and future positions of tankers after a DFA time $\hat{\Delta}_{\mathrm{a}}$. Unfortunately, no strong conclusions can be made about our suggestion for an objective evaluation method for comparing cost functions just from the qualitative results presented here.

DISCUSSION

We have previously presented a formulation for a TFO problem and two algorithms, the RGHA and the RHMIPA, that solve it (Bye et al., 2010; Bye, 2012; Assimizele et al., 2013). Here, we identify a flaw in the cost function employed in the RHGA, namely the assumptions that tugs will be alarmed immediately when an oil tanker begins drifting, and that tugs will continue to execute their original plans even after receiving an alarm. Clearly, these assumptions are unrealistic. We point out how this flaw can be corrected by means of defining a new, and shorter, drift-from-alarm (DFA) time, which is the estimated drift time from tugs receive an alarm until the drifting tanker crosses the patrol line. Incorporating this DFA time, we suggest several new cost functions that make comparisons between positions of tugs at the time of an alarm and the hypothetical future crosspoints of drifting tankers. This ensures that tugs abandon their planned original plans immediately upon receiving an alarm. The suggested cost functions can be used in a RHGA or in other TFO algorithms. We have tested the RHGA with a number of cost functions and simulation scenarios in our recently developed simulator framework. The simulation scenarios used in our previous work have been corrected and extended by allowing the realistic option of oil tankers both leaving and entering an observation zone, as well as lowering the maximum speed of patrol tugs to 20 km/h, thus making the problem even more complex to solve. Finally, we have suggested an objective evaluation measure for comparing various TFO algorithms and/or cost functions. More testing and analyses are needed in order to evaluate both the merit of the different cost functions and the suitability of the objective evaluation measure. The results are preliminary and a reflection of this paper as a report on work-in-progress but valuable nonetheless.

Simulator Framework

A thorough presentation of our new simulator framework requires a separate paper but we will cover the essential below. We chose the purely-functional programming language Haskell for our implementation, which means that functions in Haskell are pure, there is no global state, and no side effects. Code written in Haskell is therefore less error-prone and usually more concise, compact, and readable than imperative programming languages like C or Java. An additional advantages that "comes for free" with a functional language is a focus on *what* the programmer wants to achieve, rather than *how*, since functional program specifications can simply be executed directly rather than translated into imperative code.

A challenge, however, can be the use of pseudo-random number generators (PRNGs), which, by definition, are impure and require book-keeping of a system state. Nevertheless, using a functional language like Haskell provides a clear separation between pure and impure functionality, thus reducing this book-keeping to a minimum.

Whilst being strongly typed, and thus avoiding compile-time core dumps, Haskell uses polymorphism, which enhances the reusability of code. Many functions can therefore be written only once, because they accept input and output variables of many different types.

Haskell is also a good choice for parallel programming, which we believe is likely to be needed as the complexity of our simulator grows. Using pure parallelism guarantees deterministic processes and zero race conditions or deadlocks, however, non-pure concurrency related to PRNGs and other processes is also required.

Finally, it is worth mentioning that Haskell is a non-strict, lazy language, meaning that evaluation only happens on demand. This removes the need for the programmer to pre-allocate memory such as fixed-size arrays, and makes it easier to write modular programs, since functions can be passed freely to other functions, be returned as the result of a function, and stored in data structures.

Our choice of using Haskell for implementation makes our simulator very extendable and we are therefore confident that we will be able to perform several comprehensive and quantative studies in the time to come.

Future Work

There are several directions the DRAMA research group wishes to pursue in the way forward. First of all, we need to perform an extensive simulation study based on what we have presented here. This will include simulating a large number of scenarios and examining if our proposed objective evaluation method can be used for comparing our proposed cost functions, and more generally, for comparing TFO algorithms.

Furthermore, our simulator needs to be extended to accommodate a large number of realistic simulation parameters and scenarios, including variable maximum speeds of tugs (the maximum speed of a tug constantly varies with wave height and sea roughness at the geographical location), tugs being temporarily unavailable (e.g., due to change of crew), realistic drift trajectories based on realistic models, fuel consumption and environmental impact, and 2D scenarios (e.g., oil tankers entering or leaving port represent high risk).

Moreover, we are already working on probabilistic models, for example for assigning risk weights to oil tankers depending on factors such as the geographical location or size and type of oil being carried; for generating continous probability distributions of crosspoints; and for quantifying the probabiliy of intercepting tankers in drift.

Finally, it would be of interest to include historical records of traffic data for both oil tankers and tugs in the simulator and determine the performance of the real-world tugs compared to various TFO algorithms. Such a comparison, if thorough and sound, can be used for analytical purposes and to provide support (or lack thereof) of using TFO algorithms as a decision-support tool at the VTS centres around the world.

ACKNOWLEDGEMENTS

The DRAMA research group is grateful for the support provided by Regionalt Forskningsfond Midt-Norge and the Research Council of Norway through the project *Dynamic Resource Allocation with Maritime Application* (DRAMA), grant no. ES504913.

REFERENCES

Assimizele, B., Oppen, J., and Bye, R. T. (2013). A sustainable model for optimal dynamic allocation of patrol tugs to oil tankers. In *Proceedings of the 27th European Conference on Modeling and Simulation*, pages 801–807.

Breivik, Ø. and Allen, A. (2008). An operational search and rescue model for the Norwegian Sea and the North Sea. *Journal of Marine Systems*, 69(1–2):99–113.

Breivik, Ø., Allen, A., Maisondieu, C., and Roth, J. (2011). Wind-induced drift of objects at sea: The leeway field method. *Applied Ocean Research*, 33(2):100–109.

Bye, R. T. (2012). A receding horizon genetic algorithm for dynamic resource allocation: A case study on optimal positioning of tugs. *Series: Studies in Computational Intelligence*, 399:131–147. Springer-Verlag: Berlin Heidelberg.

Bye, R. T., van Albada, S. B., and Yndestad, H. (2010). A receding horizon genetic algorithm for dynamic multi-target assignment and tracking: A case study on the optimal positioning of tug vessels along the northern Norwegian coast. In *Proceedings of the International Conference on Evolutionary Computation (ICEC 2010) — part of the International Joint Conference on Computational Intelligence (IJCCI 2010)*, pages 114–125.

Det Norske Veritas (2009). Rapport Nr. 2009-1016. Revisjon Nr. 01. Tiltaksanalyse — Fartsgrenser for skip som opererer i norske farvann. Technical report, Sjøfartsdirektoratet.

Eide, M. S., Endresen, Ø., Breivik, Ø., Brude, O. W., Ellingsen, I. H., Røang, K., Hauge, J., and Brett, P. O. (2007a). Prevention of oil spill from shipping by modelling of dynamic risk. *Marine Pollution Bulletin*, 54:1619–1633.

Eide, M. S., Endresen, Ø., Brett, P. O., Ervik, J. L., and Røang, K. (2007b). Intelligent ship traffic monitoring for oil spill prevention: Risk based decision support building on AIS. *Marine Pollution Bulletin*, 54:145–148.

Goodwin, G. C., Graebe, S. F., and Salgado, M. E. (2001). *Control System Design*. Prentice Hall, New Jersey.

Hackett, B., Breivik, Ø., and Wettre, C. (2006). Forecasting the Drift of Objects and Substances in the Ocean. In *Ocean Weather Forecasting: An Integrated View of Oceanography*, pages 507–523. Springer.

Havforskningsinstituttet (2010). Fisken og havet, særnummer 1a-2010: Det faglige grunnlaget for oppdateringen av forvaltningsplanen for Barentshavet og havområdene utenfor Lofoten. Technical report, Institute of Marine Research (Havforskningsinstituttet).

Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice Hall, first edition.

Rossiter, J. A. (2004). *Model-based Predictive Control*. CRC Press.

AUTHOR BIOGRAPHIES

ROBIN T. BYE[2] graduated from the University of New South Wales, Sydney with a BE (Hons 1), MEngSc, and a PhD, all in electrical engineering. Dr. Bye began working at the AAUC as a researcher in 2008 and has since 2010 been an associate professor in automation engineering at AAUC. His research interests include dynamic resource allocation, optimisation, computational intelligence, cybernetics, and human movement control systems.

HANS G. SCHAATHUN[3] is cand.scient. 1999 and dr.scient. 2002 in coding theory from the University of Bergen and worked as lecturer and post.doc. there until 2006. He was lecturer and senior lecturer at the University of Surrey 2006–2010 specialising in multimedia security. Since 2011 he has been professor in computing at AAUC, focusing his research on modelling and simulation, machine learning, and software engineering.

[2]www.robinbye.com

[3]www.hg.schaathun.net