A Receding Horizon Genetic Algorithm for Dynamic Resource Allocation: A Case Study on Optimal Positioning of Tugs

Robin T. Bye

Department of Technology and Nautical Sciences, Ålesund University College,
Postboks 1517, N-6025 Ålesund, Norway
roby@hials.no
http://www.robinbye.com

Abstract. This paper presents a receding horizon genetic algorithm (RHGA) for dynamic resource allocation. The algorithm combines methods from control theory and computational intelligence to simultaneously solve the problems of (i) coordinated control of resources, (ii) task assignment, and (iii) multiple target tracking in a dynamic environment. A simulated case study on optimal positioning of a fleet of tugs along the northern Norwegian coast serves as a means of evaluating the algorithm. In terms of reducing the risk of oil tanker drifting accidents, the study shows that the RHGA is able to iteratively plan movement trajectories for each individual tug such that the net collective behaviour of the tugs outperforms that of stand-by tugs stationed at bases located uniformly along the coast. The promising results suggest great potential for further development and generalisation to other dynamic resource allocation problems.

Keywords: dynamic resource allocation, genetic algorithm, receding horizon control, model predictive control, optimal control.

1 Introduction

Dynamic resource allocation can be considered a broad class of optimisation problems, including search and rescue operations, vehicle routing, crew allocation and scheduling, and many others. This study focuses on the problem of allocating a group of resources to the tracking of multiple targets in a dynamic environment. Specifically, it considers a fleet of tugs operating along a coast line with the purpose of preventing oil tankers from drift grounding. The tugs must dynamically be assigned moving target positions for tracking such that the overall risk of any oil tankers drifting aground is minimised. Such a problem is a demanding one and poses a number of interrelated challenges.

A first challenge is that of task assignment: Which resources shall track which targets? On the one hand, if there are more resources than targets, a subset of resources could be given the task of tracking one target each whilst remaining resources could be given the task of self-maintenance or simply doing nothing.

On the other hand, if the number of targets exceeds the number of resources, some resources must be assigned more than one target. In both cases, some governing principle is needed for allocating resources.

A second challenge is that of target tracking: How should each resource move to best track, or cover, its assigned targets? When targets outnumber resources, there will be an inherent tradeoff between good tracking of some targets at the expense of bad tracking of others. In addition, some targets may be considered more important to track than others and therefore must be weighted more.

Collectively considering task assignment and target tracking for the resources as a group, a third challenge is that of coordinated control and collective performance: How should tasks be assigned and targets be tracked such that some net performance index is optimised?

Finally, being in a dynamic environment, these challenges need to be constantly reevaluated: How can future changes in the state space, such as motion of targets and changing dynamics of the surroundings, be incorporated?

This paper presents a receding horizon genetic algorithm (RHGA) for solving the abovementioned challenges. The performance of the algorithm is demonstrated in a simulated case study of a real-world problem, namely the positioning of tug vessels along the coastline of northern Norway.

The following sections provide background information on the tug positioning problem, derivation of a simplified and concise problem description, and details on a proposed algorithm that solves the problem by combining receding horizon control (RHC) with a genetic algorithm (GA). The performance of the RHGA will be demonstrated in some simulated scenarios. Finally, the simulation results, aspects of our approach, and future potential will be discussed.

1.1 The Tug Positioning Problem

Each year thousands of ship transits, including several hundred transits of oil tankers, are made along the coastline of northern Norway, thus exposing it to the risk of drift grounding accidents and oil spill [9]. In an effort to reduce the risk of such accidents, the Norwegian Coastal Administration (NCA) runs a vessel traffic services (VTS) centre in the town of Vardø, which administers a fleet of tugs patrolling the coastline. The main purpose of these tugs is to cleverly patrol the coastline in such a manner that if an oil tanker loses manoeuvrability through steering or propulsion failure, there will be a tug sufficiently close that it can intercept the drifting oil tanker before it runs ashore [4].

Oil tankers are required by law to sail along predefined piecewise-linear corridors approximately parallel to the coastline. Hence, for example by linearly extrapolating its speed along its corridor, it is possible to predict a tanker's future position. Moreover, all ships are required by international law to constantly transmit both static (identity, dimensions, cargo, etc.) and dynamic (position, speed, heading, etc.) ship information through the automatic identification system (AIS). The AIS information is transmitted both to other ships and nearby VTS centres and relayed on the Internet. Together with weather forecasts and dynamic models of wind, wave heights, and ocean currents, the AIS information can be used to predict potential drift trajectories and grounding locations for ships that lose manoeuvrability [4].

The NCA has developed risk-based decision support tools based on dynamical risk models that draws on a vast pool of information [4,5]. Some of this information is static and certain, such as a the type of ship, the nationality of its crew, and the amount and type of oil it is carrying. Information about other factors is dynamic and uncertain and requires modelling. Such factors include wind, waves, currents, accident frequency and consequences, oil spill size and potential impact, and others. The decision support tools aid the human operator at a VTS centre in directing tugs by determining high-risk target areas that tugs should approach. Nevertheless, with the projected rapid increase in oil tanker transits in coming years [9] and the increasing number of tugs required for adequate patrolling, the problem quickly becomes unmanageable by a human operator. 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.

1.2 Problem Formulation

Before developing an algorithm, the tug positioning problem must be formulated carefully and precisely. First, it is assumed 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 follow predefined piecewise-linear corridors. Inside of z and closer to shore, it is assumed that N_{p} tugs are patrolling along a line of motion y parallel to z. The possibility of collisions between oil tankers and patrol tugs on their respective lines of motion is not considered.

It is acknowledged that the coastline does not constitute a sequence of connected straight line segments due to its vast amount of fjords, peninsulas, and islands. Nevertheless, because tugs should stop drifting ships *before* they reach land or danger zones, a straight patrol line some distance from the rugged coastline can be considered a conservative choice. Figure 1 shows a graphical representation of the problem description, illustrated by two patrolling tugs and three oil tankers.

Moreover, the algorithm assumes real-time access to prediction data from a set of accurate models such as those developed by the NCA and described in Sect. 1.1. These models must be 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.

Suppose an oil tanker currently positioned at $z(t)$ starts drifting at some time $t = t_{\mathrm{d}}$. The algorithm requires a future position trajectory predicted 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 positions are given by $\hat{z}(t|t_{\mathrm{d}})$ for $t = t_{\mathrm{d}} + 1, t_{\mathrm{d}} + 2, \ldots, t_{\mathrm{d}} + T_{\mathrm{h}}$.

Moreover, for each predicted position $\hat{z}(t|t_{\mathrm{d}})$ there is a corresponding predicted drift trajectory starting at $\hat{z}(t|t_{\mathrm{d}})$ that may or may not intersect the

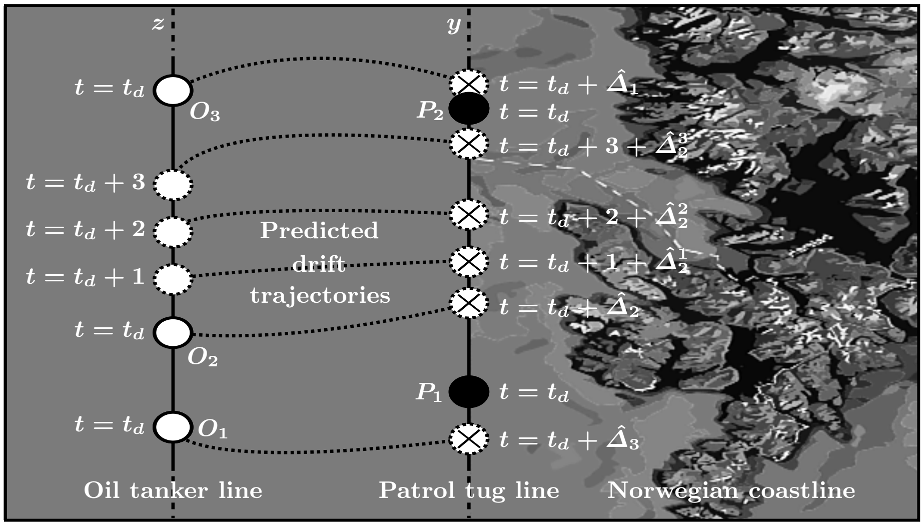


Fig. 1. Problem description. Patrol tugs P_1 and P_2 (black circles) and oil tankers O_1, O_2, and O_3 (white circles) move unidimensionally along lines y and z, respectively. Solid circles correspond to positions at current time of drift $t = t_\text{d}$, whereas dashed lines and circles indicate predicted drift trajectories and positions for $t > t_\text{d}$. The estimated duration of each drift trajectory is denoted $\hat{\Delta}_i^k$, where i and k refers to the drift trajectory for the ith oil tanker that begins k hours ahead in time at $t = t_\text{d} + k$. Circles with a cross indicate cross point points. The dynamic resource allocation problem is that of determining how the tugs collectively should move in the time ahead in order to best reduce the risk of drift grounding accidents.

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 more. Collecting all predicted drift trajectories for all oil tankers results in a distribution of *cross points* where future drift trajectories will intersect the patrol line.

Based on the predicted distribution of cross points, the 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 towed to safety before grounding is minimised. This is a difficult problem involving collective behaviour, task assignment, and multi-target tracking in a dynamic environment.

2 Method

An example scenario in Fig. 2 shows three tugs and three corresponding random walk patrol trajectories the tugs may follow in order to track the cross point distributions of six oil tankers. As seen from the plot, these patrol trajectories provide poor coverage of cross points. The tugs stay more or less around their initial positions, leaving a large number of cross points unattended. A better solution would have the tugs spread out, each tug covering its own set of cross points, and thus improving the overall coverage.

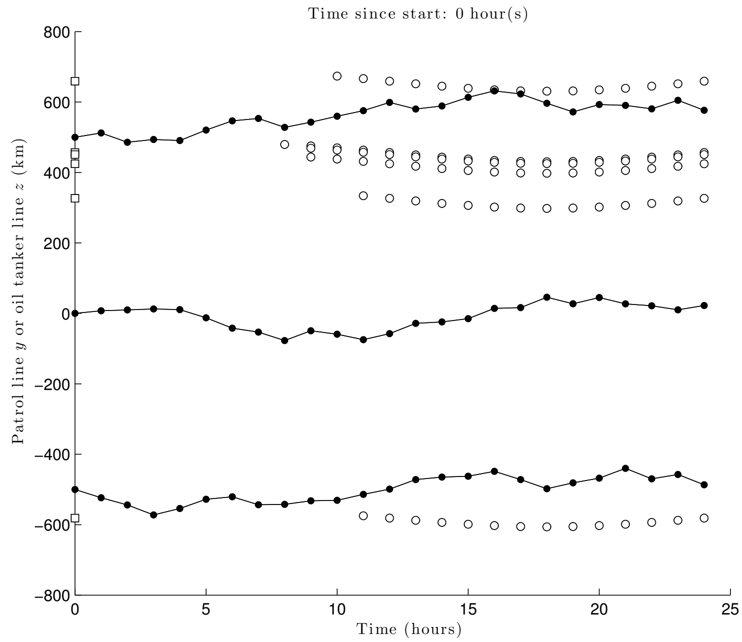


Fig. 2. Example scenario. White squares on the vertical axis indicate the initial positions of six oil tankers on the oil tanker line z at $t = 0$. White circles indicate where the corresponding predicted drift trajectories will cross the patrol line y as a function of time. Black circles indicate three random walk patrol trajectories on y for $t = 0, 1 \ldots, T_\mathrm{h}$, where the prediction horizon is set to $T_\mathrm{h} = 24$ hours. The area reachable by each of the three tugs at full speed from their initial positions are depicted by envelopes delimited by dotted lines. Note that the smallest observed drift time is $\hat{\Delta} = 8$ hours, whereas the largest drift time is $\hat{\Delta} = 11$ hours. An algorithm for tug positioning must generate patrol trajectories such that the tugs cover the distribution of cross points as well as possible.

How can the quality, or performance, of the choice of patrolling tug trajectories be measured? One possible approach is to examine a large number of sets of potential patrol trajectories and for each set evaluate a cost function that quantifies the performance of the tugs. There are several methods that likely can find near-optimal solutions in reasonable time for this approach, for example variants of Monte Carlo methods, simulated annealing, ant colony optimisation, genetic algorithms (GAs), or other methods from computational intelligence (e.g., see [14] for an overview). This study uses a continuous GA based on a version similar to that described in [8].

Another challenge is how to accomodate changing dynamics of the environment. Not only may oil tankers change their speeds and headings but weather conditions also constantly change. Consequently, cross point distributions will change with time and solutions must be recalculated. One solution is to redo the search for candidate patrol trajectories at regular intervals and replace the previously generated trajectories with new ones. Such a scheme is implemented in the tug positioning algorithm by utilising receding horizon control (RHC).

Details of the GA and RHC strategy are presented in Sects. 2.1 and 2.2, respectively.

2.1 The Genetic Algorithm

A GA is a heuristic search method based on principles of natural evolution (e.g., see [6,8,14] for detailed descriptions). GAs are particularly useful for obtaining solutions to difficult optimisation and search problems where the solution space is nonconvex. Although no mathematical analysis about the convexity of the tug position problem is provided in this paper, it seems clear that generating patrol trajectories that are optimal, or near-optimal, in some sense by minimising a cost function such as (3) (defined later) is not easy and utilising a GA is an appropriate choice.

Characteristics of the GA. The GA used here consists of the following steps, which adheres to the general scheme used in most GAs (e.g., see [8]):

1. Define a cost function and a chromosome encoding and set some GA parameters such as mutation and selection.
2. Generate an initial population of chromosomes.
3. Evaluate a cost for each chromosome.
4. Select mates based on a selection parameter.
5. Perform mating.
6. Perform mutation based on a mutation parameter.
7. If the desired number of iterations or cost level is reached, stop algorithm and return solution, otherwise, repeat from Step 3.

The selection parameter is in the range 0–1 and determines how many chromosomes in a population survives from one iteration to the next. The cost associated with each chromosome is evaluated and the chromosomes are given a weighted selection probability according to their cost, where a smaller cost results in a greater probability. For a selection parameter of 0.5, half the population is then randomly picked, with low cost chromosomes having a greater chance of being picked and kept for survival and reproduction. The other chromosomes are discarded to make room for new offspring.

For mating, the GA uses a combination of an extrapolation method and a crossover method. Information from two parent chromosomes are combined with an extrapolating method to obtain new offspring variable values bracketed by the parents' variable values. A single crossover point is used to determine which parts of the parent chromosomes are used for creating offspring.

After mating, a fraction of the genes are mutated, which means that the values of these genes are changed to random numbers within an allowable range. A mutation rate determines how many genes are mutated at every iteration.

Particular to the problem described in this paper is the choice of cost function and chromosome encoding, which are described in the following.

Cost Function. Proper choice of a cost function is imperative for the algorithm to find desirable solutions. Here, the cost function is defined as the sum of the distances between all cross points and the *nearest* patrol points. The rationale behind this choice is that if an oil tanker in drift can be saved by a tug a certain distance away, it is not important that other tugs further away can save it at a later time.

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 sum the minimum rescue times for each cross point.

By intuition, the emphasis on punishing distances to only the nearest patrol points should yield proper task assignment, as good solutions found by the GA will tend to have patrol tugs spreading out and tracking different groups of cross points, thus collectively reducing the overall risk of grounding. Indeed, this intuition is confirmed by the results presented in Sect. 3.

A cross point (position on y intersected by a drift trajectory) of the cth oil tanker's drift trajectory at time t can be defined as y_t^c. For the prediction horizon T_h there is a set of cross points given by

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

however, only a subset of these points are defined, since the drift trajectories must actually cross the patrol line at the specified times. For example, if the predicted drift time is $\hat{\Delta}$ for all drift trajectories, the earliest occurrence of a cross point of a drift trajectory starting on z at $t = t_\mathrm{d}$ will be at $t = t_\mathrm{d} + \hat{\Delta}$, and

$$\{y_t^c\} = \left\{y_{t_\mathrm{d}+\hat{\Delta}}^c, y_{t_\mathrm{d}+1+\hat{\Delta}}^c, \ldots, y_{t_\mathrm{d}+T_\mathrm{h}}^c\right\} \quad . \tag{2}$$

A patrol point (tug position on y) on the pth tug's patrol trajectory at time t can be defined as y_t^p. For N_o oil tankers and N_p patrol tugs, then, the cost $f(t, \mathbf{C}_i)$ is defined as a function of time t and the ith chromosome $\mathbf{C}_i$ by

$$f(t, \mathbf{C}_i) = \sum_{t=t_\mathrm{d}}^{t_\mathrm{d}+T_\mathrm{h}} \sum_{c=1}^{N_\mathrm{o}} \min_{p \in P} |y_t^c - y_t^p| \quad , \tag{3}$$

where $P = \{1, 2 \ldots, N_\mathrm{p}\}$ and details on y_t^p and $\mathbf{C}_i$ are given below.

Chromosome Encoding. For tug p, consider a sequence $\{u_t^p\}$ consisting of T_h normalised control inputs, or speed commands, u_t^p, where

$$\{u_t^p\} = \left\{u_{t_\mathrm{d}+1}^p, u_{t_\mathrm{d}+2}^p, \ldots, u_{t_\mathrm{d}+T_\mathrm{h}}^p\right\}, \quad -1 \le u_t^p \le 1 \quad . \tag{4}$$

The maximum control input values of -1 and 1 are equivalent to tugs going with maximum speed in the negative or positive y-direction, respectively. This encoding is generic as it is independent of each tug's maximum speed.

Given a control input u_t^p, a point y_t^p on the patrol trajectory for tug p at time t can be obtained through linear extrapolation using the difference equation

$$y_t^p = y_{t-1}^p + u_t^p v_{\mathrm{max}}^p T_{\mathrm{s}} \ , \tag{5}$$

where v_{max}^p is the maximum speed for the pth tug and T_{s} is the duration of each time step. The entire patrol trajectory is a sequence of T_{h} patrol points given by

$$\{y_t^p\} = \left\{y_{t_{\mathrm{d}}+1}^p, y_{t_{\mathrm{d}}+2}^p, \ldots, y_{t_{\mathrm{d}}+T_{\mathrm{h}}}^p\right\} \ . \tag{6}$$

To encode N_{p} control trajectories as sequences $\{u_t^p\}$ of length T_{h} for each patrol tug $p \in P$, the ith chromosome $\mathbf{C}_i$ of length $N_{\mathrm{p}} \times T_{\mathrm{h}}$ is encoded as

$$\mathbf{C}_i = \left\{u_1^1, \ldots, u_{T_{\mathrm{h}}}^1, u_1^2, \ldots, u_{T_{\mathrm{h}}}^2, \ldots, u_1^{N_{\mathrm{p}}}, \ldots, u_{T_{\mathrm{h}}}^{N_{\mathrm{p}}}\right\} \ . \tag{7}$$

That is, each chromosome is a concatenation of N_{p} control trajectories, each of which consists of T_{h} future control inputs. Given an initial tug position $y_{t_{\mathrm{d}}}^p$ and employing (5) repeatedly, these control trajectories are used to generate the patrol trajectories in (6).

2.2 Receding Horizon Control

Because of the dynamics of the problem, where neither oil tankers' speed and heading nor wind, wave, and ocean current conditions are static, patrol trajectories optimised by the GA will soon become outdated. One possibility is to run the GA at regular intervals, constantly incorporating updated *current* information about the state of the oil tankers and weathers 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 strategy is equivalent to a RHC scheme, which is interchangeably termed model predictive control (MPC) in the literature (e.g., see [11,13] 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. 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 steps.

RHC is currently one of the most popular control algorithms employed in computer-controlled systems, predominantly in the petrochemical industry, but also increasingly so in electromechanical control problems (e.g., see [7]). It can be shown that RHC can be designed with guaranteed asymptotic closed-loop stability [7] and this remarkable property is perhaps the most important reason for its popularity.

Constraints. An advantage of using RHC is that constraints can be handled in the design phase and not post hoc (e.g., see [7,11]). For tugs, such a constraint is the inherent limitation of moving no faster than their maximum speed. This speed limits the size of the envelopes in Fig. 2 and thus the number of reachable cross points. Using RHC it is possible to incorporate this constraint in the planning of tug trajectories.

Optimisation. 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 RHC step should also be a viable solution at the next RHC 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.

2.3 Simulation Study

The technical computing software package MATLAB.[1] was used for the implementation of a simulation study of the tug positioning problem. A number of choices had to be made about properties of oil tankers and patrol tugs, the GA and RHC, and general settings. Based on preliminary work [1] and extended testing, the settings described below were chosen.

Number of Ships. Based on information provided by NCA staff or affiliates and a recent report [9], it was decided to use $N_{\mathrm{p}} = 3$ tugs and $N_{\mathrm{o}} = 6$ oil tankers for the simulations. Whereas these numbers were realistic as of 2010, they will increase significantly the next decades due to the development of oil and gas fields in the area (see Sect. 4.5).

Position of Ships. The initial position of oil tankers at time $t = 0$ was varied for each simulation, with oil tankers being placed on z (in km) at positions drawn randomly from a uniform distribution in a 1500-km range from $z = -750$ to $z = 750$. Dividing the same range on y into $N_{\mathrm{p}} = 3$ equally-sized segments of length 500 km, the patrol tugs were always positioned initially at tug bases located in the centre of these segments, namely at $y = -500, 0, 500$. The reason for this was to compare the performance of the actively patrolling tugs controlled by the RHGA with keeping the patrol tugs on stand-by at uniformly distributed stationary bases.

Velocities of Ships. According to [2], oil tankers have a typical operating speed of 14–15 knots whereas tugs have a global average maximum speed of about 12

[1] MATLAB R2010b, available at www.mathworks.com.

knots, spanning from 5–26 knots [4]. In the geographical area of this case study, the typical maximum speed of tugs is 15 knots and operating speed of oil tankers is 10–14 knots.[2]

Based on these figures, each oil tanker was initialised with a random speed in either the negative (southbound) or positive (northbound) y-direction and drawn from a uniform distribution in the range $\pm[20, 30]$ (km/h). The oil tankers maintained their respective speeds throughout each simulation.

The patrol tugs were assigned a maximum speed of 30 km/h, corresponding to the envelopes presented previously in Figure 2.

Drift Trajectories. Wind, wave heights, ocean currents, oil tanker size and shape, and other factors lead to nonlinear drift trajectories perhaps resembling those in Fig. 1. To implement nonlinearity, it was assumed that any oil tanker in drift will follow an eastbound sinusoidal trajectory with period equal to T_h scaled by its velocity v.[3] That is, if the cth oil tanker with velocity v and position $z(t_\mathrm{d})$ loses manoeuvrability at $t = t_\mathrm{d}$, it is predicted to drift across the patrol line at

$$y^c_{t_\mathrm{d}+\hat{\Delta}} = z(t_\mathrm{d}) + v \sin\left(\frac{2\pi}{T_\mathrm{h}}\hat{\Delta}\right) \tag{8}$$

after a predicted drift time $\hat{\Delta}$.

For each oil tanker, a random integer drawn from a uniform distribution $[8, 9 \ldots, 12]$ was chosen as its predicted drift time and kept constant throughout each simulation. According to [4], drift times of only 10 hours are considered "fast drift," whereas "slow drift" means that most tankers will not run aground for the first 20–30 hours of uncontrolled drift. Thus, the choice of drift times in the interval 8–12 hours is a conservative estimate. In most cases, tugs will have more time to come to the rescue of a drifting ship.

GA Settings. At every RHC step, the GA was set to perform $N_\mathrm{iter} = 100$ iterations searching for a solution set of optimal patrol trajectories minimising the cost function given by (3). As discussed in Sect. 4.3, each RHC step keeps a modified version of the best chromosome found in the previous RHC step. This ensures that much fewer iterations are needed in later RHC steps than early ones.

The population size was set to 10 chromosomes, the mutation rate was set to 0.1, and the selection parameter was set to 0.5. Together with the other simulation parameters, these choices gave a good tradeoff between exploration and exploitation.

RHC Settings. The GA was used to search for optimal trajectories with a duration of $T_\mathrm{h} = 24$ hours for the patrol tugs. At every RHC step, each of

[2] Information provided by a close affiliate of the NCA.

[3] Note that this relationship is not physically realistic but simply chosen for the sake of introducing nonlinearity.

duration $T_\mathrm{s} = 1$ hour, only the first sample of these trajectories was executed by each tug before another solution set of trajectories was generated by the GA. This process was repeated for $N_\mathrm{RHC} = 26$ RHC steps, yielding scenarios simulated from $t_\mathrm{d} = 0$ to $t_\mathrm{d} = 25$ hours.

General Settings. A total of $N_\mathrm{sim} = 30$ scenarios (random initial positions, velocities, and drift times of oil tankers) were simulated. For each scenario i, the minimum costs found by the GA at each RHC step were calculated and the average cost stored as the ith element in a vector $\boldsymbol{f}_\mathrm{RHGA}$ of length N_sim. Similarly, the costs incurred if the patrol tugs stayed on stand-by at their individual bases were calculated and the average stored as the ith element in a vector $\boldsymbol{f}_\mathrm{static}$ of length N_sim.

Settings Summary. The simulation settings are summarised in Table 1.

Table 1. Simulation settings.

Oil tanker settings	
Number of tankers N_o	6
Random initial position (km)	$[-750, 750]$
Random velocity (km/h)	$\pm[20, 30]$
Drift field	sinusoidal eastbound
Random drift time $\hat{\Delta}$ (hours)	$[8, 9, \ldots, 12]$
Patrol tug settings	
Number of tugs N_p	3
Initial positions (km)	$\{-500, 0, 500\}$
Max velocity (km/h)	± 30
GA settings	
Iterations N_iter	100
Population size	10
Mutation rate	0.1
Selection	0.5
RHC settings	
Prediction horizon T_h (hours)	24
Simulation step size T_s (hours)	1
Number of steps N_RHC	26
General settings	
Number of scenarios N_sim	30
Strategies	RHGA, static

3 Results

3.1 Simulation Example

Figure 3 shows a simulation example using the settings given in Table 1. Initially at time $t_\mathrm{d} = 0$ (Fig. 3(a)), three patrol trajectories, each of duration $T_\mathrm{h} = 24$ hours, are planned for the tugs based on the predicted distributions of cross points. The first tug at $y_0^1 = -500$ is assigned the task of covering the isolated bottom cluster of cross points centred around $y = -600$, whereas the second tug at $y_0^2 = 0$ is assigned the top cluster of cross points centred around $y = 100$. The third tug at $y_0^3 = 500$ is not assigned any cross points and given a "don't care", or random walk, trajectory.

In Fig. 3(b), the positions of oil tankers and patrol tugs are shown for $t_\mathrm{d} = 5$. Because of the last five hours of oil tanker movements, there are now three distinct clusters of cross points. The GA now performs task reassignment by planning for the top tug to cover the top cluster of cross points. The middle tug is assigned the middle cluster, and the bottom tug the bottom cluster.

The remaining Figs. 3(c)–3(f) shows how the scenario develops for $t_\mathrm{d} = 10, 15, 20, 25$, with the three tugs constantly being assigned and tracking sets of cross points, whose positional distributions change with time.

In terms of performance one may compare the cost of the RHGA-generated trajectories to that of static trajectories, that is, keeping each patrol tug stationary at its base. For this simulation example, which was scenario number 4, the static cost was 10844, whereas the RHGA cost was 3130, representing a cost reduction of 71.1 %.

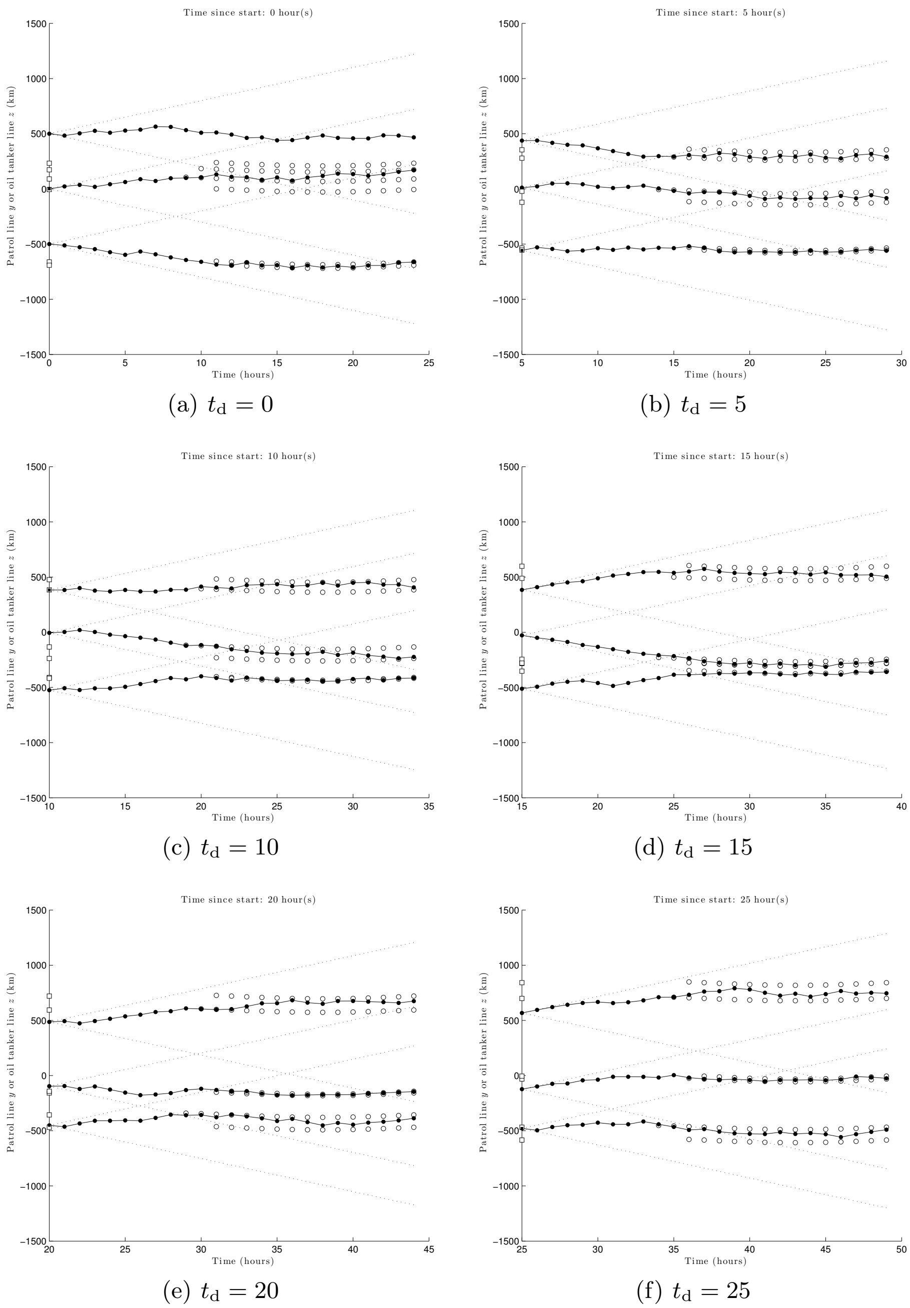
3.2 Main Study

Table 2 summarises the results from simulation scenario number 4 in Fig. 3 and 29 other simulated scenarios based on the settings presented in Table 1. For every scenario, the costs of each RHC step were summed and averaged for both the static case and the RHGA case. The mean cost for the 30 scenarios was 7372 for the RHGA and 17342 for the static strategy. This represents a mean improvement, or performance, of 57.5 % by the RHGA.

Comparing the standard deviation (STD) of the costs of the static case and the RHGA case, the STD of the RHGA was smaller by 34.6 %. However, because the static case has a much higher mean, its relative STD of 0.211 is 53.9 % smaller than the relative STD of the RHGA. Hence, relative to respective means, the cost of the static strategy varied less (it is consistently high) than that of the RHGA.

The minimum cost for a single scenario was 10844 for the static strategy and 3130 for the RHGA. Incidentally, the minimum cost occurred in scenario 4 for both cases. Inspection[4] of this simulated scenario showed that cross points

[4] Where no figure is referred to the reader must trust the inspection made by the author.



(a) $t_{\mathrm{d}} = 0$ (b) $t_{\mathrm{d}} = 5$

(c) $t_{\mathrm{d}} = 10$ (d) $t_{\mathrm{d}} = 15$

(e) $t_{\mathrm{d}} = 20$ (f) $t_{\mathrm{d}} = 25$

Fig. 3. Example simulation.

were distributed close to tug bases throughout the simulation, thus the static strategy resulted in a low cost. Still, the RHGA outperformed the static solution by 71.1 %.

The maximum cost for a single scenario for the static case was 22846 (scenario 2). Inspection of this scenario revealed that for most of the RHC steps, a large portion of cross points was distributed far to the south of the nearest southernmost tug base and thus the static strategy had a high cost for this scenario. The RHGA solution, on the other hand, was very good at this scenario with a cost of only 4796, thus outperforming the static solution by 79.0 %, which was also the highest cost reduction by the RHGA for any of the scenarios.

The maximum cost for a single scenario for the RHGA was 12503 (scenario 23). Inspection of this scenario showed that a small cluster of cross points was located far away to the south from the other cross points. Minimisation of the cost function by the RHGA resulted in a solution where patrol trajectories ignored this small cluster, which contributed to the high cost. The RHGA still outperformed the static case by 37.6 %.

The worst performance in terms of cost reduction by the RGHA occurred in scenario 3, where the static solution had a cost of 16544 and the RHGA had a cost of 11415, or a reduction of only 31.0 %. Inspection of this scenario showed that cross points were divided into six separate clusters, each far away from the others. The static solution performed better than its average cost for this scenario, which is unsurprising, given that uniformly spread out tug bases is a good choice for uniformly spread out cross points.

Finally, subtracting 1.645×STD from the mean cost of the static strategy shows that an estimated 95 % of all scenarios in the static case will have a cost *greater* than approximately 11300. Similarly, adding 1.645×STD to the mean cost of the RHGA shows that an estimated 95 % of all simulated scenarios employing the RHGA will have a cost *smaller* than approximately 11300. The likelyhood that the static solution is at least as good as the RHGA solution is thus very small.

Table 2. Simulation results.

Statistic	$\boldsymbol{f}_{\text{static}}$		$\boldsymbol{f}_{\text{RHGA}}$		Reduction by RHGA	
Mean	17342		7372		57.5 %	
STD	3667		2399		34.6 %	
Relative STD	0.211		0.325		−53.9 %	
Minimum (scenario)	10844	(4)	3130	(4)	31.0 %	(3)
Maximum (scenario)	22846	(2)	12503	(23)	79.0 %	(2)
95 % bounds	11309	or more	11318	or less	−0.1 %	

3.3 Conclusions

The simulation results show that the RHGA is able to simultaneously perform coordinated control, task assignment, and multiple target tracking in a dynamic environment. Based on current and predicted information, a GA calculates patrol trajectories that minimise a cost function. However, as the environment changes, an RHC process must be employed, where the GA constantly replans new trajectories based on the most recent data.

Employing a cost function related to the distance from each cross point to the nearest predicted patrol trajectory gives good tracking but also provides task assignment "for free." The resulting patrol trajectories suggested by the RHGA yield good prevention against possible drift accidents due to taking the predicted future environment into account.

4 Discussion

The simulation study presented here is substantially updated and extended compared to a preliminary study presented previously [1]. Modifications that have been made include realistic movement distances and speeds of tugs and oil tankers, nonlinear drift and cross point trajectories, an improved static strategy with tugs on stand-by uniformly positioned at stationary bases, a more rigorous problem formulation and definition of cost function, and a detailed analysis of the results.

In the following, some important aspects of the present study will be discussed.

4.1 Evaluation of Performance

Performance was measured by comparing the RHGA with a very simple static strategy where tugs are kept stationary on stand-by at bases located uniformly along the coastline. The static method is good for very large and uniform distributions of cross points. For small numbers of oil tankers, on the other hand, this method does not perform well because cross points will often exist far from tug bases. As demonstrated in Sect. 3, the RHGA significantly outperformed the static strategy for all simulated scenarios.

An alternative to the static method is a simple heuristic method such as letting patrol trajectories move towards the nearest cross point. Preliminary studies not presented here show that this method performs well when the numbers of tugs and tankers are approximately equal but as the number of tankers increases its performance drops significantly compared to that of the RHGA. The reason, of course, is that when all tugs have been allocated a distributions of cross points, superfluous distributions will be ignored, which in turn causes evaluations of the cost function to increase drastically.

It still remains to compare the RHGA with other intelligent algorithms for the same problem as defined in this paper. Nevertheless, the results from this study are very promising and shows that the RHGA provides a viable method for solving dynamic resource allocation problems of the kind presented here.

4.2 Choice of Cost Function

Choosing a suitable cost function is essential for a GA to be able to solve the problem at hand. Although the selected cost function (3) seems to be a reasonable choice, there are likely other choices that may be equally, or better, suited to our problem.

A potential modification to the cost function is to include a term for the control input in order to punish excessive fuel consumption. If so, care must be taken to ensure that this does not compromise the main goal of covering cross points and reducing the overall risk picture.

Another option is to introduce risk weights on oil tankers and scale the minimum distances in the cost function by these weights. Such risk weights already exists in the models of the NCA.

Finally, it would be interesting to let tugs have different maximum speeds and also let the speeds of tugs and oil tankers vary over time due to weather conditions, cargo and fuel effects, and other factors. In this case, the cost function would have to be modified to sum minimum *rescue times*, and not minimum distances.

4.3 Optimisation

For a slow-changing dynamic environment, a good chromosome at one time instant is likely a good chromosome at the next. Consequently, as described in Sect. 2.2, the RHGA keeps the best chromosome from one RHC step and places it in the initial population of the next. The other chromosomes are randomly initialised as usual.

If desirable, this strategy can be used to reduce the overall number of GA iterations since only a fraction of the initial number of iterations is needed for subsequent RHC steps. This is because the dynamics are slow-varying and the GA will tune in to good solution spaces where previously found solutions greatly assists the GA in finding new, good solutions.

4.4 Real-Time Requirements

Simulating a single RHC step with three tugs and six oil tankers for a particular scenario took about 30 seconds on a MacBook Pro Core 2 Duo 2.53 GHz computer. Increasing the number of oil tankers tenfold to 60 (which might be realistic in the not too distant future, see Section 4.5), one RHC step took slightly less than five minutes. This shows that the RHGA can accommodate much greater complexity than simulated in this study while staying within the real-time requirement of finishing each RHC step within an hour of real-time. It also implies that more accurate solutions can be obtained by increasing GA parameters such as population size and number of iterations at each RHC step.

Conversely, the small execution time for a RHC step means that the simulated duration of a RHC step can be greatly reduced if desired. This may not be relevant for the study presented here but implies that systems with much faster dynamics may take advantage of the RHGA. An example where each RHC step must be in the range of tenths of seconds or smaller is real-time control of football-playing robots, where algorithm speed will most definitely be an issue. For such applications, it is possible to adjust the GA and RHC settings to obtain small RHC step durations as required. Specifically, one may reduce the prediction horizon, number of iterations, and population size. This may not necessarily degrade performance. For example, employing a large prediction horizon in a football game where it is only possible to predict actions a short period ahead will not increase performance, it may even degrade it if it causes each RHC step to take too long.

4.5 Other Simulation Scenarios

Based on recent information in a governmental report made by the Norwegian Institute of Maritime Research [9], the choice of three tugs and six oil tankers represents a realistic and typical scenario as of today. Nevertheless, due the development of large oil and gas fields in the Barents Sea such as Goliat, Snøhvit, and Shtokman, oil and gas tanker traffic will drastically increase over the next 10–15 years. As mentioned in Section 4.1, the RHGA can easily handle the tenfold number of oil tankers while maintaining real-time requirements and thus appears well suited for much heavier traffic than that of today.

Cases where drifting actually occurs or situations where a tug becomes unavailable due to refuelling, change of crew, or being busy rescuing a drifting tanker have not been simulated. Moreover, no attempts have been made at trying to estimate how many tugs are necessary to maintain a sufficient degree of safety for a given number of tankers. These issues are highly relevant considering the foreseen increase in tanker activity.

Finally, it would be of interest to include more realistic two-dimensional (2D) planning for ships and three-dimensional (3D) planning for aeroplanes or submersible vehicles. This should be investigated further, particularly in light of other applications where dynamic resource allocation takes place at higher frequencies than for the tug positioning problem.

4.6 Other Applications

In addition to the one-dimensional (1D) problem described in this paper, the RHGA could be modify for performing multi-target allocation and tracking also in 2D and 3D dynamic environments. A 2D version in environments with slow dynamics may not require huge modifications, however, for fast dynamics and/or 3D environments, the algorithm must be improved, for example through distributed evaluation of the cost function.

Moreover, it could be interesting to combine the RHGA with so-called boid, or flocking, rules involving cohesion, separation, and alignment [12]. In a promising effort, [10] presents a flocking algorithm that modifies the flocking rules by [12] and succeeds in multi-target tracking performed by multiple agents. Through further development, a modified version of the RHGA could perform equally well as the algorithm used for the scenarios described by [10].

Furthermore, the problem definition used in this paper somewhat resembles that of the RoboFlag Drill described by [3]. They describe a scenario where a set of defenders are guarding a circular defence zone against a set of attackers. The attackers are randomly placed in an outer circle circumscribing the inner defence zone and move with constant velocity towards the zone. The goal of the defenders is to intercept as many of the incoming attacking trajectories before they reach the defence zone. It would be of great interest to test the RHGA for this scenario and compare the results with those of [3].

4.7 Concluding Remarks

This paper shows that a GA combined with RHC is able to simultaneously perform coordinated control, task assignment, and multiple target tracking in dynamically changing environments. The problem description is an interesting and non-trivial challenge for researchers in the field who are welcome to find alternative methods for solving it.

Acknowledgements. The author wishes to thank staff at the NCA and Christian Michelsen Research for providing details on oil tanker traffic and the dynamic positioning of tugs operating along the northern Norwegian coast. Collaboration with the NCA has recently been formalised and it is envisioned that a future version of the RHGA will be part of the decision support tools at VTS centres run by the NCA.

In addition, the author is grateful for the contributions of his colleagues, associate professor Siebe B. van Albada and professor Harald Yndestad at the Ålesund University College, in the early phase of this work.

References

1. Bye, R.T., van Albada, S.B., Yndestad, H.: 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. pp. 114–125. SciTePress (2010)
2. Det Norske Veritas: Rapport Nr. 2009-1016. Revisjon Nr. 01. Tiltaksanalyse — Fartsgrenser for skip som opererer i norske farvann. Tech. rep., Sjøfartsdirektoratet (2009)
3. Earl, M.G., D'andrea, R.: A decomposition approach to multi-vehicle cooperative control. Robotics and Autonomous Systems 55, 276–291 (2007)
4. Eide, M.S., Endresen, Ø., Breivik, Ø., Brude, O.W., Ellingsen, I.H., Røang, K., Hauge, J., Brett, P.O.: Prevention of oil spill from shipping by modelling of dynamic risk. Marine Pollution Bulletin 54, 1619–1633 (2007)
5. Eide, M.S., Endresen, Ø., Brett, P.O., Ervik, J.L., Røang, K.: Intelligent ship traffic monitoring for oil spill prevention: Risk based decision support building on AIS. Marine Pollution Bulletin 54, 145–148 (2007)

6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional (1989)
7. Goodwin, G.C., Graebe, S.F., Salgado, M.E.: Control System Design. Prentice Hall, New Jersey (2001)
8. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. Wiley, 2nd edn. (2004)
9. Havforskningsinstituttet: Fisken og havet, særnummer 1a-2010: Det faglige grunnlaget for oppdateringen av forvaltningsplanen for Barentshavet og havområdene utenfor Lofoten. Tech. rep., Institute of Marine Research (Havforskningsinstituttet) (2010)
10. Luo, X., Li, S., Guan, X.: Flocking algorithm with multi-target tracking for multiagent systems. Pattern Recognition Letters 31, 800–805 (2010)
11. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall, first edn. (2002)
12. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: Computer Graphics (ACM SIGGRAPH). vol. 21, pp. 25–34 (1987)
13. Rossiter, J.A.: Model-based Predictive Control. CRC Press (2004)
14. Russell, S., Norvig., P.: Artificial Intelligence: A Modern Approach. Pearson (2010)