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

Robin T. Bye, Siebe B. van Albada and Harald Yndestad

Department of Technology and Nautical Sciences, Ålesund University College, Postboks 1517, N-6025 Ålesund, Norway

Keywords: Genetic algorithm, Receding horizon predictive control, Resource allocation, Multi-target tracking, Dynamic environments.

Abstract: Combining methodologies from cybernetics and artificial intelligence (AI), we present a receding horizon genetic algorithm (RHGA) for solving the task of dynamic assignment and tracking of multiple targets. We demonstrate the capabilities of the algorithm by means of a case study on optimal positioning of tugs to reduce the risk of oil tanker drifting accidents along the northern Norwegian coast. Through simulations we show that the RHGA performs intelligent target assignment and close target tracking while constantly reevaluating its suggested solutions based on current and predicted information. We see great potential for further development and consider our RHGA and problem description a platform for further research.

1 INTRODUCTION

The task of assigning agents to the tracking of multiple targets in a dynamic environment constitutes several overlapping challenges. First, there is the problem of resource allocation: Which agents shall track which targets? If there is redundancy, that is, there are more agents than targets, each target can have at least one agent assigned to it. However, as is commonly the case, the number of targets exceeds the number of agents, meaning that agents are assigned more than one target. In addition, some targets may be more important to track than others. This poses a second problem of tracking, or dynamic positioning, of agents relative to the targets: How can the agents collectively move so that net tracking performance is maximised? Third, tracking performance can be defined as the ability to reduce a cost measure, or function. How is this cost measure defined? Finally, being in a dynamic environment, agents need to constantly reevaluate these problems: How can the agents incorporate future changes in the state space, such as motion of targets and changing dynamics of the surroundings?

In this paper we present a receding horizon genetic algorithm (RHGA) for solving the abovementioned challenges. To aid the reader and to emphasise the usefulness of such a method, we undertake a case study of a real-world example regarding the positioning of tug vessels along the coast of northern Norway. The Norwegian Coastal Administration (NCA) is in charge of a vessel traffic services (VTS) centre in the town of Vardø, which in turn administers a number of tugs. The main task of these tugs is to patrol the coastline in a manner that reduces the overall risk of oil spill resulting from drift grounding accidents involving oil tankers.

Oil tankers are required by law to sail along predefined corridors distant to the coastline, meaning that it is possible to predict a ship's future position, for example by linearly extrapolating its speed along its corridor. Furthermore, both static (identity, destination, cargo, etc.) and dynamic (speed, position, heading, etc.) ship information are constantly being transmitted through the automatic identification system (AIS) and made available to VTS centres. Together with weather forecasts and models of factors such as wind and ocean currents, this information can be used to predict potential drift trajectories and grounding positions, for example in the case of a ship losing manoeuvrability through steering or propulsion failure (Eide et al., 2007b). If a tug is close enough, it can intercept the ship's drift trajectory and tow the ship away before it runs ashore. Thus, in keeping with the termi-

T. Bye R., B. van Albada S. and 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.
DOI: 10.5220/0003087101140125
In *Proceedings of the International Conference on Evolutionary Computation* (ICEC-2010), pages 114-125
ISBN: 978-989-8425-31-7

nology in the introductory paragraph, the patrol tugs correspond to agents and the oil tankers' drift trajectories correspond to target trajectories.

The NCA has developed risk-based decision support tools based on dynamical risk models that draws on a vast pool of information (Eide et al., 2007a; Eide et al., 2007b). Some of this information is static and certain, such as a ship's type, its crew, and amount and type of oil it is carrying. Information about other factors is uncertain and based on models of wind, ocean currents, accident frequency and consequences, spill size and impact, and so on. The decision support tools are able to aid a human operator at a VTS centre in commanding tugs by pointing out high-risk positions that tugs should approach. However, as the number of tugs and oil tankers grows, the problem of allocating tugs to several positions quickly becomes non-trivial. Thus there seems to be a need of a computer program, or algorithm, that can take the existing models and decision support tools one step further by calculating where each tug should move. The calculation may be based on a number of input variables set by the operator. For example, particular oil tankers or areas may be given special priority.

The program should run in real-time, however, given that it usually takes many hours from the moment that a ship loses manoeuvrability until it runs ashore and the relatively slow dynamics of ocean currents, a moderate update rate suffices.

In the following sections, we will present a simplified version of the tug positioning problem and an algorithm that solves it by combining receding horizon control (RHC) with a genetic algorithm (GA). We will show how the RHGA performs in some simulated scenarios and finally we will discuss our results, limitations to our approach, and future potential.

1.1 Problem Description

To simplify the problem, we will assume that N_o oil tankers move in one dimension only up and down (north and south, say) an *oil tanker line of motion* y_o. This seems reasonable considering that oil tankers are required to follow predefined straight-line-segmented corridors. Closer to shore, we assume that N_p tugs are moving, or patrolling, up and down a *patrol line of motion* y_p parallel to that of the oil tankers. We ignore collisions between oil tankers and patrol tugs on their respective lines of motion. Naturally, we acknowledge that the coastline with its fjords, peninsulas, and islands does not constitute a straight line, however, given that tugs should stop drifting ships before they reach land or danger zones, it seems reasonable to simplify the problem as described. Figure 1 shows a graphical representation of the problem description.

Next, we assume the availability of an accurate

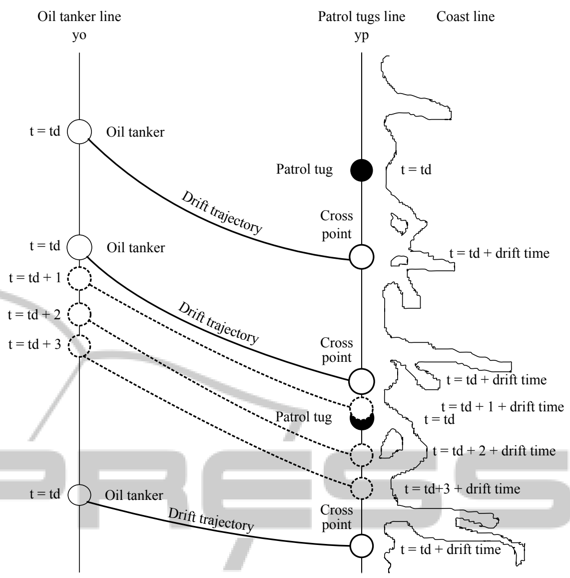


Figure 1: Problem description. Oil tankers (white circles) and patrol tugs (black circles) move in one dimension up and down (north and south, say) lines y_o and y_p, respectively. Predicted drift trajectories starting at some time $t = t_d$ with corresponding cross points on the patrol line are shown as indicated. In addition, dashed circles and lines indicate another three predicted cross points from future drift trajectories starting at $t = t_d + 1, \ldots, t_d + 3$ for the oil tanker in the middle. How should the tugs move in order to best prevent drift grounding accidents?

model, or set of models, such as those developed by the NCA and described in Section 1. These models are able to predict future positions of oil tankers along y_o and their corresponding potential drift trajectories given current and predicted information about ships (e.g., speed and direction) and the environment (e.g., wind and ocean currents). Specifically, for an oil tanker positioned at $y_o(t)$ at time $t = t_d$, the model can predict its future position $\hat{y}_o(t)$ for $t = t_d + 1, \ldots, t_d + T_h$, where T_h is the prediction horizon and we use a discrete-time formulation with a sampling time of 1 hour. For each position $\hat{y}_o(t)$ there is a corresponding predicted drift trajectory that may or may not reach the patrol line y_p at some time in the future depending on ocean currents, wind conditions, and other factors. Collecting all predicted drift trajectories for all oil tankers, the model can thus provide a distribution of *cross points* where potential drift trajectories cross the patrol line predicted a time T_h into the future.

Figure 2 shows an example scenario. Based on such a distribution, the problem is to calculate trajectories (sequences of *patrol points*) along y_p for each of the patrolling tugs in such a manner as to minimise the risk that an oil tanker in drift cannot be reached and saved before grounding. This is a difficult problem dealing both with resource allocation (which tugs shall cover which areas) and tracking (how the coverage, or tracking of cross points shall be accomplished) in a dynamic environment.

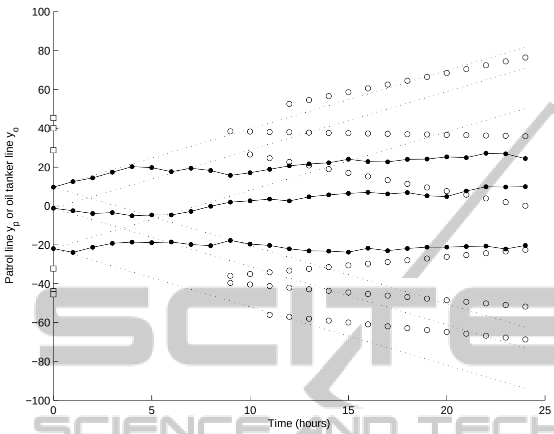


Figure 2: Example scenario. White squares indicate the positions of six oil tankers on the oil tanker line y_o at $t = 0$. White circles indicate where predicted drift trajectories cross the patrol line y_p as a function of time. Black circles indicate three random patrol trajectories on y_p for $t = 0, \ldots, T_h$, where the prediction horizon is set to $T_h = 24$ hours. The regions reachable by the three tugs at full speed from $y_p(0)$ are depicted by the dotted lines. All positions are in arbitrary units.

2 METHOD

2.1 Genetic Algorithm

For the simplest case of a single oil tanker and a single tug, classical control theory and mathematics can provide an analytical solution to solve the problem described in Section 1.1, however, this case is mostly of academic interest. Just by slightly increasing the number of oil tankers and tugs the problem becomes exceedingly difficult, cf. Figure 2, which shows three tugs and three randomly generated patrol trajectories they may follow in order to track the drift trajectories of six oil tankers. One possible approach is to examine a finite number of sets of potential patrol trajectories for the tugs and for each set evaluate some kind of cost function in order to measure 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 artificial intelligence (AI) methods. Here, we use a continuous GA based on a version as described in (Haupt and Haupt, 2004).

2.1.1 Characteristics of the GA

The GA used in this paper follows the general scheme used in most GAs (Haupt and Haupt, 2004):

1. Define a cost function and a chromosome encoding and set some GA parameters (mutation, 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. The interested reader may refer to (Haupt and Haupt, 2004) for further details on this mating process.

Specific to the problem described in this paper is the choice of cost function and chromosome encoding. These are described in detail in respective Sections 2.1.2 and 2.1.3 below.

2.1.2 Cost Function

For the results presented in this paper, we define the cost function as the sum of the distances between all cross points and the *nearest* patrol points. The reasoning behind this is that if an oil tanker in drift can be saved by a tug a certain distance away, it is irrelevant that other tugs further away can save it at a *later time*. Note that "later time" assumes that all tugs have the same maximum speed.[1] Moreover, as will be demonstrated in Section 3, choosing this cost function also yields proper task allocation, as patrol tugs will tend to spread out and track different groups of cross points, thus reducing the overall risk of grounding.

We may drop the subscript p for the patrol line y_p and define y_t^p as a patrol point on the pth tug's patrol trajectory at time t. Similarly, we may define y_t^c as a cross point on the cth oil tanker's drift trajectory at time t. For N_o oil tankers and N_p patrol tugs we can then define the cost $f(t, \mathbf{C}_i)$ as a function of time $t = t_d$ and the ith chromosome $\mathbf{C}_i$:

$$f(t, \mathbf{C}_i) = \sum_{t=t_d}^{t_d+T_h} \sum_{c=1}^{N_o} \min_{p \in P} |y_t^c - y_t^p| \tag{1}$$

for $P = \{1, \ldots, N_p\}$. $\mathbf{C}_i$ and y_t^p are defined in the following in Equations 2 and 3, respectively.

2.1.3 Chromosome Encoding

A patrol trajectory for tug p can be obtained from a sequence of piecewise constant control inputs (speeds) u_t^p in the interval $[-1, 1]$ for points in time $t = t_d + 1, \ldots, t_d + T_h$. The maximum values at -1 and 1 are equivalent to tugs going with maximum speed in the negative or positive y_p-direction, respectively. This encoding is generic as it is independent of each tug's maximum speed.

To encode N_p control trajectories as sequences u_t^p of length T_h (the prediction horizon) for each patrol tug $p \in \{1, \ldots, N_p\}$ we use chromosomes $\mathbf{C}_i$ of length $N_p \times T_h$:

$$\mathbf{C}_i = \left[u_1^1, \ldots, u_{T_h}^1, u_1^2, \ldots, u_{T_h}^2, \ldots, u_1^{N_p}, \ldots, u_{T_h}^{N_p}\right] \tag{2}$$

That is, each chromosome is a concatenation of N_p control trajectories, each consisting of T_h future control inputs.

The predicted patrol points for tug p are then obtained through linear extrapolation using the difference equation

$$y_t^p = y_{t-1}^p + u_t^p v_m^p t_s, \tag{3}$$

[1] For cases where tugs have different maximum speeds, one could define arrival time as distance divided by maximum tug speed and sum the minimum arrival times for each cross point.

where $t_s = 1$ hour is the sampling time, v_m^p is the maximum speed for the pth tug, and $t = t_d + 1, \ldots, t_d + T_h$.

2.2 Receding Horizon Control

Given a GA that performs satisfactorily by obtaining patrol trajectories that minimise our desired cost function, the human operator at a VTS centre could instruct the tugs to follow the chosen patrol trajectories in an open-loop manner and perhaps run the algorithm again after some time has passed. A better choice, however, would be to have the GA run at regular intervals, constantly incorporating new information about oil tanker positions and direction, ocean currents, weather conditions, and so forth. When new patrol trajectories have been calculated, the tugs could replace their respective patrol trajectories with the new ones. This strategy is equivalent to a receding horizon control (RHC) scheme, which is interchangeably termed model predictive control (MPC) in the literature.

In RHC, a control strategy, or optimal trajectory, that minimises some cost function is calculated a prespecified duration, or prediction horizon, into the future. However only the first portion of the strategy is implemented before another optimal trajectory is calculated based on new and predicted information available. This new trajectory replaces the old one but again only the first portion is implemented and the process then repeats.

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 (Goodwin et al., 2001). It can be shown that RHC can be designed with guaranteed asymptotic closed-loop stability (Goodwin et al., 2001) and this remarkable property is perhaps the most important reason for its popularity.

2.2.1 Constraints

An advantage of using RHC is that constraints can be handled in the design phase of a control system and not in some *post hoc* fashion after the design (Goodwin et al., 2001; Maciejowski, 2002). For the tugs, an inherent limitation is the maximum speed at which they can operate. This speed limits the size of the envelopes in Figure 2 and thus the number of reachable cross points. Using RHC combined with the GA ensures that this limitation is incorporated in the planning of tug trajectories.

2.2.2 Optimisation

A good choice of initial population for a GA allows it to quickly (with fewer iterations) find a good solution. Given that the dynamics of the simulated scenario has not changed "too much," a solution found at one RHC step should also be a viable solution at the next RHC step. This is achieved by keeping the best chromosome at one RHC step, modifying it slightly, and inserting it into the GA's initial population at the next RHC step.

The kept chromosome has to be slightly modified to accommodate planned trajectories now starting at the next time instance of the simulation. This involves a time-shift by discarding the first sample of the trajectories in the best chromosome and adding a new sample at the end, either with a random value, or, as we choose to do here, with the same value as the next-to-last sample.

2.3 Simulation Study

To implement and simulate the scenario presented above we used the technical computing software package MATLAB.[2] A number of choices had to be made about positions, speeds, and directions of ships, drift rates and directions, the GA and RHC, and general settings. After performing some preliminary testing we decided to use the settings described below.

2.3.1 Number of Ships

Based on information provided by NCA staff or affiliates and a recent report (Havforskningsinstituttet, 2010), we chose to use $N_p = 3$ tugs and $N_o = 6$ oil tankers for our simulations. Whereas these figures are realistic as of 2009, this situation will change drastically in the near future due to the development of nearby oil and gas fields (see Section 4.5).

2.3.2 Position of Ships

The initial position of ships at time $t = 0$ was varied for each simulation. Specifically, we placed the oil tankers on the oil tanker line y_o at positions drawn randomly from a uniform distribution in the range $[-50, 50]$ (arbitrary units). Likewise, we placed the patrol tugs randomly in the same range on the patrol line y_p. The reason for doing so was to compare the performance of the trajectories found by the RHGA with just keeping the patrol tugs stationary at their initial positions.

[2] MATLAB R2010a, available at http://www.mathworks.com/.

2.3.3 Velocities of Ships

Each oil tanker was initialised with a random speed in either the negative (southbound) or positive (northbound) y_o-direction and drawn from a uniform distribution in the range $[-1, 1]$ (arbitrary units). The oil tankers kept their respective speeds throughout each simulation.

The patrol tugs were assigned a maximum speed of 3 (arbitrary units) corresponding to the envelopes presented previously in Figure 2. This choice was made to ensure that tugs are much faster than oil tankers and hence allow for more dynamic task allocation (change of targets to track) and better tracking in our simulations. However, it should be kept in mind that in the real world, oil tankers have a typical operating speed of 14–15 knots (Det Norske Veritas, 2009) whereas tugs have a global average maximum speed of about 12 knots, spanning from 5–26 knots (Eide et al., 2007b).[3]

2.3.4 Drift Trajectories

While we acknowledge that ocean currents and other factors likely lead to curved drift trajectories perhaps resembling those in Figure 1, we chose to assume that any oil tanker in drift will move in a straight line perpendicular (directly east) to the patrol line and cross it after some drift time. That is, if an oil tanker loses manoeuvrability at $y_o(t_d) = x$, it will cross the patrol line at $y_p(t_d + \Delta t) = x$ after some drift time Δt.

For each oil tanker we chose a random integer drawn from a uniform distribution $[8, \ldots, 12]$ to be its drift time and kept it constant throughout each simulation. According to (Eide et al., 2007b), this choice corresponds to a situation of "fast drift," whereas "slow drift" means that most tankers will not run aground within the first 30 hours. Thus, keeping our drift times in the interval 8–12 hours is a conservative estimate and in most cases, tugs will have more time to come to the rescue of a drifting ship.

2.3.5 GA Settings

At every RHC step, we set the GA to perform $N_{iter} = 100$ iterations searching for a solution set of optimal trajectories minimising the cost function given by Equation 1. A larger number improves the solutions but it should be kept in mind that each solution set is replaced at the next RHC step. This next step also

[3] A close affiliate to the NCA recently informed us that typically, in the geographical area of this case study, the maximum speed of tugs is 15 knots and operating speed of oil tankers is 10–14 knots.

takes advantage of the predictability of the drift trajectories (straight lines) and uses a modified version of the best chromosome found in the previous RHC step (see Section 2.2.2).

The population size was set to 10 chromosomes while the mutation rate was set to 0.1. The selection parameter was set to 0.5, which implies that the best half of each population was kept for mating at each GA interation. These choices gave a good tradeoff between exploration and exploitation given the other simulation parameters described previously.

2.3.6 RHC Settings

The GA were used to find optimal trajectories with a prediction horizon of $T_h = 24$ hours for the patrol tugs. At every RHC step, only the first sample (1 hour) of these trajectories was executed by the tugs, before another solution set of trajectories was found by the GA. This process was repeated for $N_{RHC} = 26$ RHC steps (that is, each scenario was simulated for $t_d = 0$ to $t_d = 25$ hours).

2.3.7 General Settings

A total of $N_{sim} = 20$ scenarios (random initial positions, velocities, and drift times) were simulated. For each scenario, the optimal costs at each RHC step were calculated and averaged and stored in vector $\mathbf{f}_{RHGA}$ of length N_{sim}. Similarly, the costs that would occur if the patrol tugs would not move from their initial random positions (static trajectories) were calculated and averaged and stored in a vector $\mathbf{f}_{static}$ of length N_{sim}.

2.3.8 Settings Summary

The simulation settings for patrol tugs and oil tankers as well as algorithm settings for the GA, the RHC scheme, and some general settings are summarised in Table 1.

Table 1: Simulation settings.

Oil tankers	
Number of tankers N_o	6
Random initial position	$[-50, 50]$
Random velocity	$[-1, 1]$
Drift direction	East
Random drift time Δt (hours)	$[8, 9, \ldots, 12]$
Patrol tugs	
Number of tugs N_p	3
Random initial position	$[-50, 50]$
Max velocity	± 3
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 (hours)	1
Number of steps N_{RHC}	26
General settings	
Number of scenarios N_{sim}	20
Cost comparison	$\mathbf{f}_{RHGA}$, $\mathbf{f}_{static}$

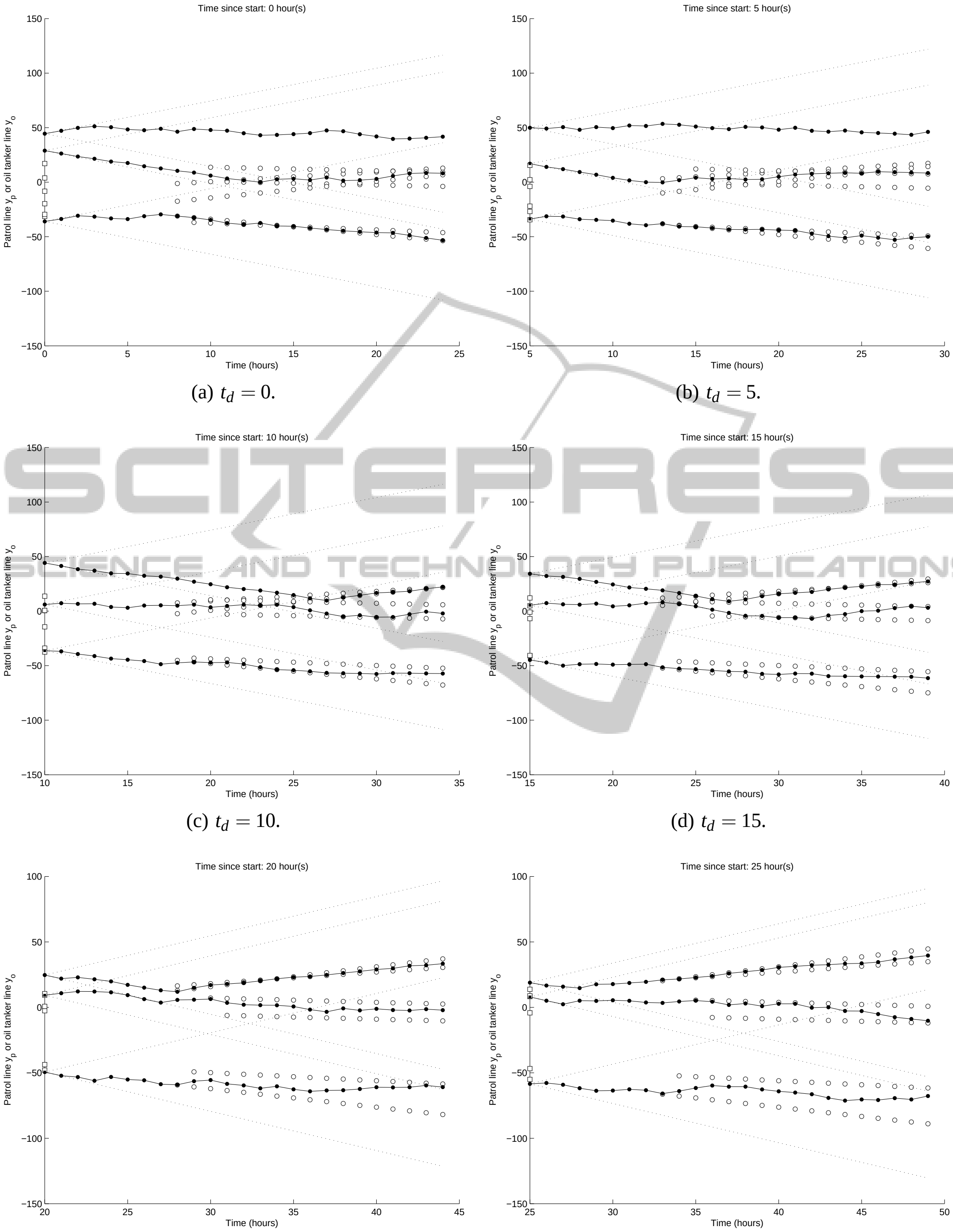
3 RESULTS

3.1 Simulation example

Figure 3 shows a simulation example using the settings given in Table 1. Initially at drift time $t_d = 0$ (Figure 3(a)), three patrol trajectories, each of duration $T_h = 24$ hours into the future, are planned for the tugs based on the predicted distributions of cross points, that is, the positions and times where drift trajectories cross the patrol line. The bottom two drift trajectories are close together at around $y_p \approx -40$, thus the RHGA allocates the first and closest tug positioned at $y_p^1 \approx -38$ to these two oil tankers by planning a patrol trajectory that minimises the sum of distances from these drift trajectories' cross points to the nearest patrol points (see Section 2.1.2). Similarly, the remaining four drift trajectories are clustered around $y_p = 0$, and the RHGA allocates the second and closest tug at $y_p^2 \approx 35$ to these four oil tankers. Finally, the third and northmost tug at $y_p^3 \approx 48$ is not assigned any oil tanker and a roughly stationary "don't care" trajectory is planned for this tug.

At the next RHC step at $t_d = 1$, each tug executes the first sample (1 hour) of their planned trajectory before a new set of patrol trajectories, again of duration $T_h = 24$ hours, is planned based on updated information about tug and oil tanker positions and predicted drift trajectories and cross points. This process repeats for $t_d = 2, 3, \ldots, 25$, with the prediction horizon constantly being shifted one sample, hence the concept of RHC.

Figure 3(b) shows the current positions of oil tankers and patrol tugs at $t_d = 5$. The tugs are allocated the same oil tankers as at $t_d = 0$, however, because some oil tankers are moving north, their predicted drift trajectories are getting closer to the third tug at $y_p^3 \approx 50$. As a result, a change in target allocation occurs in Figure 3(c), which shows a snapshot of the situation at $t_d = 10$.



(a) $t_d = 0$. (b) $t_d = 5$.

(c) $t_d = 10$. (d) $t_d = 15$.

(e) $t_d = 20$. (f) $t_d = 25$.

Figure 3: Example simulation. White squares indicate the positions of six oil tankers on the oil tanker line y_o at time of drift $t_d = 0, 5, \ldots, 25$. White circles indicates where predicted drift trajectories cross the patrol line y_p as a function of time. Black circles indicate the current positions and patrol trajectories suggested by the RHGA for three tugs on y_p for $t = t_d, \ldots, t_d + T_h$, where $T_h = 24$ hours is the prediction horizon. The regions reachable by tugs going at maximum speed from their current positions $y_p(t_d)$ are depicted by the dotted lines. All positions are in arbitrary units.

Here, the RHGA plans a trajectory for the third and northmost tug to begin tracking of the two northmost drift trajectories (which are so closely positioned that they are difficult to distinguish), another trajectory is planned for the second tug to track the two middle drift trajectories, and the first and southmost tug is still assigned the two southmost drift trajectories.

Figures 3(d)–3(f) show that this allocation scheme is maintained for the rest of the simulation: The first (southmost) tug moves south while tracking the two southmost drift trajectories, the second (middle) tug is slowly moving north in order to track the two middle drift trajectories, and the third (northmost) tug moves south to track the two northmost drift trajectories.

Notice that in the latter case for the third tug, the planned trajectories involves moving south at first to "meet" the northbound drift trajectories and then move north again while performing close tracking. If the simulations had been run for $t_d > 25$, this would have become evident in the actually executed position of the third tug (given that the oil tankers continue their predicted movements). Similarly, the trajectory for the second tug begins to tend southward at $t_d = 25$ due to the predicted drift trajectories also going south.

Quantitatively, one possible performance measure is obtained by comparing the cost of applying the RHGA-generated trajectories to that of static trajectories, that is, keeping each patrol tug p stationary at its initial position, or $y^p_{t=0} = y^p_{t=1} = \ldots = y^p_{t=T_h}$. For this simulation example, the overall RHGA cost was found by summing the costs at $t_d = 0,\ldots,25$ and finding the average cost for each RHC step, which equalled 463.7 (arbitrary distance units). The static cost was 1837.4, thus the RHGA improved the static cost by 74.8 %.

3.2 Main Study

Table 2 summarises the results from the simulation example above (simulation run 1) and 19 other simulated scenarios based on the settings presented in Table 1. For every run, the average cost at each RHC step $t_d = 0,\ldots,25$ is calculated for the static case ($\mathbf{f}_{static}$) and the RHGA case ($\mathbf{f}_{RHGA}$). Performance is measured as the reduction in cost (in percent) from applying the RHGA instead of the static strategy.

The best performance of the RHGA compared to static trajectories was obtained in simulation run 12 and showed a cost reduction of 79.0%. The worst performance happened in simulation run 2 and showed a cost reduction of 26.2%. The mean performance was a cost reduction of 65.8%.

Comparing the standard deviation of the costs for static and RHGA trajectories, RHGA showed a reduction of 70.4%. This is not surprising, considering that the static strategy can result in low cost trajectories by chance simply because the drift trajectories roughly coincide with the initial positions of tugs. Likewise, the static strategy can lead to some very costly trajectories, because the cross points are far away from the static tug positions. As a result, the cost for each simulation scenario will vary greatly depending on each scenario. The RGHA trajectories, on the other hand, are constantly being reevaluated at each sample interval, accommodating both current and predicted changes to the overall situation. This results in the RHGA trajectories very effectively tracking the drift trajectories and reducing the overall cost, which tend to vary much less than in the static case.

Table 2: Simulation results.

Simulation run	$\mathbf{f}_{static}$	$\mathbf{f}_{RHGA}$	Performance (%)
1	1837.4	463.7	74.8
2	1552.2	1145.9	26.2
3	2278.0	675.1	70.4
4	3097.3	1314.0	57.6
5	2822.3	855.8	69.7
6	3929.4	1526.9	61.1
7	2431.7	633.5	73.9
8	2877.1	880.2	69.4
9	3174.7	794.0	75.0
10	1221.5	665.2	45.5
11	3839.0	1113.4	71.0
12	4356.1	914.3	79.0
13	1921.9	818.8	57.4
14	1536.1	583.4	62.0
15	1489.2	869.5	41.6
16	1546.6	575.5	62.8
17	1456.7	457.1	68.6
18	1836.8	445.8	75.7
19	950.8	559.9	41.1
20	3068.0	874.4	71.5
Mean	2361.2	808.3	65.8
Standard dev.	984.7	291.6	70.4
Best run: 12			79.0
Worst run: 2			26.2

3.3 Conclusions

The simulation results show that the RHGA is able to simultaneously perform multi-target allocation and tracking in a dynamic environment. The GA calculates patrol trajectories for allocation and tracking at regular intervals in time. However, as the environment changes, an RHC process is employed, where the GA is constantly used to replan new trajectories based on current and predicted information.

Employing a cost function related to the distance from each cross point (where the predicted drift trajectories crosses the patrol line y_p) to the nearest predicted patrol trajectory gives good tracking but also provides task allocation "for free." In short, the patrol trajectories suggested by the RHGA yield good prevention against possible drift accidents due to taking the predicted future environment into account.

Nevertheless, this study should be considered a preliminary study only that perhaps raises more questions than it answers. These questions will be discussed in Section 4.

4 DISCUSSION

4.1 Optimisation

Whereas this paper is not concerned with algorithmic speed optimisation *per se*, an obvious point of optimisation is the choice of initial population in the GA that must be set at every step of the receding horizon planning process. Because the dynamics of the problem scenario is relatively slow-changing (e.g., ocean currents, wind, or oil tanker speeds and directions), the initial population should include good individuals, and/or their offspring, from the previous time step. This will ensure that the GA quickly can tune in on good solutions, given that the scenario has not changed too much from the previous time step. For the simulations presented here, we chose to keep a modified version of the best chromosome found in one RHC planning step and put it in the initial population of the next one as described in more detail in Section 2.2.2.

Moreover, this strategy can be further modified to reduce the overall number of GA iterations: After running a number of iterations at the first RHC step until a good solution is obtained, only a fraction of this number of iterations is needed for subsequent RHC steps. This is because the dynamics are slow-varying and the GA has "tuned in" to a good solution space where the previously found solution greatly assists the GA in finding the new, best solution.

Another modification that may speed up algorithmic conversion is to include boundary solutions corresponding to the trajectory envelope of each tug, that is, to include solutions in the population corresponding to tugs moving with maximum speed either north or south. This ensures that "hard-to-find" cross points far away from the tugs are detected and taken into considerations.

Finally, it should be kept in mind that only the best chromosome is transferred at every RHC step. The other chromosomes are still initialised randomly (or with some set to boundary solutions as suggested above). Hence, if novelties have been introduced at some RHC step, the RHGA will still be able to find good solutions (albeit requiring more iterations than for a more static scenario).

4.2 Evaluating Performance

There are several ways to evaluate the performance of an algorithm. If one is concerned with algorithm convergence speed, one may compare the time or number of iterations it takes an algorithm to find a solution with that of other algorithms. We are not particularly concerned with the rate of convergence of the RHGA for the case study investigated here. The reason for this is that the dynamic scenario is relatively slow-changing and even for a much larger number of simulated oil tankers and tugs, our algorithm will still be able to plan trajectories and update them faster than the real-world time.

To simulate one 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 an RHC step (1 hour) 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.

Another method for measuring performance of an algorithm is to investigate its ability to reduce a cost function (or conversely, increase some fitness function). Using the cost function defined by Equation 1 in Section 2.1.2, we compared the cost of applying an RHGA strategy with a static strategy where tugs are kept stationary at their initial random positions. We found that the RHGA greatly reduces the cost compared with the static strategy, however, it still remains to compare the RHGA with other intelligent algorithms for the same problem defined in this paper. Nevertheless, we believe that the results from this study are very promising and that the RHGA provides a viable method for solving problems of the kind presented here.

4.3 Cost Functions

Choosing a suitable cost function is essential for a GA to be able to solve the problem at hand. While we believe our choice of cost function in Section 2.1.2 is a very reasonable one, there are likely other choices that may be equally, or better, suited to our problem.

One possible cost measure is to count the number of nonreachable cross points from tug positions at time $t = t_d$ to $t = t_d + T_h$ ahead. Nonreachable points are easily identified as the points that lie outside the dotted lines, or envelopes, that depict patrol trajectories of tugs at maximum speed in Figure 2. Here, one of the southernmost drift trajectory's cross points is nonreachable by the southernmost tug at $y_p(0) \approx -20$. Similarly, seven of the northmost drift trajectory's cross points are nonreachable by the northmost tug at $y_p(0) \approx 10$. This count may then be repeated for envelopes starting at $t = 1, 2, \ldots, T_h - 1$ and ending at $t = T_h$ and finally integrated for all t. The goal of the GA would then be to select patrol trajectories that minimise the total number of nonreachable points, integrated over time.

Preliminary results (not presented in this paper) using a cost function involving nonreachable cross points are promising but needs to be investigated further. One major drawback is that this method is computationally much more expensive compared to using the cost function in Section 2.1.2. Another possible drawback is that the RHGA will not lead to the same close tracking of predicted drift trajectories as seen in Figure 3. The reason for this is that the algorithm will not punish patrol trajectories further away from cross points than close ones as long as the number of nonreachable crosspoints does not increase. On the other hand, this may not be a problem for the case study presented here. After all, the main goal of the tugs is to be able to stop a drifting oil tanker from running aground. Whether this is achieved by close tracking of predicted drift trajectories or by some other intelligent positioning is not as relevant, although secondary goals such as minimisation of fuel consumption could mean that close tracking be given importance.

A potential modification to the cost function in Section 2.1.2 is to include a term for the control input in order to punish excessive fuel consumption. For example, consider Figure 3(a), where the third and nortmost tug is not assigned any drift trajectory and as a consequence is given a somewhat "random" oscillatory trajectory. Instead of planning such a fuel-consuming trajectory, including a small input term in the cost function would ensure that this patrol tug was simply assigned a stationary trajectory, thus avoiding unnecessary fuel consumption.

4.4 Other Heuristics

While we have compared the performance of our algorithm with that of a static solution, it could have been informative to compare its performance with that of an algorithm using some simple heuristic method. One such method is to let each patrol tug be allocated the nearest oil tanker not already allocated to another tug and then let it track that tanker's drift trajectory. We have not included any simulations using such a heuristic in this paper. Nevertheless, we have found 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 an oil tanker, they "do not care" about the drift trajectories of the remaining oil tankers, which in turn causes evaluations of the cost function to increase.

Another simple heuristic is to spread out the tugs evenly along the patrol line. Although not simulated, we assume that this method could work reasonably well for very large numbers of tankers since the tankers' cross points would then likely be roughly uniformly distributed along the patrol line (assuming the tankers not clustering). For a small number of tankers, on the other hand, this heuristic would not be able to perform well as cross points would not very often occur close to the positions of the tugs.

4.5 Simulation Scenarios

There are numerous alternative scenarios that we could have investigated in this paper. For example, we could have lifted some of our restrictions and incorporated changing drift dynamics (e.g., use a dynamically-changing vector map for ocean currents), let patrol tugs have different maximum speeds, let oil tankers be weighted (e.g., based on their age, cargo, crew, etc.), and so on. However, most of these changes will be handled with ease by the RHGA as they only amount to scaling of variables.

More relevant is the choice of number of ships. Based on recent information in a governmental report made by the Norwegian Institute of Maritime Research (Havforskningsinstituttet, 2010), our choice of three tugs and six oil tankers represents a realistic and typical scenario as of today. Nevertheless, since these figures are calculated based on measures of yearly traffic, there will inevitably be days when the number of oil tankers is greater than six. Moreover, 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 greatly increase over the next 10–15 years. As mentioned in Section 4.2, the RHGA can easily handle the tenfold number of oil tankers while maintaining real-time requirements and thus appear well suited for much heavier traffic than that of today.

We have not simulated cases where drifting actually occurs nor the situation where a tug becomes unavailable due to refuelling at land or being busy rescuing a drifting tanker. Nor have we tried to estimate how many tugs are necessary to maintain a sufficient degree of safety for a given number of tankers, which is highly relevant considering the foreseen increase in tanker activity. We intend to investigate this further, in particular in light of other applications where resource allocation and deallocation happens even more frequently.

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. As discussed below, we intend to continue development of our RHGA to accommodate such scenarios.

4.6 Other Applications

In addition to the one-dimensional (1D) problem described in this paper, we believe that the RHGA is suitable for performing multi-target allocation and tracking in dynamic environments also in 2D and 3D. This will require modifications to the algorithm and we have already started this work. A 2D version in environments with slow dynamics will required little modifications, however, for fast dynamics and/or 3D environments, it is likely that the algorithm must be improved, for example through distributed, parallel control.

Moreover, it could be interesting to combine the RHGA with so-called boid, or flocking, rules involving cohesion, separation, and alignment (Reynolds, 1987). In a promising effort, (Luo et al., 2010) presents a flocking algorithm that modifies the flocking rules by (Reynolds, 1987) and succeeds in multi-target tracking performed by multiple agents. We envision that through further development a modified version of the RHGA could perform equally well as the algorithm used for the scenarios described by (Luo et al., 2010).

Furthermore, we note that the problem definition used in this paper somewhat resembles that of the RoboFlag Drill described by (Earl and D'andrea, 2007). 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 our RHGA for this scenario and compare our results with those of (Earl and D'andrea, 2007).

4.7 Concluding Remarks

This paper shows that a GA combined with RHC is able to perform multi-target assignment and tracking in dynamically changing environments. Whilst our results are promising, we consider our study a preliminary one that needs to be extended for more scenarios and compared with other intelligent algorithms. Nevertheless, we believe that our problem description is an interesting and non-trivial challenge for researchers in the field and welcome alternative attempts at solving it. Building on the work presented here, we will continue to refine and extend our RHGA to other problem scenarios.

Whereas classical control theory and mathematics can aid in solving assignment and tracking problems, we see tremendous potential for artificial intelligence (AI) methods in problems where analytical solutions become non-existent or at least very hard to find. Such problems may involve nonlinearities and constraints that AI methods solve with ease. We firmly believe that the fusion between control systems theory and AI is the way forward for solving difficult multi-agent, multi-target problems.

ACKNOWLEDGEMENTS

We wish to thank Jon Leon Ervik at the Norwegian Coastal Administration and Kjell Røang at Christian

Michelsen Research for providing useful information about oil tanker traffic and positioning of tugs along the northern Norwegian coast.

REFERENCES

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

Earl, M. G. and D'andrea, R. (2007). A decomposition approach to multi-vehicle cooperative control. *Robotics and Autonomous Systems*, 55:276–291.

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

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

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

Haupt, R. L. and Haupt, S. E. (2004). *Practical Genetic Algorithms*. Wiley, second edition.

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).

Luo, X., Li, S., and Guan, X. (2010). Flocking algorithm with multi-target tracking for multi-agent systems. *Pattern Recognition Letters*, 31:800–805.

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

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Computer Graphics (ACM SIGGRAPH)*, volume 21, pages 25–34.