CHALMERS

Multi-objective optimisation of ship routes

Master's thesis in Complex Adaptive Systems

ANGELICA ANDERSSON

Department of Applied Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

MASTER'S THESIS IN COMPLEX ADAPTIVE SYSTEMS

Multi-objective optimisation of ship routes

ANGELICA ANDERSSON

Department of Applied Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

Multi-objective optimisation of ship routes
ANGELICA ANDERSSON

ISSN 1652-8557
Department of Applied Physics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Bulk carrier vessel © ABB. Reproduced with permission.

Chalmers Reproservice
Göteborg, Sweden 2015

Multi-objective optimisation of ship routes
Master's thesis in Complex Adaptive Systems
ANGELICA ANDERSSON
Department of Applied Physics
Chalmers University of Technology

Abstract

In this master thesis two different approaches of solving a three-criteria multi-objective ship route optimisation are developed and compared. The first is a grid search approach, while the second one is a modification of the distance based Pareto genetic algorithm, which has previously been proven useful in other multi-objective optimisation problems. It is found that the modified distance based Pareto genetic algorithm can give an equivalently good result using approximately 1% of the computing time, but also that a penalty needs to be introduced in the second method in order for it to be used in an actual product.

Keywords: Ship route optimisation, Multi-objective optimisation, Weather routing, Voyage planning, Genetic algorithms

Acknowledgements

First of all I would like to thank my supervisor Mats Molander without whose support, inspiring discussions and useful advice this thesis would not have been possible. I would also like to thank all of ABB Corporate Research, in particular Rickard Lindkvist and Shiva Sander-Tavallaey for letting me be a part of the company, and to Kateryna Mishchenko who gave me useful advice about parallel programming.

A special thanks to my examiner Mattias Wahde for his tireless proof-reading of my report and for his good advice.

A warm thanks also to the other thesis workers at ABB Corporate research for their friendship and to Mikael Frisk for bringing home made cookies almost every week, it was truly inspirational.

Finally I would like to thank my family and friends for their support through my studies and throughout this thesis.

Angelica Andersson, Västerås, May 2015

Contents

Chapter 1

Introduction

1.1 Theory and problem description

Voyage planning was traditionally performed manually using nautical charts together with information about weather predictions and current data. In recent years there has been a shift towards using computers to aid the planning of the route. Additional computing power together with more detailed weather predictions makes it possible to move from simply choosing a decent route into instead being able to optimise the route with respect to a wanted criterion. Most existing voyage optimisation tools optimise with respect to one criterion at a time, treating other relevant criteria as constraints.

This thesis aims to develop a a three-criteria multi-objective optimisation tool in Matlab for optimisation of ship routes, also referred to as voyage planning or weather routing. The objectives considered are travel time, fuel consumption and wave height (which is a simple criteria to ensure safety). A range of trade-off solutions between the criteria are found using weather and current data as inputs. Two different methods of multi-objective optimisation are implemented and compared. The first method is a grid search approach, which is based on an extension of a previously developed single objective optimisation framework for voyage planning at ABB. The second method is a modification of the distance based Pareto genetic algorithm.

1.1.1 Background and motivation

Between 1990 and 2013 the worldwide maritime trade more than doubled, with total volumes in 2013 reaching nearly 9.6 billion tons [1]. Although total fuel consumption depends on ship size and chosen speed, most container vessels consume somewhere between 50-350 tons of oil per day [2]. As a consequence, choosing a suboptimal route and cruising speed can have a great negative effect both in terms of environmental impact and in terms of fuel costs.

Estimates of how many shipping containers are lost at sea each year range from about 1 500 to 10 000 [3]. This, in combination with safety considerations for the crew, makes it important to also take weather conditions into consideration when selecting an optimal route.

1.1.2 Multi-objective optimisation

Multi-objective optimisation (MOO) differs from single objective optimisation (SOO) in the sense that while a problem involving SOO generally has only one global optimum, the MOO has a large number of trade-off solutions. The trade-off solutions arise due to the fact that the different objectives are conflicting with one another. For example if a robot should perform a task as fast as possible, but at the same time accelerate as little as possible for safety and life length reasons, then a trade-off has to be made between the two objectives.

1.1.3 Pareto optimality

Since it is not evident which of the trade-off solutions should be chosen, the concept of Pareto optimality is introduced. In order to define Pareto optimality it is first necessary to define the concept of dominated solutions.

Let $\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), ...f_n(\vec{x}))$ be the set of functions to optimise, i.e. the objective functions, and let $\vec{x}_1$ and $\vec{x}_2$ be two different solutions of the feasible set. Then (for the case of minimisation) $\vec{x}_1$ dominates $\vec{x}_2$ if:

$$\begin{aligned} &f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \quad \forall\, i \\ &f_i(\vec{x}_1) < f_i(\vec{x}_2) \quad \text{for at least one } i \end{aligned} \tag{1.1}$$

A solution is Pareto optimal if no other feasible solution dominates it. In the case of maximisation the inequality sign is simply switched.

1.1.4 Ideal and nadir point

The ideal point is defined as the best possible value of each objective function if each criterion is considered individually. In most cases this point is not part of the feasible set. The nadir point is instead the individual worst value of each objective function when evaluated on the Pareto front. The ideal and nadir point will be used for scaling of the objective functions in this thesis.

1.2 Scope and limitations

The idea of the final product is that the end user specifies the relative importance of the different criteria. The Matlab framework should then directly give a suggested route, with suggested speeds throughout the whole trip. The main focus of this thesis is the underlying algorithm that finds the possible optimal routes. Creating a user interface is of less importance.

Weather conditions for ships can in general be described in terms of wave height, wave direction, wave frequency, wind speed and wind direction. In this thesis only dominant wave height, dominant wave direction and dominant wave frequency will be considered to affect the parameters to be optimised. This is since they have a larger impact on travel time, fuel consumption and safety than wind speed and wind direction. Even if it might be useful to include wind predictions at a later stage, neglecting it does not greatly influence the design of the optimisation algorithm. The fact that weather reports are in fact stochastic will not be taken into account in this thesis.

The results presented in this thesis are based on only one specific dataset. This is due to the limitations of the second optimisaton routine together with the limited available datasets. The consequence of this is that it is difficult to distinguish between general conclusions of the algorithms and conclusions specific to the presented dataset.

Chapter 2

Related work

2.1 Classical approaches

Weather routing has been done in a number of different ways. This section mentions some of the classical approaches to weather routing.

2.1.1 Isochrone method

The Isochrone method, developed by James [5], is one of the earliest methods used for weather routing. It is only optimised with respect to time, and then an acceptable solution for minimal fuel consumption is simply approximated. It is based on the idea of isochrones (i.e. time fronts). Given a certain propeller speed and a set of weather conditions, an isochrone is the set of points that can be reached in a predefined time step under the given conditions (see Figure 2.1). The route passing though the smallest number of time fronts is then taken as the fastest route. By adjusting the propeller speed of the fastest route and instead using the speed that makes the arrival fall on the isochrone representing the scheduled time of arrival, an approximate minimal fuel solution can be found. This is applicable when the aim is to arrive to port within a specific time slot. An improvement of the Isochrone method was developed by Hagiwara in 1989 [6]. Hagiwara attempts to solve the main problem in [5], which is that for a given point on an isochrone the perpendicular to the isochrone is not always the optimal direction.

2.1.2 Isopone method

The Isopone method is an extension of the Isochrone method and was developed in 1992 by Klompstra [7]. The main difference from the Isochrone method is that the Isopone method considers a specified amount of fuel, instead of a specified time step. This makes it possible to also optimise directly with respect to fuel consumption, instead of indirectly as in the Isochrone method. It should be noted that an invertible relationship between fuel consumption and speed must be known in order to use the Isopone method.

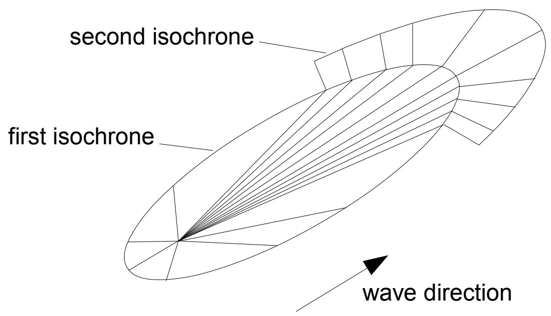

Figure 2.1: *Isochrones (time fronts) of the Isochrone method.*

2.1.3 Calculus of variation

Optimal control, which is a special case of calculus of variation, can be used for simple route optimisation problems. An optimal control problem can be described as:

$$\begin{aligned} &\min_{\vec{u}} J = \varphi(x(t_f), t_f) + \int_{t_0}^{t_1} L(x(t), \vec{u}(t), t)dt \\ &\frac{dx}{dt} = f(x(t), \vec{u}(t), t) \end{aligned} \tag{2.1}$$

where L is the constant 1 in the case of time minimization, and the rate of fuel consumption in the case of fuel minimization. The function φ is a penalty put on the arrival. Depending on the problem it can either give a penalty for late arrival, or for not arriving within a certain time slot. For the minimisation of time, the manipulated variable $\vec{u}$ could be the heading angle, and for fuel minimization it could be the heading and the velocity.

In order for (2.1) to have a local minimum the so called Euler-Lagrange equations must also be fulfilled:

$$\begin{aligned} \frac{d\lambda}{dt} &= -\left(\frac{\partial f}{\partial x}\right)^T - \left(\frac{\partial L}{\partial x}\right)^T \\ 0 &= \left(\frac{\partial f}{\partial u}\right)^T \lambda + \left(\frac{\partial L}{\partial u}\right)^T \end{aligned} \tag{2.2}$$

where $\lambda(t_f) = \frac{\partial \varphi}{\partial x(t_f)}$.

Haltiner [8] and Bleick and Faulkner [9] solves the time minimisation problem by numerically solving the Euler-Lagrange equations (they assume that the state-derivative function f is a known function of position [8] and time [9]).

Bijlsma [10] solved the minimisation problem in 1975 by solving the Euler-Lagrange equations with the destination position as a parameter. The method is built on the idea that a family of time optimal paths is created (where the end point of each branch in the family lands on the time front corresponding to the isochrone in the Isochrone method). The first branch of the family to reach the real destination is taken as the optimal solution. Since then the same author has extended his work to a more general computation method based on optimal control [11], as well as a methods based on dynamic programming [12], [13] which will be described in the next section.

2.1.4 Dynamic programming

Dynamic programming (see Bryson and Ho [14] for a detailed description of the original formulation) has been applied in several different ways to solve MOO problems within the field of voyage planning.

Zoppoli [15] was one of the first to use dynamic programming (DP) for weather routing. The main objective was to minimise travel time i.e. to solve an SOO problem, rather than to solve an MOO problem. The route between start and end point was divided into different layers, with each layer containing only a discrete set of possible positions. Each chosen route in the discrete grid is then associated with a certain cost, and the total cost is then the quantity that should be minimised. See Figure 2.2 for a visualisation of this. The cost takes into account both the chosen route and weather predictions.

By demanding that any partial sum of the cost must be the minimum cost for that part of the journey, it is possible to minimise the cost recursively. This property is known as Bellman's principle of optimality (see Bellman [16]). It is possible to use both a forward recursive algorithm and a backward recursive algorithm. In this thesis the forward version is considered.

A more recent work on voyage planning was made by Wei and Zhou in 2012 [17]. A recursive forward algorithm is used to minimise fuel consumption. The heading and velocity of the ship are the variables that determine the route. The shortest path is used as a reference to find possible alternative routes which hopefully have a lower fuel consumption.

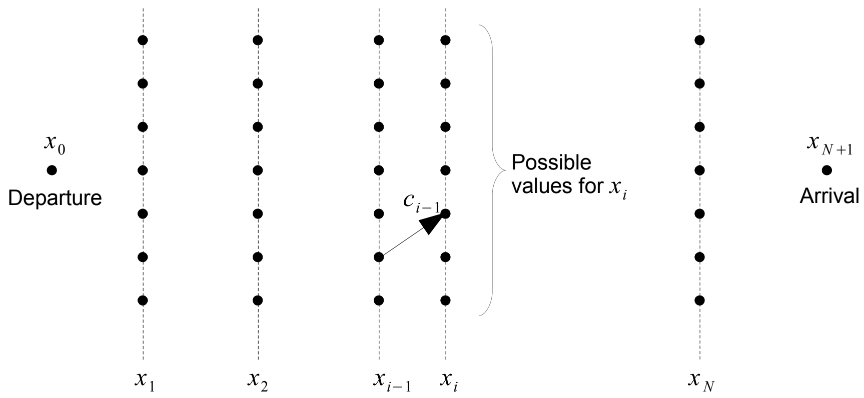

Figure 2.2: *The route divided into steps for Dynamic programming. x_i denotes the position and c_i the cost.*

2.1.5 Graph Algorithms

Much work has been done in the field of graph algorithms. Here only the part of the research field which has been previously applied in weather routing will be presented.

Graph algorithms are algorithms for finding the shortest path between nodes in a graph. One important contribution to the field is the A* method [18]. This method follows the path of the lowest expected total cost, and keeps a priority queue of alternative paths. The cost is the sum of the cost for arriving at the current node $g(x)$ and the expected remaining cost for arriving at the final destination $h(x)$. $h(x)$ must be admissible heuristic, that is it must not overestimate the cost to the final destination. In the special case where $h(x) = 0$ for all x, the A* method is equivalent to the previously developed Dijkstra's algorithm [19]. Padhy [20] based the solution of a weather routing problem on creating a grid, with each connection in the grid associated with a certain weight, representing the cost function. The best way is then found using Dijkstra's algorithm.

2.2 Evolutionary algorithms

Successful attempts have been made to use evolutionary algorithms (EA) to solve MOO problems. EAs are inspired by biological evolution, they use mechanisms such as selection, mutation and recombination to find an optimal solution. Different candidate solutions are referred to as individuals. Each individual is given a certain fitness, reflecting how well the solution solves the optimisation problem. In the process of selection, individuals with a high fitness value have a higher probability of being picked to form the basis of the new generation, which is commonly formed after recombination and mutation of the picked solutions. The process is then repeated generation after generation until the solution is considered to be good enough.

The first attempt to use EAs for multi-objective optimisation was made by Schaffer in 1984 [21]. He used a subclass of EAs called genetic algorithms (abbreviated GA, see Holland [22]) in which the variables are stored in strings of digits of fixed length, referred to as chromosomes. Schaffer made only a small modification to a GA used for SOO; however his work did not initially receive much attention since it tended to converge towards one solution when given enough time.

In 1989 Goldberg suggested in his book [23] that domination could be used in EAs to find multiple trade-off solutions. Goldberg's book became the inspiration for several different versions of EAs suitable for MOO problems. For example multi-objective GAs (MOGAs) [24], niched Pareto GAs (NPGAs) [25] and non-dominated sorting GAs (NSGAs) [26] which all differ in the way fitness is assigned to the candidate solutions.

The implementation of elitism has proven to be important for the convergence of multi-objective evolutionary algorithms. An elite in the context of evolutionary algorithms is a solution that is judged to be so good that it is chosen to be saved in its current form to the next generation (i.e. it is not subject to mutation or crossover). Elitism can be implemented in different ways in an evolutionary algorithm. Rudolph [27] proved that GAs converge to the global solution of some functions when elitism is used. Some examples of multi-objective evolutionary algorithms using elitism in different ways are the elitist non-dominated sorting GA (termed NSGA-II) [28], [29], the distance based Pareto genetic algorithm (DPGA) [30], and the strength

Pareto evolutionary algorithm (SPEA) [38].

It is common, in multi-objective evolutionary algorithms to include some mechanism to assign higher fitness to those solutions of the Pareto front which are not close to any other solutions on the Pareto front. This gives an advantage to this type of algorithm since (given enough iterations) the solutions will be spread out on the Pareto front, which is useful in most practical applications of MOO.

2.2.1 Evolutionary algorithms in voyage planning

Hinnenthal [32] uses evolutionary algorithms to optimise with respect to travel time and fuel consumption. The route and the velocity profile are both parametrized as splines. By using the Simplex method of Nelder and Mead [33] as a benchmark, Hinnenthal showed that GAs had a potential for dealing with MOO weather routing. Hinnenthal also takes into account that weather predictions are in fact probabilistic. To handle this fact ensemble weather forecasts are used as input.

Maki et al. [34] uses a real valued GA to perform weather routing. Apart from fuel consumption, Maki et al. also introduces parametric rolling as an objective function. Parametric rolling is a phenomenon limited to container ships and car or truck carriers. It can cause the ship serious damage and loss of cargo since it can result in roll angles as great as $35^\circ - 40^\circ$ even in relatively calm waters [35] (parametric rolling is also considered in [32], but only as a constraint).

Marie and Courteille [36] also uses a multi-objective GA, in this case to minimise fuel consumption in a limited or optimum time. A discrete grid based on spherical rhombus is used to automatically generate a mesh suitable for the problem.

Szlapczynska [37] uses the Strength Pareto evolutionary algorithm presented in [38], instead of the more commonly used multi-objective GA. The motivation for this is that the former is considered to be more robust [37]. Instead of giving a Pareto front as the final solution, a single route most suitable for the decision maker is selected by the algorithm. This is made possible by an additional multi criteria ranking method provided the decision maker's preferences.

Cheng and Tsou [39] uses a GA in combination with an ant colony algorithm (abbreviated ACA, see Dorigo [40]) to minimise travel time and fuel consumption while avoiding rough sea as much as possible. The ACA is inspired by the way ants emit pheromones to collectively find the shortest route to a food source. It is commonly used to solve travelling salesman problems. They found that it was possible to combine a GA with an ACA to perform weather routing, but also that the convergence was very sensitive to the chosen parameters of the ACA.

Chapter 3

Methods

3.1 Surrounding framework

The first method developed in this thesis is integrated with an already existing framework for single objective optimisation. The previous work includes generating a discrete 3D-grid from which the search for the optimal solutions begins. The x-y plane of the grid corresponds to the longitude and latitude positions in space, and the z coordinate represent the time at the node, see Figure 3.1.

The previously existing code also includes routines for importing weather data. Example data for different trips have been provided by ABB for the thesis. There is also code for modelling the ship's response to different wave spectra, as well as the fuel needed to travel at a certain speed. This will not be modified further in this thesis. Finally the previous code also performs checks whether it is physically possible to move between two points, given the positions, start and arrival times, capacity of the ship engine and current and wave data.

3.2 Grid search approach

3.2.1 Basic Algorithm

The grid search approach is based on the previously existing 3D-grid (see Figure 3.1), and is inspired by the method presented in [41]. A schematic view of the algorithm is given in Figure 3.2. Given a stage i in the grid, the algorithm loops through all feasible nodes at the stage. For each possible arrival time to the current node, a feasibility check is performed on all points at stage $i-1$ which have previously been judged as optimal routes from the initial starting point to stage $i-1$. Information about node indices in the grid and the incremental increase of each objective function are stored in a matrix for all route segments that are judged to be feasible. The feasible continuations of the routes that are optimal until stage $i-1$ are then compared (with respect to the value of the objective function for the total route between the starting position and current node) using

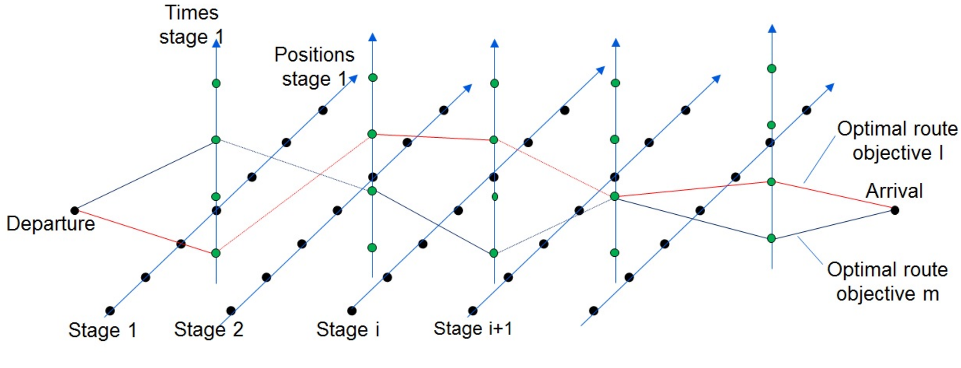

Figure 3.1: *3D search grid with two example routes. The x-y plane corresponds to the longitude and latitude positions in space, and the z coordinate represents time.*

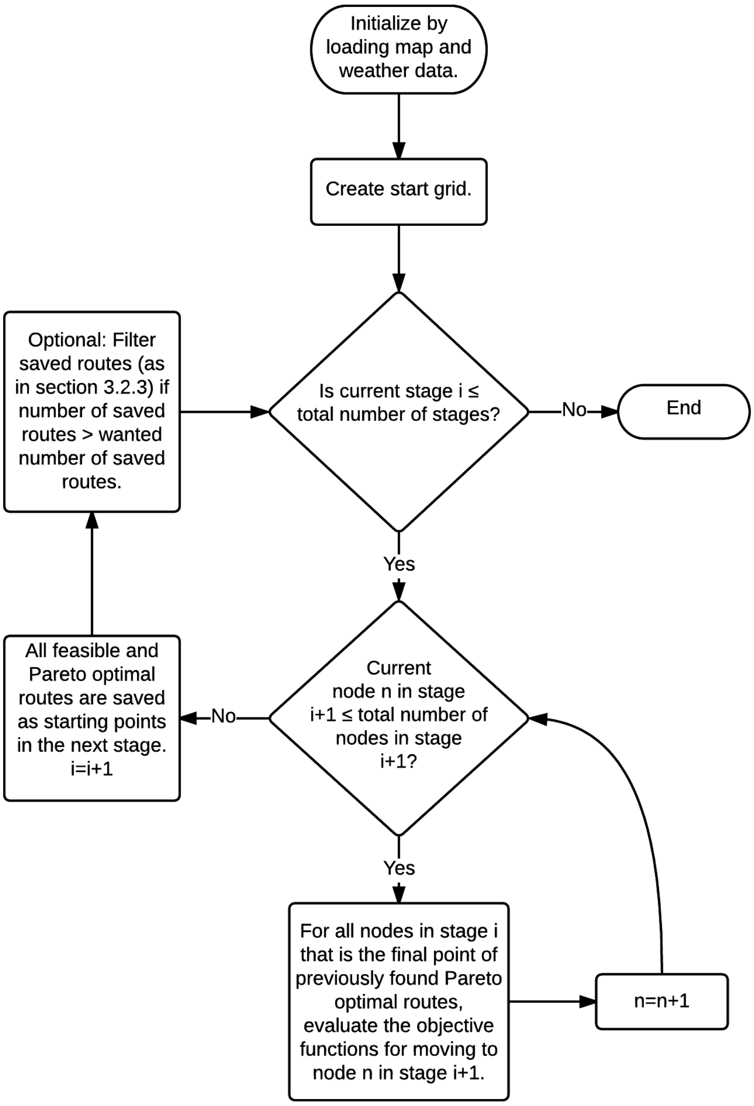

Figure 3.2: *Schematic view of the grid search approach.*

Pareto optimality (see Section 1.1.3). The solutions that are Pareto optimal are stored in another matrix, which contains grid indices and incremental increases of the objective functions for each computed stage of the route. The Pareto optimal route segments found in stage i are then used as starting points when moving on to stage $i + 1$.

3.2.2 Evaluation of objective functions

All objective functions are evaluated using weather data and ship models provided by ABB. The objective function for the time between two nodes is simply chosen with the constraint that it must be a feasible time difference between the start and end node (i.e. it cannot be faster than the maximum of what the ship engine can handle). The objective function for the wave minimisation only express the maximum wave height encountered during the whole trip. The objective function for the fuel consumption is calculated using the chosen time difference between the nodes, the chosen ship model, and wave and current data for the relevant location in time and space.

3.2.3 Optional filtering of routes

The algorithm described in Section 3.2.1 works well for a short trip (for example from Gothenburg to le Havre) or for a very sparse grid. However for a longer trip with a useful resolution of the grid the number of computations needed becomes too large to be useful. For this reason an optional filtering to only keep a maximum number of routes per Pareto front is included. Since it is desirable to have non-clustered solutions on the Pareto front, the filtering routine removes the solutions that are the most clustered. In order for the solutions to be compared in a meaningful way the objective functions are first scaled (as in [42]):

$$\bar{f}_i(x) = \frac{f_i(x) - z_i^I}{z_i^N - z_i^I} \tag{3.1}$$

where $\bar{f}_i(x)$ is the scaled objective function i, $f_i(x)$ is objective function i, z_i^N is the nadir point of the set and z_i^I is the ideal point of the set (see Section 1.1.4). The algorithm first finds the two points with the minimum Euclidean distance of the scaled values of the objective functions. Then the one with the second smallest distance out of the two is removed. This is done until there are only as many points as the chosen maximum left.

3.3 Evolutionary algorithm approach

The evolutionary algorithm approach is to a large extent based on the distance-based Pareto genetic algorithm (DPGA) developed by Osyczka and Kundu [43]. There are two main modifications made to the DPGA in this thesis. Firstly, the DPGA uses a separate elite set with an unlimited size. In the problem presented in this thesis there is a large number of points on the Pareto front. As a consequence the elite set quickly grows very large. Since the computational time of the algorithm depends on the number of elite set members an upper boundary is set on the number of elites as an addition to the original DPGA. The elite set is filtered in each generational step to obtain a fixed maximum number of individuals. The filtering attempts to spread the solutions on the Pareto front. It works as the filtering described in Section 3.2.3.

Secondly, in the original DPGA selection, crossover and mutation are used as evolutionary operators. In the case of route optimization it would be necessary to find two routes which are at the same location at the same time for a conventional crossover operation to work while assuring continuity. This task was judged to be outside the scope of this thesis and consequently only selection and mutation are used as evolutionary operators.

3.3.1 Basic algorithm

A schematic view of the evolutionary algorithm approach can be seen in Figure 3.3. The initial population is generated using the same grid as in the grid search approach. During the optimisation the solutions are fixed in the original stages (east-west direction) but not in the north-south direction nor in time. In this sense it is similar to the work done in [34].

In this thesis there are two decision variables subject to optimisation. The latitude in each stage of the solution, and the speed through water in each stage. Note that the speed through water is the speed with

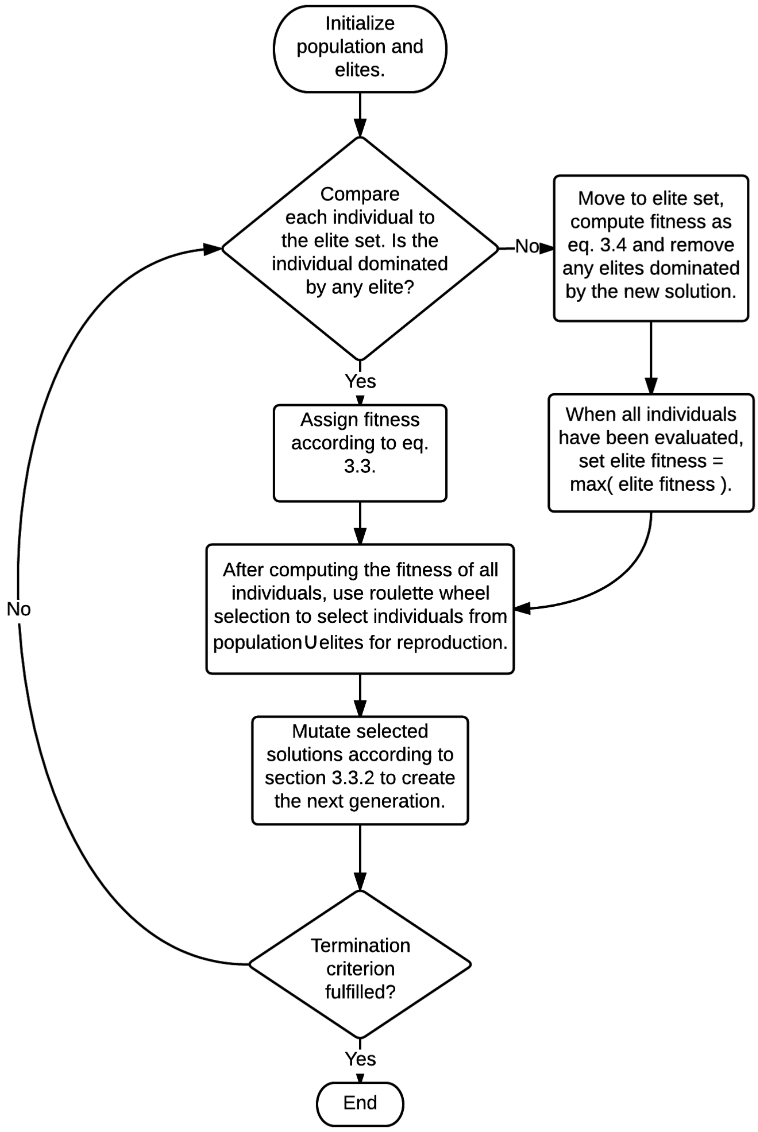

Figure 3.3: *Schematic view of the evolutionary algorithm approach.*

which the water flows past the ship as a result of both currents and the engine propulsion. The conventional speed of traversing between two points on a map is referred to as speed over ground.

The elite population is initialised to the three single objective optima found by the previously existing optimisation framework. Each individual in the standard population is then compared to each of the elites with respect to the values of the objective functions. Fitness is assigned both depending on how good the values of the objective functions are, and on the distance $d_j^{(k)}$ from individual j to elite member k in the objective space, calculated as:

$$d_j^{(k)} = \sqrt{\sum_{m=1}^{M} \left(\frac{e_m^{(k)} - f_m^{(j)}}{e_m^{(k)}} \right)^2} \tag{3.2}$$

where e and f refer to the objective function values of the elite and normal population respectively, and M is the number of objective functions. If the individual is dominated (see Section 1.1.3) by any elite with respect to the objective functions, then the fitness of the individual is assigned as:

$$F_j = \max[0, F(e^{(k_j^*)}) - d_j^{\min}] \tag{3.3}$$

If it is not dominated by any elite it is moved to the elite set and the fitness is instead computed as:

$$F_j = F(e^{(k_j^*)}) + d_j^{\min} \tag{3.4}$$

Here $d_j^{\min}$ is defined as:

$$d_j^{\min} = \min_{k=1}^{|E_t|} d_j^{(k)} \tag{3.5}$$

where $|E_t|$ denotes the elite set and $k_j^* = \{k : d_j^{(k)} = d_j^{\min}\}$. Since the only input the fitness function gets from the objective function is whether or not the point is on the Pareto front, the fitness will be a positive scalar regardless of if the problem considered is a minimisation or a maximisation problem. As a consequence of Equation (3.3) and Equation (3.4) a point on the Pareto front is highly valued if it is far away from other points on the Pareto front, while a point which is not on the Pareto front is highly valued if it is close to the Pareto front. This helps steering the population towards a Pareto front with maximal spread.

Any old elites dominated by the new elite member are removed from the elite set. When the fitness of all individuals have been computed, the fitness of all elites is set to the maximum fitness of the elites, since any point on the Pareto front should not be valued higher than any other point on the Pareto front.

In this thesis roulette wheel selection is used to select individuals to form the next generation. In roulette wheel selection each individual has a probability of being chosen proportional to its fitness value. This can be visualised as a roulette wheel with slices proportional to the different fitness values, hence the name. The selected individuals are then subject to mutation (described in more detail in Section 3.3.2) with a mutation probability $p_{\mathrm{mut}} = 1/(s-2)$ for each stage of the route, where s is the total number of stages. In evolutionary algorithms in general there is always a trade-off between exploration vs exploitation, that is between exploring as much of the objective function space as possible and giving the algorithm a good chance of converging. In this case $p_{\mathrm{mut}} = 1/(s-2)$ resulted in a good trade-off between exploration and exploitation, which is why it was chosen.

If the number of elite members exceeds the maximum they are filtered as described in Section 3.2.3. If the termination criterion (here set to be a maximum time limit for comparison with the grid search method) is fulfilled the algorithm is done, otherwise the process is repeated from computing the fitness of the new population.

3.3.2 Mutation operator

In order to have a chance of generating feasible new solutions, creep mutation is implemented. A creep mutation is a small modification of the current solution which is kept close to the current solution by using some probability distribution with the current value as mean value. In this case an Irwin–Hall distribution (see [44] and [45]), which is a 12-section eleventh-order polynomial approximation to the normal distribution is used. If the latitude of a stage in the route is selected for mutation the latitude is changed to its new value after checking that the new route does not cross land. If the new latitude results in a route that crosses land a new latitude that does not cross land is found by incrementally alternating between increasing and decreasing latitudes.

The speed through water to the stage is also changed within the suggested speed boundaries using a uniform random distribution. In order to maintain continuity throughout the route it is demanded that the time at the stage before the mutation stage, as well as the time at the stage after the mutation stage are both kept the same before and after the mutation. This means that the speed both to and from the mutation stage potentially needs to change in order to fulfil the time demand.

After the new latitude and speed to and from the mutated stage of the route have been determined, the objective functions to and from the mutated stage are updated using the new latitude and velocities.

Chapter 4

Result

The following results were obtained using an example trip from St. Johns in Canada to Bodø in Norway with weather data from July, 3$^{\text{rd}}$ 2014. The start and destination ports were chosen such that the trip would be mostly in the east-west direction, rather than the north-south direction, because of how the coordinate system and decision variables are defined in the second method.

4.1 Grid search approach

In a potential weather routing product the output to the user would be the location of the route on the map and the velocities at different times during the journey. In order to get a specific route suggestion as output the user would first need to specify how much he or she cares about the different criteria. Figure 4.1 shows the suggested locations and Figure 4.2 shows velocities for all Pareto optimal solutions found by the algorithm. That is, all possible Pareto optimal choices before specification of the relative importance of the different criteria. Figure 4.3 shows the resulting routes when choosing to consider only a single objective at a time, together with the compromise solution singled out in the Pareto front of Figure 4.11. Figure 4.4 shows the suggested speeds for the same. Running the optimisation in parallel on 16 Matlab workers took 13068 seconds (3 hours and 38 minutes) and generated in total 896 Pareto optimal solutions. The 16 threads were run on two Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz processors with 8 cores in each processor.

Figure 4.5 shows the fuel consumption as a function of time for the same four routes as in Figure 4.4.

4.2 Evolutionary algorithm approach

For the following result, the runtime was set to 13068 seconds and the maximum number of elites was chosen to 896 in order to give a fair comparison with the grid search method. It should be noted though that the comparison is still not completely fair since the genetic algorithm only used one Matlab worker for the optimisation.

Figure 4.6 shows the best, average and worst values of the fitness function plotted against the generation number. Figure 4.7 shows the suggested locations for all Pareto optimal solutions found by the genetic algorithm. The equivalent of Figure 4.2 for the second method is too cluttered to be meaningful to watch, but it confirms that there is a large number of different suggested speeds. Figure 4.8 shows the optimal routes when choosing to consider only a single objective at a time, together with the compromise solution singled out in the Pareto front of Figure 4.11. Figure 4.9 shows the optimal velocities for the same routes as in Figure 4.8. Figure 4.10 shows the fuel consumption as a function of time for the same four routes as in Figure 4.8.

4.3 Comparison of the Pareto fronts of the two methods

An important factor to consider in the case of multi-objective optimization is the final spread of the different solutions along the Pareto front. This is because the aim is for the user to be able to obtain a suggested route very close to the specified relative preferences, regardless of which preferences the user has. Figure 4.11, Figure 4.12 and Figure 4.13 shows the final Pareto fronts of the two methods.

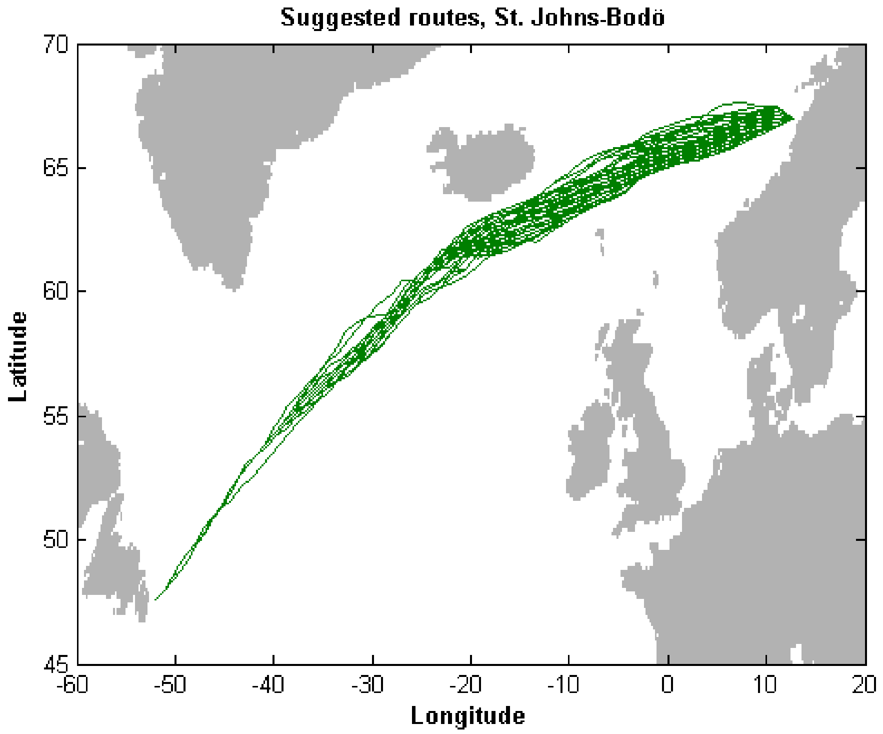

Figure 4.1: *Output of the grid search approach from St. Johns to Bodö with a maximum of three Pareto optimal routes per node in the grid. The plot shows all found Pareto optimal routes in space.*

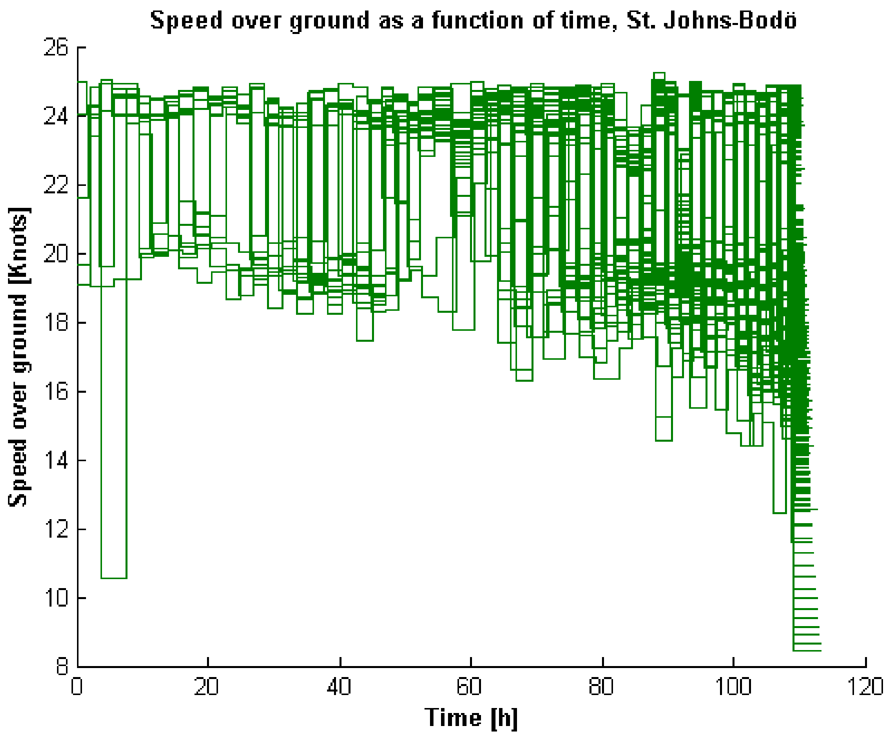

Figure 4.2: *Output of the grid search approach from St. Johns to Bodö with a maximum of three Pareto optimal routes per node in the grid. The plot shows the suggested speeds at different times of the route.*

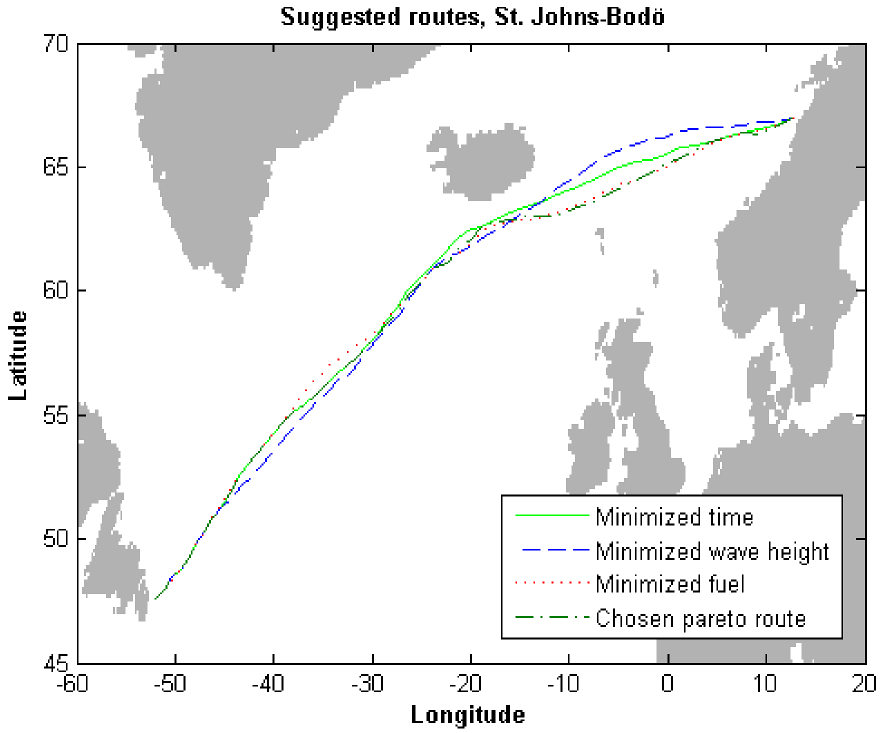

Figure 4.3: *Output of the grid search approach from St. Johns to Bodö with a maximum of three Pareto optimal routes per node in the grid when choosing to consider only a single objective at a time, together with the compromise solution singled out in the Pareto front of Figure 4.11.*

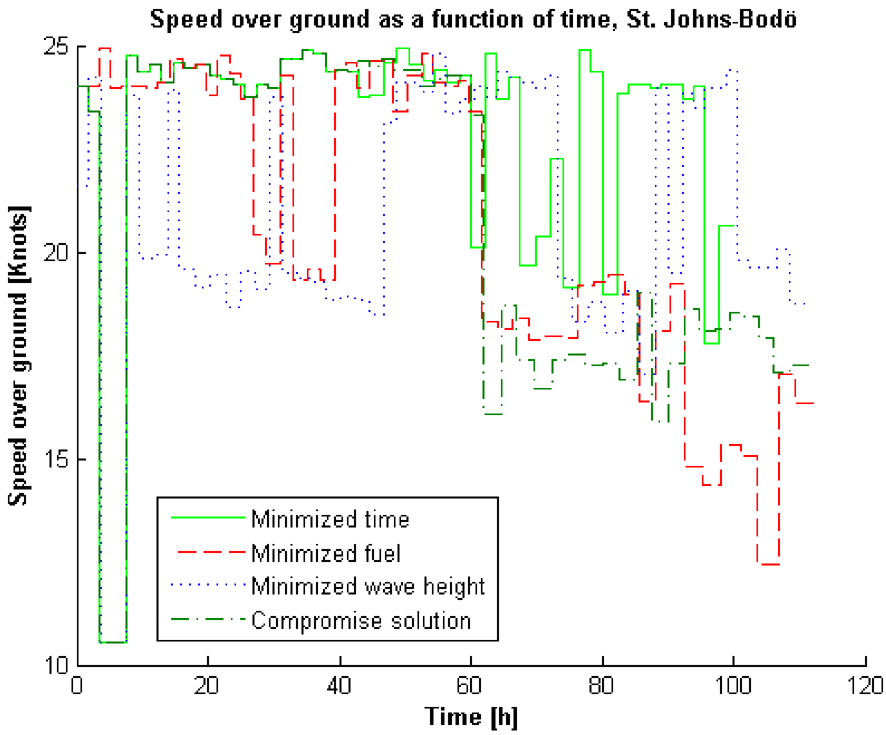

Figure 4.4: *Output of the grid search approach from St. Johns to Bodö with a maximum of three Pareto optimal routes per node in the grid when choosing to consider only a single objective at a time, together with the compromise solution singled out in the Pareto front of Figure 4.11. The plot shows suggested speeds at different times of the route.*

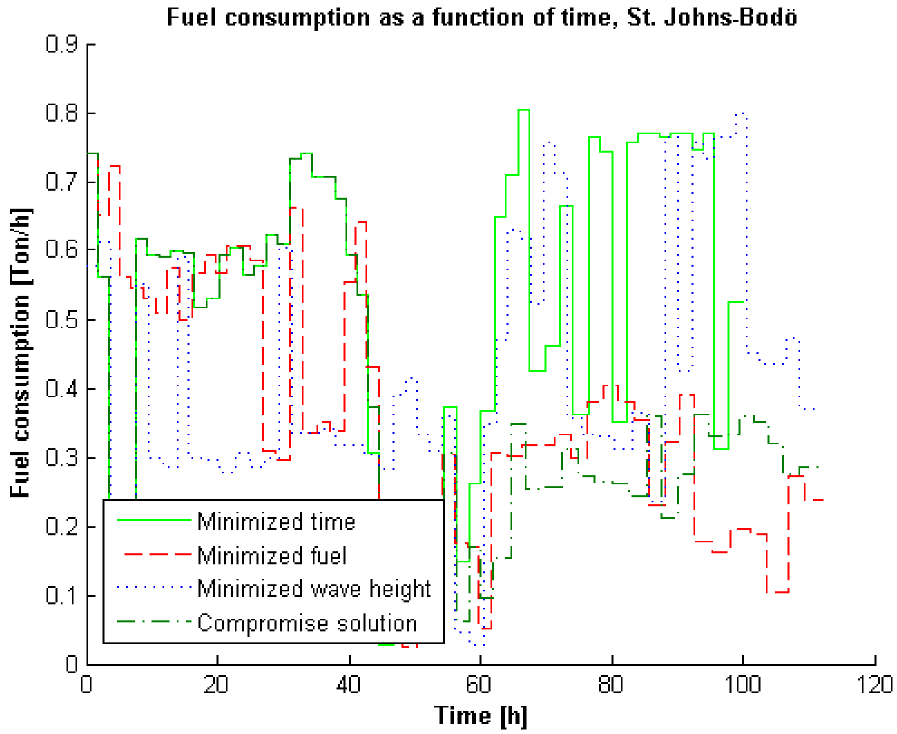

Figure 4.5: *Fuel consumption as a function of time for the single objective solutions, together with the compromise solution singled out in the Pareto front of Figure 4.11. From St. Johns to Bodö with a maximum of three Pareto optimal routes per node in the grid.*

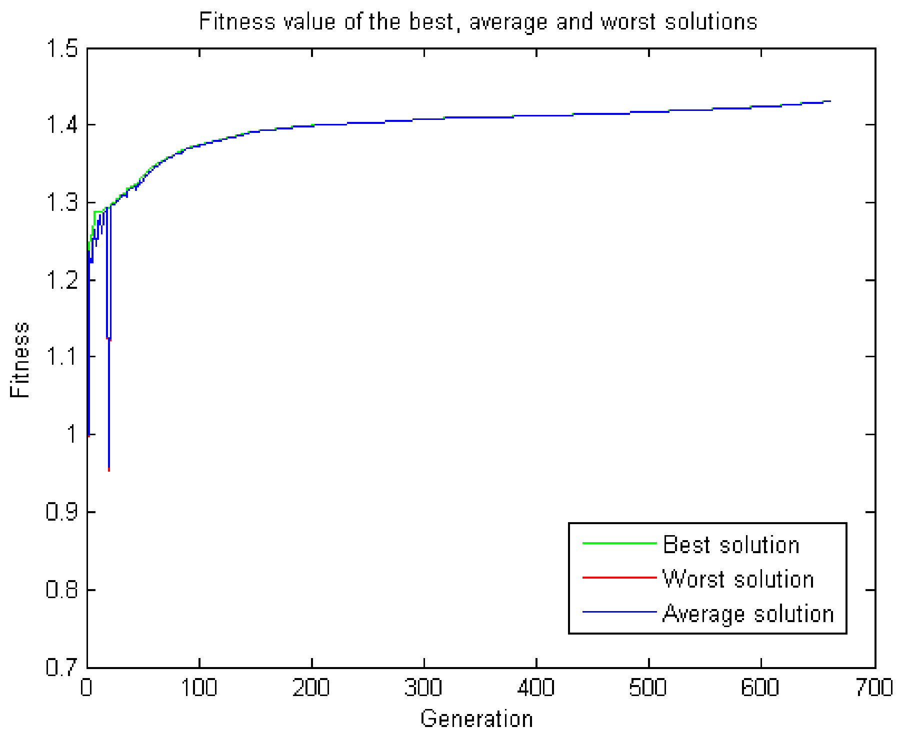

Figure 4.6: *Fitness as a funcion of generation number for the best, average and worst of the solutions. From St. Johns to Bodö. Parameters used: mutation rate $p_{mut} = 1/(s-2)$, where s is the total number of stages, population size 1000, maximum number of elites 896.*

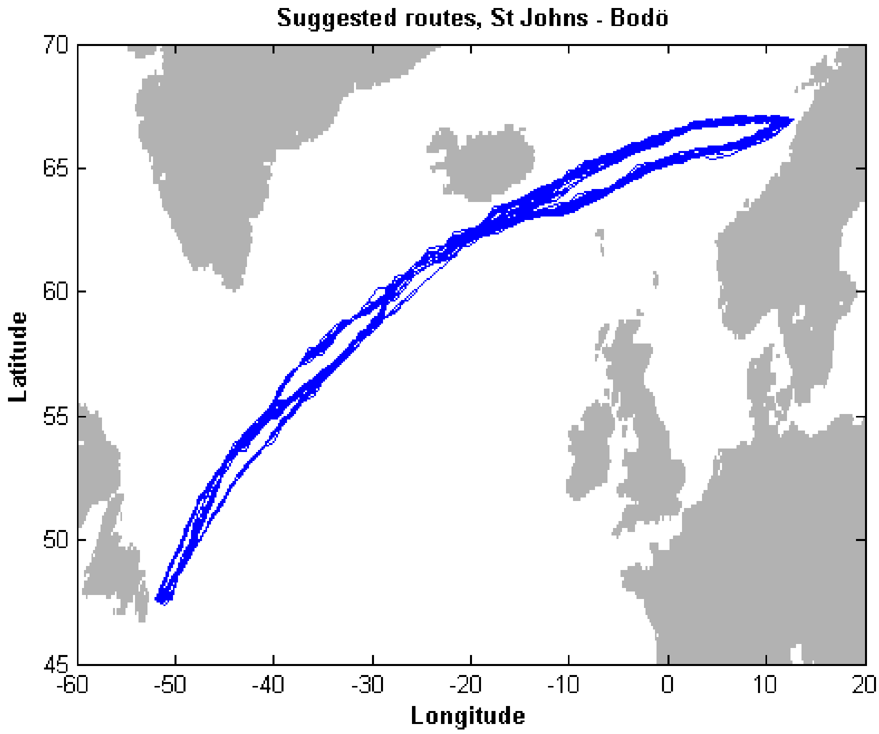

Figure 4.7: *Output of the modified DPGA approach, from St. Johns to Bodö. The plot shows all found Pareto optimal routes in space. Parameters used: mutation rate* $p_{mut} = 1/(s-2)$*, where s is the total number of stages, creep rate = 3 deg, population size 1000, maximum number of elites 896.*

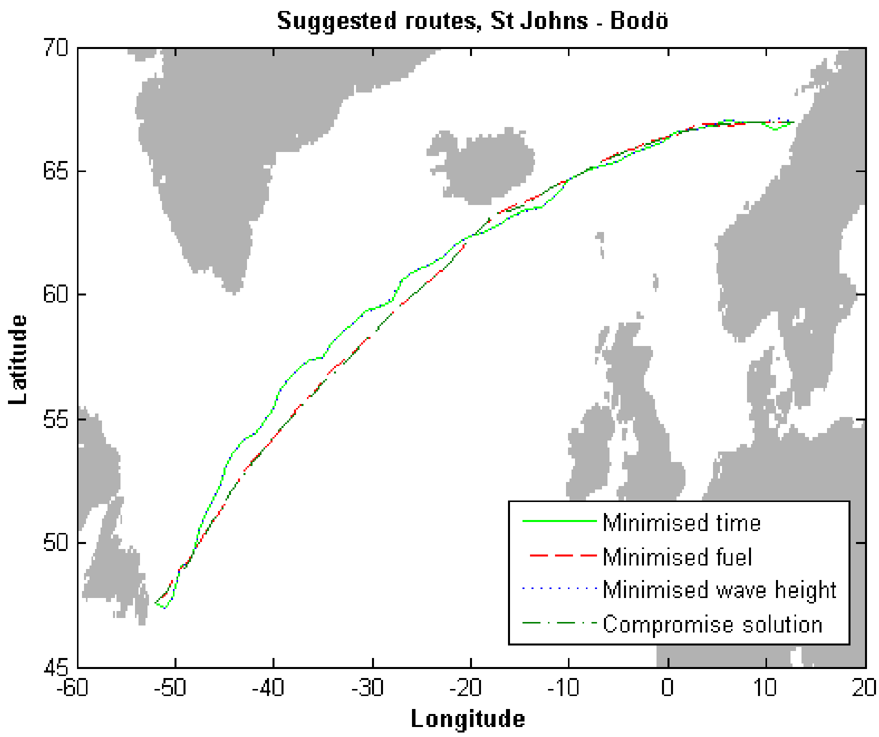

Figure 4.8: *Output of the modified DPGA approach, from St. Johns to Bodö. The plot shows the best found route with respect to one criteria at a time, together with the compromise solution singled out in the Pareto front of Figure 4.11. Parameters used: mutation rate* $p_{mut} = 1/(s-2)$*, where s is the total number of stages, creep rate = 3 deg, population size 1000, maximum number of elites 896.*

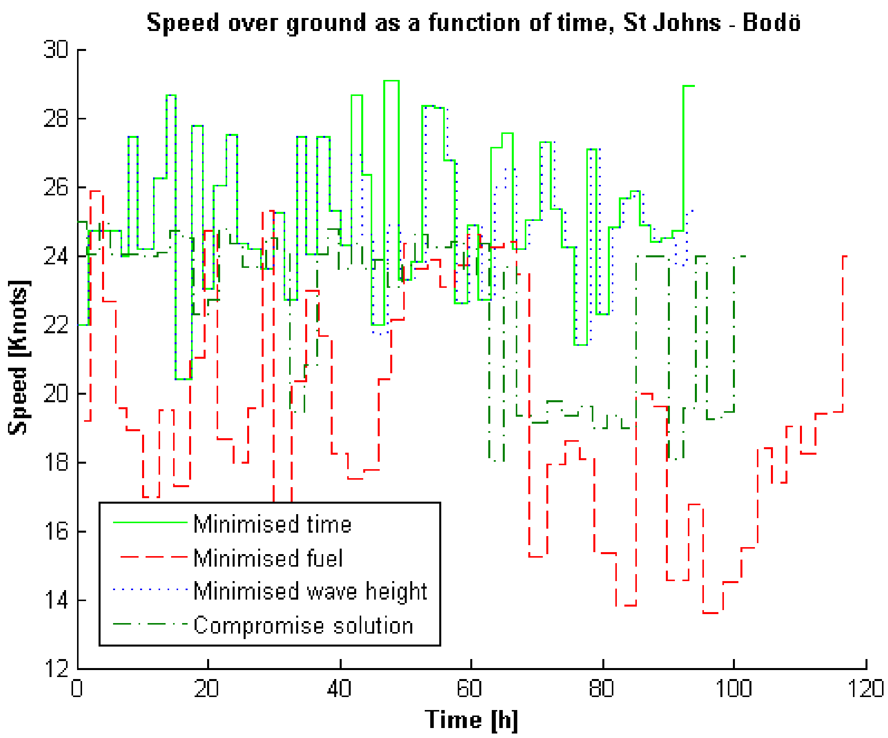

Figure 4.9: *Output of the modified DPGA approach, from St. Johns to Bodö. The plot shows the best found speeds with respect to one criteria at a time, together with the compromise solution singled out in the Pareto front of Figure 4.11. Parameters used: mutation rate* $p_{mut} = 1/(s-2)$*, where s is the total number of stages, creep rate = 3 deg, population size 1000, maximum number of elites 896.*

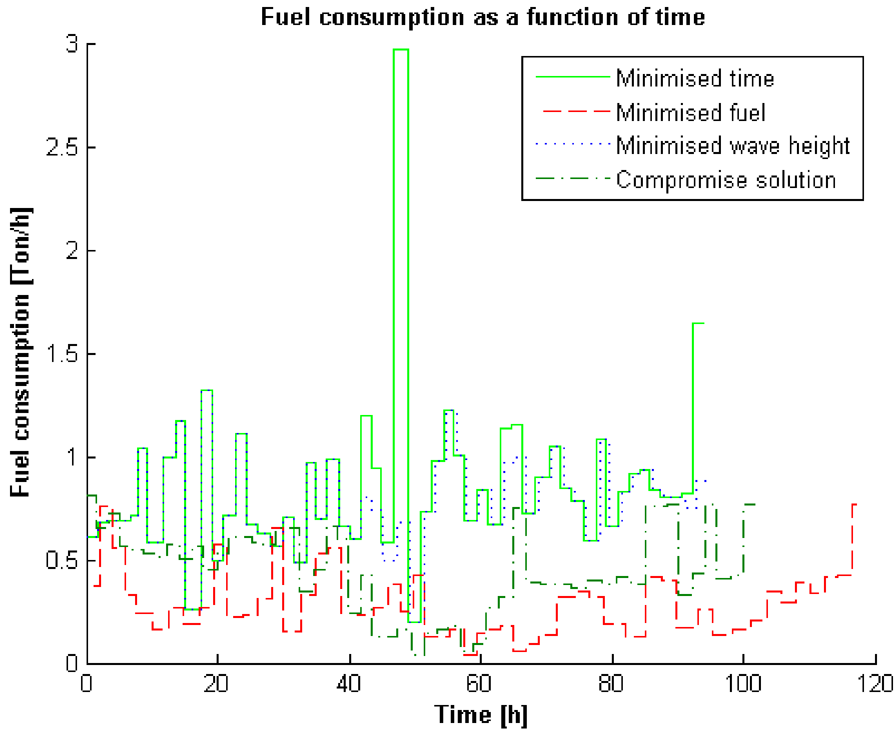

Figure 4.10: *Fuel consumption as a function of time for the single objective solutions, together with the compromise solution singled out in the Pareto front of Figure 4.11. From St. Johns to Bodö. Parameters used: mutation rate $p_{mut} = 1/(s-2)$, where s is the total number of stages, creep rate = 3 deg, population size 1000, maximum number of elites 896.*

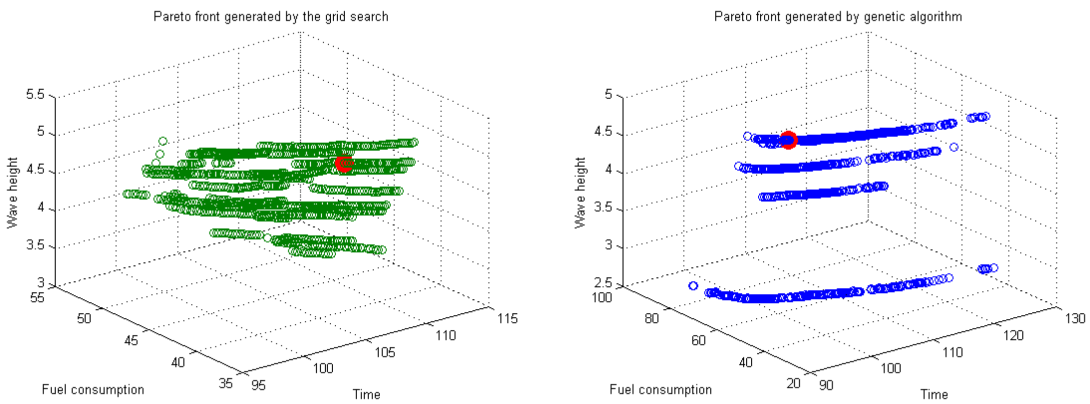

Figure 4.11: *Left: Normalised Pareto optimal routes of the grid search approach, with a maximum of three Pareto optimal routes per node in the grid. Right: Normalised Pareto optimal routes of the modified DPGA with parameters mutation rate $p_{mut} = 1/(s-2)$, where s is the total number of stages, population size 1000, maximum number of elites 896. From St. Johns to Bodö. Larger red marks the compromise solutions displayed in previous plots.*

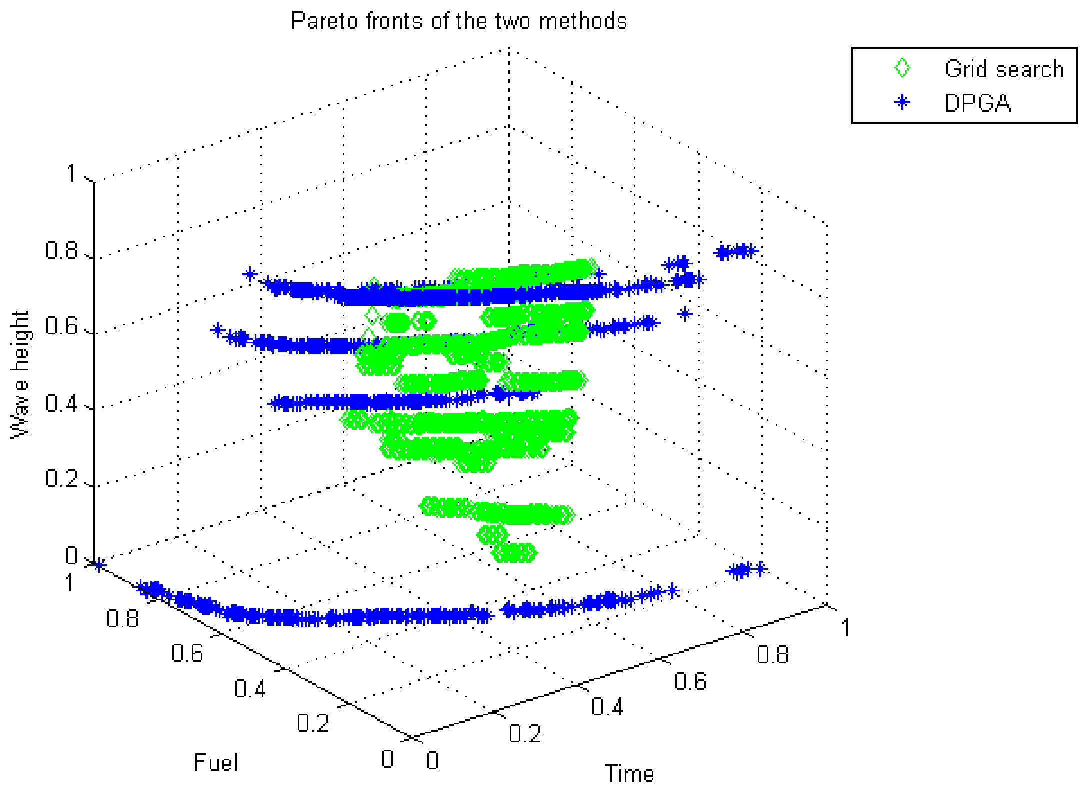

Figure 4.12: *Final Pareto front found by the two approaches, normalised.*

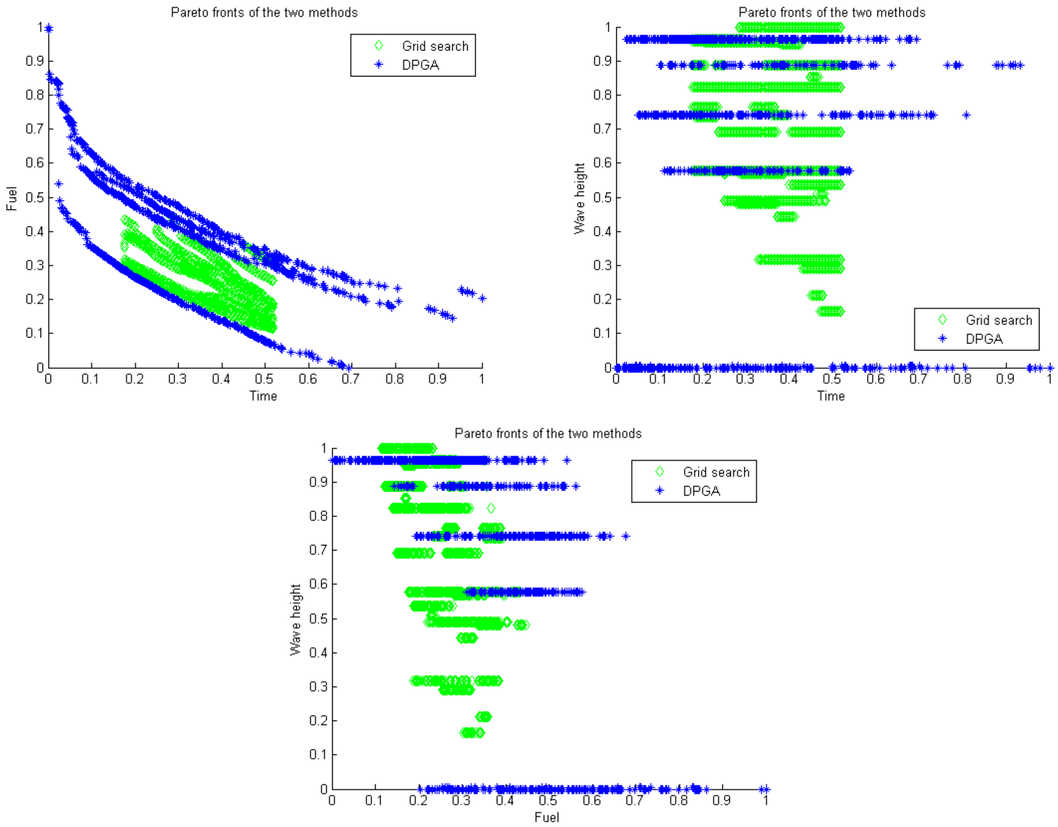

Figure 4.13: *Final Pareto front found by the two approaches, normalised and shown from different angles.*

Chapter 5

Discussion and conclusion

5.1 Grid search approach

From the result in Figures 4.1 and 4.2 it is evident that the grid based approach manages to find a range of different Pareto optimal routes. Figures 4.3 and 4.4 show more clearly how a specific route can be described in terms of location and speed. For a short trip (for example Gothenburg to le Havre) it is possible to obtain the Pareto front within a reasonable amount of time without discarding any solutions. For longer trips, for example St Johns to Bodö, the number of saved solutions needs to be filtered in each stage of the route in order to finish the optimisation before there is an updated weather report. On one hand this means that some optimal solutions will be thrown away. On the other hand there is an upper limit to how many optimal choices are actually useful for the end user of the final optimisation product. In other words this does not necessarily have a large impact on the usefulness of the product.

Optimising the trip from St Johns to Bodö on a standard stationary computer on a single Matlab worker takes approximately 93 hours. The same trip takes 4 hours and 47 minutes to optimise on 8 workers and 3 hours and 38 minutes to optimise on 16 workers. This suggests that there is much to be gained from running the grid method in parallel, however the usefulness of adding more workers decreases with the number of workers already in use. It has been investigated which parts of the code has the most to gain from being run in parallel. In the current implementation the parts of the code which would take up 99,6% of the runtime in the unparallellised version has been designed to run in parallel. In other words there is not much to be gained on rewriting a larger part of the code to run in parallel. It is clear that choosing the grid search method in its current form as the main optimisation tool would require a cluster in order to be updated fast enough to be in phase with the latest weather reports.

Figure 4.5 shows the fuel consumption at different times during the trip. Since the fuel consumption gives an indication of the engine load, it is highly relevant when considering what effect this kind of voyage planning would have on for example engine life length. This could be yet another criterion to optimise in the future.

5.2 Evolutionary algorithm approach

Figure 4.6 shows that the evolutionary algorithm approach manages to improve the initial population significantly. The fitness reflects both how good the objective values are, and how well the solutions are spread out in the objective function space. The dip in the average fitness in generation 20 indicates that at least one individual very far from the Pareto front was included in the population. This could happen either from mutating to an area of very high waves, or from increasing the speed between two stages. Since the interpolating function describing the fuel consumption is similar to a cubic function a small increase in speed could give a large increase in fuel consumption. The figure also shows that the result would probably also have been acceptable if the optimisation had only been allowed to run for about 200 generations, which corresponds to approximately 1 hour. In addition, the number of solutions presented in the two methods are unnecessarily large. Considering the fact that the runtime of the GA scales with the maximum number of chosen elites, the GA could most likely output a decent result a lot faster than in 1 hour if the maximum number of elites is set to something smaller. In this thesis it was set to such a large number simply for comparison. Figure 4.7 shows that the evolutionary algorithm approach also successfully finds a range of different Pareto optimal routes. Note that while the optimal routes found in the grid search method reflect the Pareto optimal solutions relative to almost all possible routes, the optimal routes found in the evolutionary algorithm method is only Pareto optimal with respect to the other routes which have been generated so far. Whether or not this has been achieved can be judged by checking if the fitness in Figure 4.6 seems to saturate at some value. In other words in this case the fitness indicates that it is very near to the true Pareto front.

A big advantage of the modified DPGA method is that the total runtime of the optimisation can be chosen. This is not possible in the grid search approach since stopping the algorithm halfway would only output routes for half the trip.

As in the grid search case Figures 4.8 and 4.9 gives a more clear indication of what an optimal solution can look like, and Figure 4.10 gives some interesting information about engine loads.

5.3 Comparison of the Pareto fronts

The normalised Pareto fronts of the two methods displayed in Figures 4.12 and 4.13 gives an indication of how well the two methods manage to find a good spread of compromise solutions. This is crucial for the final optimisation tool since the end user should not feel restricted by the options presented. The figures show that the GA approach finds a larger spread in terms of time. This is a consequence of the fact that the grid of the grid search method only contains a limited set of arrival times. This is a sensible approach in the case of container vessels since they typically get a slot time for unloading and loading cargo, which would mean that the objective is to arrive within a given time frame rather than as soon as possible. This could be achieved in the GA approach by using penalty functions.

The figures also show that the GA approach finds a larger spread in terms of fuel consumption. This could be both due to the limited time of the grid search approach, but also a consequence of the fact that there is a maximum fuel consumption set in each stage of the grid search approach, but not in the GA approach. As compensation the grid search approach instead finds a better spread of different wave heights. This is not too surprising since the solutions from the grid search approach are more evenly spread out in space than those of the GA approach (compare Figure 4.1 and Figure 4.7). This could be explained by the fact that the GA uses the three single objective minima (which are better than the randomly generated routes) as the initial elites.

5.4 Future work

A major challenge in the genetic algorithm approach is the difficulty of finding feasible solutions in terms of the maximum capacity of the ship engines. Since almost every randomly generated route is infeasible with respect to the maximum capacity of the ship engines, completely forbidding infeasible routes is not an option. Instead the constraint on the maximum capacity was indirectly implemented by the fact that the fuel consumption grows approximately cubically with the speed through water, and the fuel consumption is one of the criteria being minimised. Any future work could implement this more directly by adding a penalty to a violation of the maximum capacity. This would indeed be useful for any practical application of the optimisation tool.

Using the latitude as a decision variable clearly restricts the route to be mainly in the east-west direction, rather than in the north-south direction. This could easily be solved by defining a new coordinate system based on the great circle route between start and destination points (the great circe route is defined as is the intersection of the earth and a plane which passes through the center of the earth). The coordinate lines parallel with the great circle route would then correspond to the longitudes described in this thesis, and the coordinate lines perpendicular to the great circle route would correspond to the latitudes as described in this thesis. More work could also be done on evaluating different mutation and crossover operators for the specific case of multi-objective optimisation in weather routing.

One possible modification to the grid search method could be to sample the Pareto front, rather than to first evaluate the whole Pareto front and then remove superfluous points. That would probably make the algorithm run a lot faster and perhaps turn it into a viable alternative.

Much remains to be done in the field of voyage optimisation, in particular in the case of multiple criteria. As more weather, current and ship data become available and as more computing power can be utilised the prospect of more complex optimisation routines grows. More relevant criteria to take into account includes for example the life length of the ship engine and propellers, as well as avoiding areas with a high probability of parametric rolling.

5.5 Conclusion

While the modified DPGA is a lot faster than the grid search method (a fair estimation would be that the modified DPGA takes about 1% of the time to output an equivalently good result), it still lacks two critical points in order for it to be implemented in a working product. Firstly, it is only possible to optimise trips that are mainly in the east-west direction due to the implementation of the decision variables. Secondly, there is no constraint on the instantaneous fuel consumption. This means that there is nothing that limits the optimisation tool to suggest the ship to travel with a speed faster than the maximum capacity of the engine. Modifying these two points is necessary, but it is not expected to increase the runtime significantly. The grid search method on the other hand fulfils all requirements to be run in practice, but lacks the potential to be able to run fast enough. It also requires more computing power in order to be even considered an alternative optimisation tool.

In short the modified DPGA has a much higher potential than the grid search methods, but the former would need to be developed further in order to be ready for practical use.

Bibliography

[1] Review of Maritime Transport 2014, United Nations Conference on Trade and Development UNCTAD; 2014

[2] Notteboom T, Carriou P. Fuel surcharge practices of container shipping lines: Is it about cost recovery or revenue making?. Proceedings of the 2009 International Association of Maritime Economists (IAME) Conference, June, Copenhagen, Denmark; 2009

[3] Schuler M. How Many Shipping Containers Are Really Lost At Sea? gCapitan June 29, URL: http://gcaptain.com/how-many-shipping-containers-lost-at-sea/, Accessed 2015-05-05; 2014

[4] Deb K. Multi-objective optimization using evolutionary algorithms. Chichester: Wiley; 2001

[5] James R. W. Application of wave forecasts to marine navigation. Technical report, NAVAL OCEANOGRAPHIC OFFICE NSTL STATION MS; 1957

[6] Hagiwara H. Weather routing of (sail-assisted) motor vessels. PhD thesis, TU Delft, Delft University of Technology; 1989

[7] Klompstra M. 1B. Time aspects in games and in optimal control. PhD thesis, TU Delft, Delft University of Technology; 1992

[8] Haltiner G. J, Hamilton H. D, Arnason G. Minimal-time ship routing. Journal of Applied Meteorology, 1: 1–7; 1962

[9] Bleick W. E, Faulkner F.D. Minimal-time ship routing. Journal of Applied Meteorology, 4:217–221; 1965

[10] Bijlsma S. J. On minimal-time ship routing. PhD thesis, TU Delft, Delft University of Technology; 1975

[11] Bijlsma S. J. A computational method for the solution of optimal control problems in ship routing. Navigation, 48(3):145–154; 2001.

[12] Bijlsma S. J. On the applications of optimal control theory and dynamic programming in ship routing. Navigation, 49(2):71–80; 2002

[13] Bijlsma S. J. A computational method in ship routing using the concept of limited manoeuvrability. Journal of Navigation, 57(03):357–369; 2004

[14] Bryson A. E, Ho Y. Applied optimal control: optimization, estimation, and control. Taylor & Francis Group; 1975

[15] Zoppoli R. Minimum-time routing as an n-stage decision process. Journal of applied Meteorology, 11:429–435; 1972

[16] Bellman R. Dynamic Programming. Princeton University Press, Princeton, NJ, USA, 1 edition; 1957.

[17] Wei S, Zhou P. Development of a 3d dynamic programming method for weather routing. TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation, 6(1):79–85; 2012 http://www.transnav.eu/Article_Development_of_a_3D_Dynamic_Pro gramming_,21,337.html

[18] Hart P. E, Nilsson N. J, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2): 100–107; 1968

[19] Dijkstra E. W. A note on two problems in connexion with graphs. Numerische mathematik, 1(1):269–271; 1959

[20] Padhy C. P, Sen D, and Bhaskaran P. K. Application of wave model for weather routing of ships in the north indian ocean. Natural Hazards, 44 (3):373–385; 2008

[21] Schaffer J. D. Some experiments in machine learning using vector evaluated genetic algorithms. Ph. D. Thesis, Nashville, TN, Vanderbilt University; 1984

[22] Holland J. H. Adaption in natural and artificial systems. Ann Arbor, MI: MIT Press; 1975

[23] Goldberg D. E. Genetic algorithms for search, optimization, and machine learning. Reading, MA, Addison-Wesley; 1989

[24] Fonseca C. M, Fleming P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. Proceedings of the Fifth International Conference on Genetic Programming, pp.355-365; 1993

[25] Horn J, Nafploitis N, Goldberg D. A niched Pareto genetic algorithm for multi-objective optimization. Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 82-87; 1994

[26] Srinivas N, Deb K. Multi-objective function optimization using non-dominated sorting genetic algorithms. Evolutionary Computation Journal 2(3), 221-248; 1994

[27] Rudolph G. Convergence of evolutionary algorithms in general search spaces. Proceedings of the Third IEEE Conference on Evolutionary Computation, pp. 50-54; 1996

[28] Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. Technical Report 200001, Indian Institute of Technology, Kanpur: Kanpur Genetic Algorithms Laboratory (KanGAL); 2000

[29] Deb K, Agrawal S, Pratap A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI), pp. 849-858; 2000

[30] Osyczka A, Kundu S. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Structural Optimization 10(2) 94-99; 1995

[31] Zitzler E, Thiele L. An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. Technical Report 43, Zürich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH); 1998

[32] Hinnenthal J. Robust Pareto-Optimum Routing of Ships utilizing Deterministic and Ensemble Weather Forecasts. PhD thesis, der Technischen Universität Berlin; 2007

[33] Nelder J. A, Mead R. A simplex method for function minimization. The computer journal, 7(4):308–313; 1965

[34] Maki A, Akimoto Y, Nagata Y, Kobayashi S, Kobayashi E, Shiotani S, Ohsawa T, Umeda N. A new weather-routing system that accounts for ship stability based on a real-coded genetic algorithm. J Mar Sci Technol 16:311–322; 2011

[35] France W. N, Levadou M, Treakle T. W, Paulling J. R, Michel R. K, Moore C. An investigation of head-sea parametric rolling and its influence on container lashing systems. Mar Technol 40:1–19; 2003

[36] Marie S, Courteille E. Multi-Objective Optimization of Motor Vessel Route. TransNav - International Journal on Marine Navigation and Safety of Sea Transportation. Vol. 3. No. 2. pp. 133-141. Poland; 2009

[37] Szlapczynska J. Multicriteria Evolutionary Weather Routing Algorithm in Practice. TransNav - International Journal on Marine Navigation and Safety of Sea Transportation. Vol. 7. No.1. pp. 61-65. Poland; 2013

[38] Zitzler E, Thiele L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation, No. 3, Vol. 4, pp. 257-271. IEEE Computational Intelligence Society. USA; 1999

[39] Cheng H, Tsou M. An Ant Colony Algorithm for efficient ship routing. POLISH MARITIME RESEARCH 3(79) Vol 20; pp. 28-38; 2013

[40] Colorni A, Dorigo M, Maniezzo V, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142; 1991

[41] Skoglund L, et al. A comparative study of deterministic and ensemble weather forecasts for weather routing. Journal of Marince Science and Technology 11/2014; 2014

[42] K. Miettinen. Nonlinear Multiobjective Optimization. Springer, 1998.

[43] Osyczka A, Kundu S. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Structural Optimization 10(2), 94-99; 1995

[44] Hall P. The Distribution of Means for Samples of Size N Drawn from a Population in which the Variate Takes Values Between 0 and 1, All Such Values Being Equally Probable. Biometrika, Vol. 19, No. 3/4., pp. 240–245; 1927

[45] Irwin, J.O. On the Frequency Distribution of the Means of Samples from a Population Having any Law of Frequency with Finite Moments, with Special Reference to Pearson's Type II. Biometrika, Vol. 19, No. 3/4., pp. 225–239; 1927