# Surrogate-Assisted Ship Route Optimisation

Roman Dębski[0000−0003−3283−6032] and Rafał Dreżewski[0000−0001−8607−3478]

Institute of Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
{rdebski,drezew}@agh.edu.pl

**Abstract.** A new surrogate-assisted, pruned dynamic programming-based optimal path search algorithm – studied in the context of *ship weather routing* – is shown to be both effective and (energy) efficient. The key elements in achieving this – the fast and accurate physics-based surrogate model, the pruned simulation, and the OpenCL-based SPMD-parallelisation of the algorithm – are presented in detail. The included results show the high accuracy of the surrogate model (relative approximation error medians smaller than 0.2%), its efficacy in terms of computing time reduction resulting from pruning (from 43 to 60 times), and the notable speedup of the parallel algorithm (up to 9.4). Combining these effects gives up to 565 times faster execution. The proposed approach can also be applied to other domains. It can be considered as a dynamic programming based, optimal path planning framework parameterised by a problem specific (potentially variable-fidelity) cost-function evaluator.

**Keywords:** simulation-based optimisation · surrogate model · ship weather routing · optimal path planning · heterogeneous computing

## 1 Introduction

In international trade, approximately 80% of goods are transported by sea and this number is likely to increase [25]. This is why, in 2018, the International Maritime Organisation approved an agreement to reduce carbon emissions (from fuel consumption) per transport unit by 40% by 2030 as compared with 2008. As a result, fuel consumption reduction – one of the objectives in most ship route optimisation tasks – and optimal navigation itself have become more important than ever [19,23,6].

A model of fuel consumption usually assumes its dependence on the ship propulsion system and hull characteristics, the sea state, and the ship speed and its heading angle against the waves [13,30,8]. Any application of such a complex model in an optimisation task in most cases leads to a simulation-based algorithm, which in many instances can be computationally expensive. This issue is of particular importance in the context of algorithms based on dynamic programming, mostly due to search space size.

One way to address this is to parallelise the algorithm. In many scenarios, however, it is at most a partial solution to the problem. This can be due to target

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

computer system constraints, the intrinsic strong sequential component of the algorithm, and/or the cost of a single simulation, which is often by far the most computationally expensive part of the optimisation process.

Another option is to simplify the model (often significantly) so that the computational cost of a single simulation is acceptable, or even no simulation is required at all. Two examples of such simplifications are the assumption of a constant ship speed along a single segment of a route [19,26] and the use of the ship speed as the control input/signal [27]. But to ignore completely the ship acceleration may lead to inaccurate results, especially from the fuel consumption perspective. This approach also does not generalise to more complex, often multi-objective ship route optimisation tasks, which require simulators based on ship dynamics.

A surrogate-assisted algorithm could potentially address all the previously mentioned issues [11,12]. But, to the best of our knowledge, surrogate-assisted ship route optimisation based on dynamic programming has not been studied yet. Our aim is to present such an algorithm.

The main contributions of the paper are:

1. effective, physics-based surrogate model of ship motion (with fuel consumption included) defined in the spatial-domain rather than the time domain, in which the ODE-based (hi-fidelity) model is set (section 4.3),
2. admissible heuristic function used to accelerate (by search space pruning) both the surrogate-based estimation and hi-fidelity model based simulation (section 4.2),
3. surrogate-assisted, SPMD-parallel, dynamic programming based ship route optimisation algorithm, which incorporates a refinement of the search space (section 4.4),
4. results which demonstrate three important aspects of the algorithm: the accuracy of the surrogate-model, the pruning-related speedup, and the parallelisation capabilities (section 5).

The remainder of this paper is organised as follows. The next section presents related research. Following that, the optimisation problem under consideration is defined, and the proposed algorithm is described. After that, experimental results are presented and discussed. The last section contains the conclusion of the study.

## 2 Related research

Ship safety navigation and energy efficiency are crucial factors for enhancing the competitiveness and sustainability of ship operations [6]. Consequently, an increasing number of researchers work on ship route optimisation based on factors such as weather forecasts and sea conditions, which will significantly impact the ship velocity, primary engine output, fuel oil consumption (FOC), and emissions [19,23,5,6]. Typically, the ship route optimisation problem is reduced to a sub-path search problem in two dimensions, taking into account only the position

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

and assuming that either the ship velocity or primary engine output remains constant [27,6]. The methods applied to solve such simplified ship weather routing problem include [6]: the modified isochrone method [9], the isopone method for the minimum-time problem [20], dynamic programming for ship route optimisation with minimising FOC [2], and Dijkstra's algorithm taking into account the weather forecast data [18]. Also, evolutionary algorithms were applied to find the optimal ship route [22]. Nonetheless, in reality the long routes and severe fluctuations in weather and sea conditions make it impossible to maintain a constant sailing velocity and primary engine output for ships crossing the ocean [6].

The quest for a more realistic simulation of fuel consumption has led to the proposition of a number of three-dimensional optimisation algorithms. In [16] a real-coded genetic algorithm was used for weather routing optimisation taking into account the weighted criteria of fuel efficiency and ship safety. An alternative strategy is to integrate the time factor into the existing two-dimensional path search algorithm and solve the three-dimensional path search optimisation problem. This can be accomplished, for example, by adapting algorithms like the three-dimensional isochrones method with weighting factors [14].

Three-dimensional dynamic programming algorithms taking into account meteorological factors were also proposed by researchers. A novel three-dimensional dynamic programming based method for ship weather routing, minimising ship fuel consumption during a voyage was presented in [19]. In contrast to the approaches mentioned above, which modify only the ship heading while keeping the engine output or propeller rotation speed constant, the method proposed in [19] took into account both factors: the engine power and heading settings. The ship heading and speed planning taking into account the weather conditions (and some additional constraints and objectives) was realised using an improved three-dimensional dynamic programming algorithm for ship route optimisation [6]. The model of ship route estimation based on weather conditions, and additional constraints like the main engine rated power and ship navigation safety was proposed in [5]. The model could analyse multiple routes taking into account the indicators such as travel speed, fuel consumption, estimated time of arrival (ETA), and carbon emissions.

The selected machine learning models were also applied to the ship weather routing problem. The proposed approaches included training neural networks and machine learning models using available ship navigation data for different weather conditions. Such trained models were then used to predict FOC and optimise the ship route, taking into account fuel consumption and time of arrival [31]. Statistical models taking into account wave height, wave period, wind speed, and main engine's RPM data [17] were used to plan ship routes while maintaining safety, on-time arrival and reducing fuel consumption.

The approaches based on multi-objective optimisation to solving the ship route optimisation problem taking into account weather conditions were also proposed. A new algorithm for solving the two-objective problem (minimising the fuel consumption and the total risk) formulated as a nonlinear integer programming problem was introduced in [26]. The authors considered time-dependent

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

point-to-point shortest path problem with constraints including constant nominal ship speed and the total travel time.

Multi-objective evolutionary algorithms were also applied in several works to solve the ship weather routing problem, taking into account several objectives like estimated time of arrival, fuel consumption and safety [21,22,10]. The preference-based multi-objective evolutionary algorithm with weight intervals for ship weather routing optimisation was introduced in [23]. In the problem considered, the route consisted of control points. Each of them stored the information about its location, estimated time of arrival (based on the previous part of the route, weather forecast in the immediate vicinity, and speed of the ship), engine settings, and ship heading.

Computer simulations are now extensively employed to validate engineering designs, to refine the parameters of created systems and in the optimal search problems. Unfortunately, accurate (high-fidelity) simulation models for such tasks tend to be computationally expensive, making them often difficult or even unfeasible to apply in practice. When dealing with such situations, methods that employ surrogates, simplified (low-fidelity) simulation models, are utilised. Although these models are less complex than the original simulation model of a system or process, they accurately represent it and are significantly more efficient in terms of computation [11,12].

Broadly speaking, surrogate simulation models can be classified into two main categories. The first type, called *approximation-based models*, involves constructing a function approximation using data obtained from high-fidelity simulation models that are precise. The second type, known as *physics-based models*, involves building surrogates based on simplified physical models of systems or processes [12].

The approximation-based models are usually developed with the use of radial basis functions, Kriging, polynomial response surfaces, artificial neural networks, support vector regression, Gaussian process regression, or multidimensional rational approximation [12,28]. Moreover, recently deep-learning based surrogate models were proposed, for example, for characterisation of buried objects using Ground Penetrating Radar [29], design optimisation procedure of a Frequency Selective Surface based filtering antenna [15], and modelling of microwave transistors [3].

Physics-based surrogates typically incorporate simplified knowledge of the system or processes, as they are based on low-fidelity models. As a result, they usually only need a few high-fidelity simulations to be properly configured and provide reliable results [12]. Physics-based surrogates possess inherent knowledge about the simulated system or process, which gives them strong generalisation abilities. Consequently, they are capable of producing high-quality predictions of the accurate simulation model, even for system designs or configurations that were not used during the training phase [12].

It appears that the surrogate-based approach can help solve the problems that arise when using high-accuracy simulation models in the problem of optimising the ship's route when weather conditions are taken into account. These

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

problems have so far limited the ability to use accurate simulation models, and to the best of the authors' knowledge, the surrogate-based approach has not been used so far in the case of the ship weather routing problem.

Contrary to the above-mentioned approaches to the ship weather routing problem, in this paper we propose the physics-based surrogate model of ship motion, with fuel consumption included, defined in the spatial-domain. Furthermore, the admissible heuristic function for search space pruning (used to accelerate both the surrogate-based and hi-fidelity simulation model) is proposed. Finally, surrogate-assisted, SPMD-parallel, dynamic programming based ship weather routing algorithm, which incorporates a refinement of the search space, is introduced.

## 3 Problem formulation

Consider a ship sailing in the given area – $S_A$, from point $A$ to $B$, along the path/route:

$$\widetilde{AB}(t) = \left(x_1^{(\mathrm{AB})}(t), x_2^{(\mathrm{AB})}(t)\right)^T \tag{1}$$

resulting from a specific control input:

$$\mathbf{u}(t) = \mathbf{u}^{(\mathrm{AB})}(t) = \left(c(t), n(t)\right)^T, \tag{2}$$

where: $c$ denotes the ship course and $n$ - the propeller speed of rotation (RPM). Given the sea state[1] at time $t_k$ (see Fig. 1), we can express the ship dynamics in the following way:

$$\dot{\mathbf{x}}(t) = \mathbf{a}\left(\mathbf{x}(t), \mathbf{u}(t), t_k\right) = \mathbf{a}^{(k)}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \tag{3}$$

with $\mathbf{x}(t) = \left(x_1(t), x_2(t), \dot{x}_1(t), \dot{x}_2(t)\right)^T$ denoting the (ship) state vector.

The performance of the ship is evaluated in the following way:

$$J_s = C_1\ t_f + C_2 \int_0^{t_f} f_{cr}\left(\mathbf{x}(t), \mathbf{u}(t)\right) dt \tag{4}$$

where: $t_f$ is the sailing duration (the final time), $f_{cr}$ - the ship fuel consumption rate function, and $C_1, C_2$ are user-defined constants.

*Problem statement.* The route optimisation problem under consideration can be defined as follows[2]: *find an admissible control* $\mathbf{u}^*$ *which causes the ship to follow an admissible trajectory* $\mathbf{x}^*$ *that minimises the performance measure* $J_s$.

*Remark 1.* We assume that the values of $J_s$ can be found only through simulation and only on-board, off-line computers can be used.

[1] either explicitly or derived (computed) from the wind vector field

[2] $\mathbf{u}^*$ denotes an optimal control and $\mathbf{x}^*$ an optimal trajectory (they may not be *unique*)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

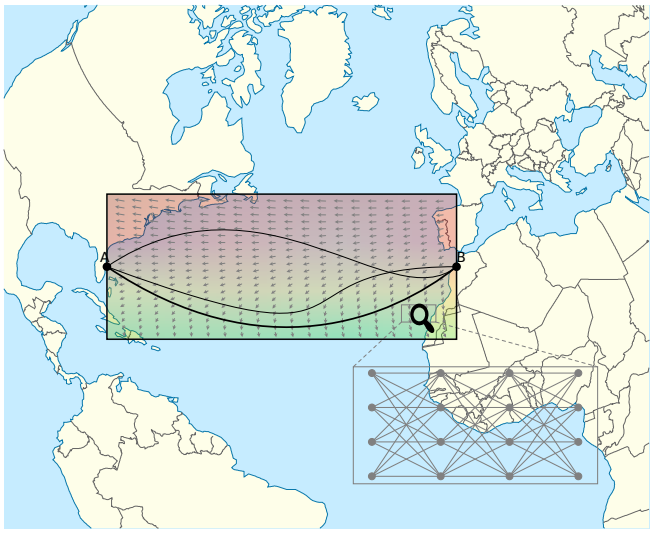


**Fig. 1.** Conceptual diagram of the ship route optimisation problem under consideration (source of the North Atlantic Ocean map: [24]).

## 4 Proposed solution

The approach we propose in this paper is based on the following two main steps:

1. transformation of the continuous optimisation problem into a (discrete) search problem over a specially constructed graph;
2. application of surrogate-assisted, pruned dynamic programming to find the approximation of the optimal control input sequence:

$$(\mathbf{u}_k)_{k=1}^N = \left([c_1, n_1]^T, [c_2, n_2]^T, \ldots, [c_N, n_N]^T\right). \tag{5}$$

*Remark 2.* The sequence $(c_k)_{k=1}^N$ represents a *continuous piecewise-linear* sailing path[3], whilst $(n_k)_{k=1}^N$ is the sequence of the propeller RPMs, which forms a *piecewise-constant* function; in both cases $t \in [0, t_f]$.

*Remark 3.* The above two steps repeated several times form an adaptive version of the algorithm in which subsequent search spaces are generated through mesh refinement making use of the best solution found so far.

The key elements of the proposed algorithm, i.e.:

1. *3D-graph* based solution space and the SPMD-parallel computational topology it generates,
2. pruning-accelerated simulation and estimation,
3. fast and accurate estimator of the performance measure (surrogate model)

are discussed in the following sub-sections.

[3] formed by the sequence of the graph edges that correspond to $(c_k)$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

### 4.1 The solution space representation

Discretisation of the original problem domain may be seen as a two-phase process. In the first phase, we construct a multi-stage graph ($G_2$) that creates the *'route space'* (i.e., $c_k$-space/spatial-dimension of ($\mathbf{u}_k$), see Remark 2). In most instances, this graph is regular with equidistant nodes grouped in rows and columns: $n_c$ columns and $(n_r - 2)$ regular rows, plus two special (single-node) rows – one with point $A$ and the other with point $B$ (see Fig. 2).

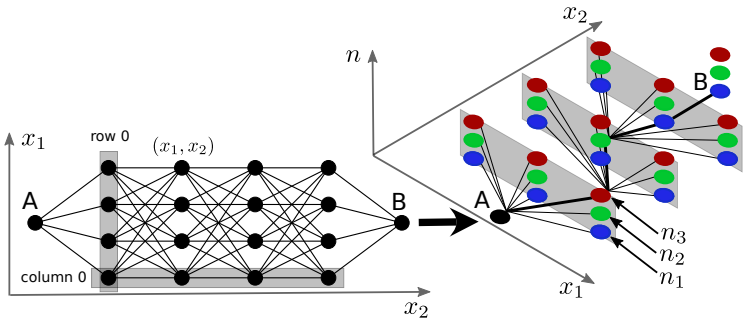


**Fig. 2.** Solution space representation.

In the second phase, each node of $G_2$ (apart from the one corresponding to point $A$) is replicated $n_p$ times to add the $n_k$-dimension to the solution space (of control inputs), which can be seen as an addition of the 'third dimension' to $G_2$. This new, '*3D-graph*' ($G_3$):

- has $n_p\,[n_c(n_r - 2) + 1] + 1$ nodes,
- has $n_c n_p\,[(n_r - 3) n_c n_p + n_p + 1]$ edges,
- represents $n_p (n_c n_p)^{n_r - 2}$ different control input sequences.

*Remark 4.* The piecewise-linear approximation of the ship route simplifies the problem significantly. Instead of one complex, continuous two-dimensional problem, we have a series of simple one-dimensional sub-problems – each corresponding to a single route segment only.

### 4.2 Pruning in simulation and estimation

The evaluation of a single route segment, $s$, can be aborted as soon as we know that the computation cannot lead to a better solution than the best one found so far. This process can be based on any effective *admissible heuristic*[4].

Such a heuristic can be derived in the following way (see Eq. 4):

$$J_s \leq J_s^{(adm)} = C_1 \frac{|s|}{v_{max}} + C_2\, f_{cr}^{(min)} \frac{|s|}{v_{max}} = \left(C_1 + C_2\, f_{cr}^{(min)}\right) \frac{|s|}{v_{max}} \qquad (6)$$

[4] an *admissible heuristic* never overestimates the actual value of performance measure

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

where: $f_{cr}^{(min)}$ denotes the *minimum fuel consumption rate* and $v_{max}$ is the *maximum possible speed* of the ship.

*Remark 5.* The values of $f_{cr}^{(min)}$ and $v_{max}$ can be either taken from the ship characteristics or approximated during the segment estimation phase.

### 4.3 Surrogate-based performance measure estimator

The proposed estimator of the performance measure/cost of a single, straight route segment $AB$ (corresponding to one edge of $G_2$) is an extended version of the one introduced in [7,4], and based on the *work-energy principle*:

$$\int_A^B \vec{F} \cdot d\vec{r} = \frac{1}{2}\, m(v_B^2 - v_A^2),$$

This formula can be used to transform the original problem from *time domain* to *spatial domain*, i.e.,

$$m\frac{dv}{dt} = mv\frac{dv}{ds} = F \rightarrow mvdv = Fds = dW.$$

From this, we can find the distribution of the velocity along the sailing line:

$$\begin{cases} s_0 = s_A, \\ v_0 = v_A, \\ v_i^2 = v_{i-1}^2 + \frac{2}{m} F(s_{i-1}, v_{i-1}) \Delta s_i \end{cases} \tag{7}$$

where: $i = 1, 2, \ldots, i_B$, $\Delta s_i = s_i - s_{i-1}$, and $s_{i_B} = s_B$. Knowing $(v_i)_{i=1}^{i_B}$ and assuming a constant value of $F$ in each sub-interval, we can find the sailing duration and fuel consumption (see Appendix A and [7,4]).

*Remark 6.* Operating in the *spatial domain* is the key property of this estimator because it takes a *predetermined number of steps*, $N_s$, stemming from the spatial discretisation of a route segment. In the original, hi-fidelity model – given in the time domain – the number of time-steps to be taken by the simulator (i.e., the ODE solver) to reach the final point is unknown upfront. In some cases, it can be several orders of magnitude larger than $N_s$.

### 4.4 The algorithm

Graph $G_3$, representing the solution space, is directed, acyclic (DAG) and has a layered structure[5]. Since, at the beginning of the search process, the cost of each route segment is unknown, it has to be obtained from simulation. The cost of reaching each node of $G_3$ can be computed using the *Principle of Optimality* [1]. It can be expressed for an example path $A\text{-}N_{r,c,n}$ in the following way:

$$J_s^*\left(A \rightarrow N_{r,c,n}\right) = \min_{c_j, n_k} \left\{ J^*\left(A \rightarrow N_{r-1,c_j,n_k}\right) + J\left(N_{r-1,c_j,n_k} \rightarrow N_{r,c,n}\right) \right\} \tag{8}$$

[5] therefore, it is a multistage graph

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

where: $c_j = 0, \ldots, (n_c - 1)$, $n_k = 0, \ldots, (n_p - 1)$, $J_s\,(N_s \to N_e)$ is the cost corresponding to the path $N_s \to N_e$ ($N_s$ – the start node, $N_e$ – the end node), $J_s^*$ represents the optimal value of $J_s$, and $N_{r,c,s}$ is the node of $G_3$ with 'graph coordinates' $(row, column, RPM) = (r, c, n)$.

The structure of the computation (i.e., simulation flow) is reflected in the SPMD-structure of Algorithm 1 (see annotation **@parallel**). The computation begins from point $A$ in layer 1, taking into account the corresponding initial conditions, and is continued (layer by layer) for the nodes in subsequent rows. On the completion of the simulations for the last layer (i.e., reaching the end node $B$), we get the optimal route and its cost.

**Algorithm 1:** Surrogate-assisted optimal route search

**Input:**
- $G_3^{(\mathrm{AB})}$: initial search space with start point $A$ and target point $B$,
- $S_{\mathrm{st}}$: sea state,
- HFM: ship simulator (high-fidelity model),
- LFM: surrogate (low-fidelity model).

**Output:** the optimal route and RPMs

**foreach** *refinement* $r_G$ *of* $G_3^{(AB)}$ **do**
  **foreach** *layer* $l_{r_G}$ *in* $r_G$ **do**
    **@parallel foreach** *entry point* $e_p$ *of* $l_{r_G}$ **do**
      $(s_k)_{k=1}^{d} \leftarrow$ all segments ending in $e_p$ `// d - the in-degree of` $e_p$
      `/* 1. pruned estimation using LFM,` $d_1 \leq d$`, Section 4.3 */`
      $(s_k, \mathrm{cost}_k^{\mathrm{est}})_{k=1}^{d_1} \leftarrow$ pruned estimation of $(s_k)_{k=1}^{d}$
      `/* 2. pruned simulation using HFM, see Appendix A */`
      $(s_k, \mathrm{cost}_k^{\mathrm{sim}})_{k=1}^{d_1} \leftarrow$ pruned simulation of $(s_k, \mathrm{cost}_k^{\mathrm{est}})_{k=1}^{d_1}$
      `/* 3. saving the optimal` $e_p$`-entry info,` $J_s^*\,(A \to e_p)$ `*/`
      $(s_{e_p}^{\min}, \mathrm{cost}_{e_p}^{\min}) \leftarrow \min_{\text{by cost}} (s_k, \mathrm{cost}_k^{\mathrm{sim}})_{k=1}^{d_1}$
      `save` $(e_p, (s_{e_p}^{\min}, \mathrm{cost}_{e_p}^{\min}))$

*Complexity analysis.* Algorithm 1 average-case time complexity is determined by the number of solution space refinements, $n_i$, the average number of reduced force evaluations[6] for a single path/route segment, $\bar{n}_F$, and the number of such segments, $n_c n_p\,[(n_r - 3) n_c n_p + n_p + 1]$ (see Section 4.1). For the sequential version of the algorithm it can be expressed as:

$$T_s = \Theta\left(n_i\ n_r\ n_c^2\ n_p^2\ \bar{n}_F\right). \tag{9}$$

[6] values of $F$ are used both in HFM and LFM; Runge-Kutta-Fehlberg 4(5) method, used in the simulator, requires at each step six evaluations of $F$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

In the SPMD-parallel version of the algorithm, the evaluations for all nodes in a given row can be performed in parallel (using $p$ processing units), thus:

$$T_p = \Theta\left(n_i\ n_r\ n_c\ n_p \left\lceil \frac{n_c\ n_p}{p} \right\rceil\ \bar{n}_F\right). \tag{10}$$

The Algorithm 1 *space complexity* formula, $\Theta(n_r\ n_c\ n_p)$, arises from the solution space representation.

*Remark 7.* Significant reduction of the average-case of computational cost of the algorithm is important for at least two reasons. Firstly, we often need to know the solution as soon as possible (sometimes for safety). Secondly, since we only use on-board computers, energy efficiency is critical while at sea.

## 5 Results and discussion

To demonstrate the effectiveness of the algorithm, a series of experiments was carried out using a MacBook Pro[7] with macOS 12.6.3 and OpenCL 1.2. This system has one operational OpenCL-capable device: Intel Iris Graphics 6100, 1536 MB (the integrated GPU). The aim of the experiments was to investigate three important aspects of the algorithm: the accuracy of the surrogate-model, the pruning-related computational time cost reduction, and the SPMD-parallelisation efficiency. The results are presented in the subsequent paragraphs.

*The accuracy of the surrogate-model.* This element has a direct and significant impact on the computational cost reduction since it is strictly related to pruning efficiency. Indeed, the more accurate the estimator is, the more simulations can be aborted (or even completely omitted). The results are given in the form of a violin plot in Fig. 3. The plot shows the distributions of relative estimation errors, i.e.:

$$\mathrm{err}(J_s|^{est}_{sim}) = \left| \frac{J_s^{(est)} - J_s^{(sim)}}{J_s^{(sim)}} \right| \tag{11}$$

for route segments from the solution spaces corresponding to $n_c = 16$, 32, 64, and 128.

*Remark 8.* In all cases, the medians of relative estimation errors were smaller than 0.002 (i.e., 0.2%), which confirms the very high accuracy of the proposed surrogate-based segment cost estimator.

*Efficacy of the surrogate-based pruning.* Having verified the accuracy of the surrogate-model, we can test the computation speedup resulting from surrogate-based pruning applied both in the simulation and estimation phases. The corresponding results are given in Table 1 and Fig. 4.

[7] Retina, 13-inch, Early 2015, with 16GB of DDR3 1867 MHz RAM

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

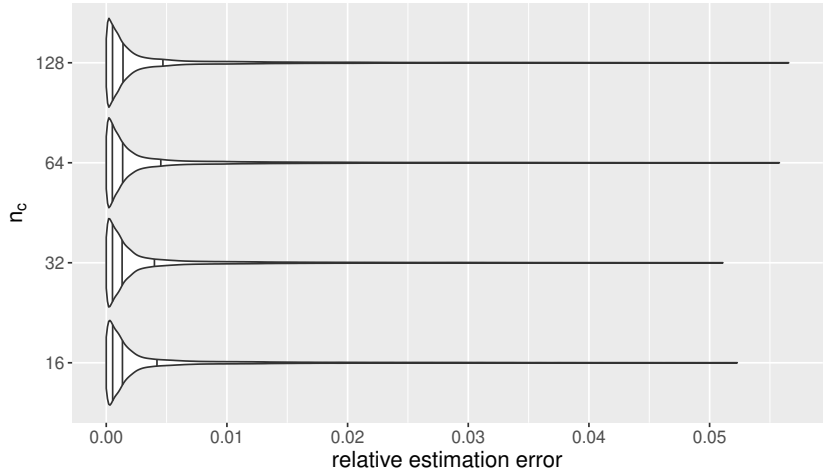


**Fig. 3.** Distributions of relative errors of the performance measure/cost estimate (see Eq. 11) for solution spaces with different $n_c$, with outliers ($e > \bar{e}+3\,\sigma_e$, if any) excluded.

**Table 1.** Efficacy of surrogate-based pruning: execution times (in seconds), and the pruning-related speedup for different $n_c$. The solution space (one refinement) with $n_r = 32$, $n_p = 8$. Statistics from ten runs.

| $n_c$ | EXECUTION TIMES ($t_{sim}$) | | | | | | | | SPEEDUP |
|---|---|---|---|---|---|---|---|---|---|
| | BASE MODEL | | | | SURROGATE-ASSISTED | | | | |
| | min | max | avg | sd | min | max | avg | sd | |
| 16 | 271.1 | 272.6 | 271.8 | 0.52 | 6.2 | 6.3 | 6.2 | 0.05 | 43.5 |
| 32 | 1077.3 | 1080.7 | 1078.9 | 1.25 | 18.5 | 18.7 | 18.6 | 0.08 | 58.0 |
| 64 | 4399.9 | 4424.3 | 4410.8 | 7.24 | 74.5 | 75.5 | 74.7 | 0.22 | 59.1 |
| 128 | 17494.2 | 17609.7 | 17559.3 | 39.62 | 292.4 | 294.3 | 292.7 | 0.59 | 60.0 |

SPMD-*parallelisation efficiency.* Parallelisation is another way of lowering the total computation time. Contemporary mobile/on-board computers are usually equipped with more than one type of processor, typically one CPU and at least one GPU. OpenCL makes it possible to use these heterogeneous platforms effectively, since the same code can be executed on any OpenCL-capable processor. The execution times for different sizes of the solution space and the corresponding parallel-speedups are presented in Table 2 and Fig .4. Its maximum recorded value was 9.4 (see Table 2 and Fig. 4). With reference point set as the sequential search based on the full simulation, it gives in total 565.4 times faster execution.

## 6 Conclusions

It has been shown that the surrogate-assisted, pruned dynamic programming based ship route optimisation algorithm can be both effective and (energy)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

**Table 2.** SPMD-parallelisation efficiency: execution times, $t_{sim}$ (in seconds) and (parallel) speedup for different $n_c$. The remaining parameters as in Table 1.

| $n_c$ | EXECUTION TIMES ($t_{sim}$) | | | | SPEEDUP |
|---|---|---|---|---|---|
| | min | max | avg | sd | |
| 16 | 4.0 | 4.0 | 4.0 | 0.02 | 1.6 |
| 32 | 6.9 | 6.9 | 6.9 | 0.01 | 2.7 |
| 64 | 15.1 | 15.1 | 15.1 | 0.02 | 4.9 |
| 128 | 31.0 | 31.1 | 31.1 | 0.02 | 9.4 |

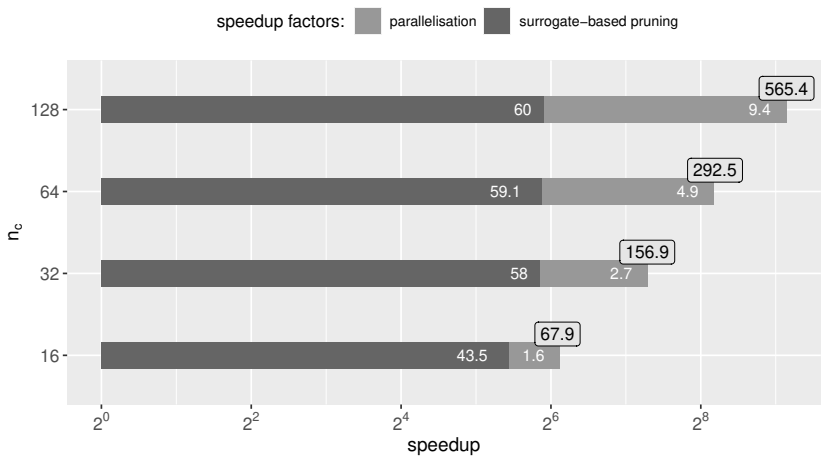


**Fig. 4.** Total speedup (boxed numbers at the end of each bar) and its factors: surrogate-based pruning (dark-grey part) and SPMD-parallelisation (light-grey part) for different $n_c$. The remaining parameters as in Table 1.

efficient. The key elements in achieving this have been the fast and accurate physics-based surrogate model, the pruned simulation, and the OpenCL-based SPMD-parallelisation of the algorithm.

The results show the high accuracy of the surrogate model (the medians of relative approximation errors were smaller than 0.2%, see Fig. 3), its efficacy in terms of the reduction of computing time resulting from pruning (from 43 to 60 times, see Table 1 and Fig. 4), and the notable speedup of the parallel algorithm (its maximum observed value was 9.4, see Fig. 4). Combining these effects has given up to 565 times faster execution time (see Fig. 4).

The proposed approach can also be applied to other scenarios. In fact, it can be considered as a dynamic programming-based, optimal path planning framework parameterised by a problem specific (potentially variable-fidelity) cost-function evaluator.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

Future research work could concentrate on verification of the proposed algorithm with a more accurate ship simulator and different types of surrogate models (e.g., data-driven deep learning based).

## Appendix A The ship simulation model

1. The *ship equation of motion* along a straight route segment:

$$(M + M_A)\frac{dv_s}{dt} = T_p(t) - (R_T(t) + R_{AW}(t)) \tag{12}$$

where: $M$ denotes the ship mass, $M_A$ - the 'added' mass, $v_s$ - speed of ship, $T_p$ - trust force of the propeller, $R_T$ - frictional resistance, and $R_{AW}$ - wave-making resistance; in addition: $M_R = M + M_A = 43\ 341\ 239$,

2. *Trust force*:

$$T_p(t) = 425\left(1 - 4.45\frac{v_s(t)}{n}\right)n^2 \tag{13}$$

3. *Frictional resistance*:

$$R_T(t) = 11000\,(v_s(t))^2 \tag{14}$$

4. *Wave-making resistance*:

$$R_{AW}(t) = 562.5\left[\frac{2\pi}{0.75v_w(t)}\left(1 - \frac{2\pi}{0.75v_w(t)}v_s(t)\cos\mu\right)v_s(t)\right]^2 \tag{15}$$

where: $v_w$ is wave speed, $v_s$ - ship speed, and $\mu$ is the 'encountering angle', i.e., the angle between the direction of wave travel and the direction of ship heading, which is measured in a clockwise manner from the direction of wave travel,

5. *Fuel consumption* (for a single route segment):

$$FOC_s = \int_0^{t_f} \max\left\{2\,\frac{R_T(t) + R_{AW}(t)}{M_R}\,v_s(t), 0.0625\right\}dt \tag{16}$$

6. *Weights of the objectives* (see Eq. 4):

$$C_1 = C_2 = 0.5 \tag{17}$$

**Acknowledgement.** The research presented in this paper was partially supported by the funds of Polish Ministry of Education and Science assigned to AGH University of Science and Technology.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

## References


1. Bellman, R., Dreyfus, S.: Applied Dynamic Programming. Princeton University Press, Princeton, New Jersey (1962)
2. Bijlsma, S.J.: On the applications of optimal control theory and dynamic programming in ship routing. Navigation **49**(2), 71–80 (2002). https://doi.org/10.1002/j.2161-4296.2002.tb00256.x
3. Calik, N., Güneş, F., Koziel, S., Pietrenko-Dabrowska, A., Belen, M.A., Mahouti, P.: Deep-learning-based precise characterization of microwave transistors using fully-automated regression surrogates. Scientific Reports **13**(1), 1445 (2023). https://doi.org/10.1038/s41598-023-28639-4
4. Dębski, R., Sniezynski, B.: Pruned simulation-based optimal sailboat path search using micro hpc systems. In: International Conference on Computational Science. pp. 158–172. Springer (2021). https://doi.org/10.1007/978-3-030-77970-2_13
5. Du, W., Li, Y., Zhang, G., Wang, C., Chen, P., Qiao, J.: Estimation of ship routes considering weather and constraints. Ocean Engineering **228**, 108695 (2021). https://doi.org/10.1016/j.oceaneng.2021.108695
6. Du, W., Li, Y., Zhang, G., Wang, C., Zhu, B., Qiao, J.: Energy saving method for ship weather routing optimization. Ocean Engineering **258**, 111771 (2022). https://doi.org/10.1016/j.oceaneng.2022.111771
7. Dębski, R., Dreżewski, R.: Adaptive surrogate-assisted optimal sailboat path search using onboard computers. In: Groen, D., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) Computational Science — ICCS 2022. pp. 355–368. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-08757-8_30
8. Ghaemi, M.H., Zeraatgar, H.: Analysis of hull, propeller and engine interactions in regular waves by a combination of experiment and simulation. Journal of Marine Science and Technology **26**(1), 257–272 (2021). https://doi.org/10.1007/s00773-020-00734-5
9. Hagiwara, H., Spaans, J.A.: Practical weather routing of sail-assisted motor vessels. The Journal of Navigation **40**(1), 96–119 (1987). https://doi.org/10.1017/S0373463300000333
10. Hinnenthal, J., Clauss, G.: Robust pareto-optimum routing of ships utilising deterministic and ensemble weather forecasts. Ships and Offshore Structures **5**(2), 105–114 (2010). https://doi.org/10.1080/17445300903210988
11. Koziel, S., Leifsson, L. (eds.): Surrogate-Based Modeling and Optimization. Springer, New York, NY (2013). https://doi.org/10.1007/978-1-4614-7551-4
12. Koziel, S., Ogurtsov, S.: Antenna Design by Simulation-Driven Optimization. SpringerBriefs in Optimization, Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04367-8
13. Li, J., Jia, Y.: Calculation Method of Marine Ship Fuel Consumption. IOP Conference Series: Earth and Environmental Science **571**(1), 012078 (2020). https://doi.org/10.1088/1755-1315/571/1/012078
14. Lin, Y.H., Fang, M.C., Yeung, R.W.: The optimization of ship weather-routing algorithm based on the composite influence of multi-dynamic elements. Applied Ocean Research **43**, 184–194 (2013). https://doi.org/10.1016/j.apor.2013.07.010
15. Mahouti, P., Belen, A., Tari, O., Belen, M.A., Karahan, S., Koziel, S.: Data-Driven Surrogate-Assisted Optimization of Metamaterial-Based Filtenna Using Deep Learning. Electronics **12**(7), 1584 (2023). https://doi.org/10.3390/electronics12071584


ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31

16. 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. Journal of Marine Science and Technology **16**(3), 311–322 (2011). https://doi.org/10.1007/s00773-011-0128-z
17. Mao, W., Rychlik, I., Wallin, J., Storhaug, G.: Statistical models for the speed prediction of a container ship. Ocean Engineering **126**, 152–162 (2016). https://doi.org/10.1016/j.oceaneng.2016.08.033
18. Padhy, C., Sen, D., Bhaskaran, P.: Application of wave model for weather routing of ships in the north indian ocean. Natural Hazards **44**, 373–385 (2008). https://doi.org/10.1007/s11069-007-9126-1
19. Shao, W., Zhou, P., Thong, S.K.: Development of a novel forward dynamic programming method for weather routing. Journal of Marine Science and Technology **17**(2), 239–251 (2012). https://doi.org/10.1007/s00773-011-0152-z
20. Spaans, J.: New developments in ship weather routing. Navigation pp. 95–106 (1995)
21. Szlapczynska, J.: Multiobjective Approach to Weather Routing. TransNav International Journal on Marine Navigation and Safety of Sea Transportation **1**(3) (2007)
22. Szlapczynska, J., Smierzchalski, R.: Multicriteria optimisation in weather routing. TransNav the International Journal on Marine Navigation and Safety of Sea Transportation **3**(4), 393–400 (2009). https://doi.org/10.1201/9780203869345.ch74
23. Szlapczynska, J., Szlapczynski, R.: Preference-based evolutionary multi-objective optimization in ship weather routing. Applied Soft Computing **84**, 105742 (2019). https://doi.org/10.1016/j.asoc.2019.105742
24. The North Atlantic Ocean map: `https://commons.wikimedia.org/wiki/File:North_Atlantic_Ocean_laea_location_map.svg` (2012), licensed under the Creative Commons Attribution-Share Alike 3.0 Unported
25. United Nations Conference on Trade and Development (UNCTAD): Review of maritime transport (2022), `https://unctad.org/topic/transport-and-trade-logistics/review-of-maritime-transport`
26. Veneti, A., Makrygiorgos, A., Konstantopoulos, C., Pantziou, G., Vetsikas, I.A.: Minimizing the fuel consumption and the risk in maritime transportation: A bi-objective weather routing approach. Computers & Operations Research **88**, 220–236 (2017). https://doi.org/10.1016/j.cor.2017.07.010
27. Wang, H., Mao, W., Eriksson, L.: A Three-Dimensional Dijkstra's algorithm for multi-objective ship voyage optimization. Ocean Engineering **186**, 106131 (2019). https://doi.org/10.1016/j.oceaneng.2019.106131
28. Wang, X., Song, X., Sun, W.: Surrogate based trajectory planning method for an unmanned electric shovel. Mechanism and Machine Theory **158**, 104230 (2021). https://doi.org/10.1016/j.mechmachtheory.2020.104230
29. Yurt, R., Torpi, H., Mahouti, P., Kizilay, A., Koziel, S.: Buried Object Characterization Using Ground Penetrating Radar Assisted by Data-Driven Surrogate-Models. IEEE Access **11**(1), 13309–13323 (2023). https://doi.org/10.1109/ACCESS.2023.3243132
30. Zeraatgar, H., Ghaemi, H.: The Analysis of Overall Ship Fuel Consumption in Acceleration Manoeuvre Using Hull-Propeller-Engine Interaction Principles and Governor Features. Polish Maritime Research **26**, 162–173 (2019). https://doi.org/10.2478/pomr-2019-0018
31. Zis, T.P., Psaraftis, H.N., Ding, L.: Ship weather routing: A taxonomy and survey. Ocean Engineering **213**, 107697 (2020). https://doi.org/10.1016/j.oceaneng.2020.107697

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_31