Adaptive Surrogate-Assisted Optimal Sailboat Path Search Using Onboard Computers

Roman Dębski[0000−0003−3283−6032] and Rafał Drężewski[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 dynamic programming based optimal path search algorithm – studied in the context of *high-performance sailing* – is shown to be both effective and (energy) efficient. The key elements in achieving this – the fast and accurate physics-based surrogate model, the integrated refinement of the solution space and simulation model fidelity, and the OpenCL-based SPMD-parallelisation of the algorithm – are presented in detail. The included numerical results show the high accuracy of the surrogate model (relative approximation error medians smaller than 0.85%), its efficacy in terms of computing time reduction (from 39.2 to 45.4 times), and the high speedup of the parallel algorithm (from 5.5 to 54.2). Combining these effects gives (up to) 2461 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 (surrogate).

Keywords: simulation-based optimisation · surrogate model · optimal path planning · trajectory optimisation · heterogeneous computing

1 Introduction

High-fidelity (HF) simulations are often computationally too expensive to be used in a simulation-based optimisation. One obvious way to address this issue is to parallelise the algorithm. In many instances, however, it is at most a partial solution to the problem. This can be due to target platform constraints (e.g., limited number of processors/cores, energy consumption), an intrinsic strong sequential component of the algorithm (limited parallel speedup), 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 replace as many as possible high-fidelity simulations with evaluations of an auxiliary/approximation model – *the surrogate*. This auxiliary model should be a reasonably accurate representation of the HF-model and, at the same time (often much more importantly), remain computationally inexpensive.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

The discussed issue is of particular importance in the context of optimal path search algorithms based on dynamic programming. This approach usually leads to accurate results but is computationally very expensive, mostly due to search space size[1]. In a number of cases, including (hard) real-time embedded systems but also near-real-time autonomous robot or sailboat path planners, this is not acceptable. To the best of our knowledge, surrogate-assisted optimal path search based on dynamic programming has not been studied yet (section 2).

The aim of this paper, which is a significant extension of [3] and [4], is to present such an algorithm. When compared to these reference algorithms, it is equally accurate but, at the same time, is significantly faster and more energy-efficient, which is of primary importance for on-board systems (especially when at sea). The main contributions of the paper are:

1. an effective (time-constrained, fast, and accurate) physics-based surrogate model of sailboat motion (duration), defined in the spatial-domain rather than in the time domain, as the original, ODE-based model is (section 4.2),
2. surrogate-assisted, SPMD-parallel, dynamic programming based optimal path search algorithm, which incorporates an integrated refinement of the solution space and simulation model fidelity (section 4.3),
3. numerical results which demonstrate three important aspects of the algorithm: the accuracy of the surrogate-model, the surrogate-related speedup, and the SPMD-parallelisation capabilities (section 5).

The remainder of this paper is organised as follows. The next section presents related research. Following that, the search 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

Computer simulations are now widely used to verify engineering designs and to fine-tune the parameters of designed systems. Computer simulation-based approaches are also used in the optimal search problems when the performance measures are represented as a black box. In such cases, classical optimization methods cannot be used directly, which causes that AI-based approaches are often applied [14,17]. Unfortunately, accurate (high-fidelity) simulation models for such tasks are usually computationally expensive, which makes them often hard or even impossible to apply in practice. In such cases, approaches based on the so-called surrogates are used, i.e. simplified (low-fidelity) simulation models, which however reliably represent a complex simulation model of a system or process, and are much more computationally efficient [10,11].

In general, there are two types of surrogate simulation models: *approximation-based*, in which function approximation is constructed based on data sampled

[1] it can be significantly larger than 10^6

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

from accurate (high-fidelity) simulation models, and *physics-based*, in which surrogates are constructed based on simplified physical models of systems or processes [11].

The approximation-based surrogate models [11,19] are usually developed with the use of radial basis functions (RBF) [6], Kriging [5], polynomial response surfaces [12], artificial neural networks [7], support vector regression [16], Gaussian process regression [1], or multidimensional rational approximation [15].

Because physics-based surrogates are based on low-fidelity, simplified models of the system or processes, and thus contain some (simplified) knowledge about them, they usually require only a few accurate (high-fidelity) simulations runs to be reliably configured [11]. Due to the intrinsic knowledge about the simulated system/process, physics-based surrogates are also characterized by good generalization capabilities, so they can generate high-quality predictions of the accurate simulation model in the case of system designs or configurations not used during the training [11].

In the case of some optimization problems, evaluation of the objectives and constraints require data from physical experiments or numerical simulations, so such optimization problems are called data-driven [9]. To reduce the computation costs of data-driven problems, the surrogate-based models are used together with metaheuristic optimization algorithms, like evolutionary algorithms, particle swarm optimization and ant colony optimization. The excellent review of surrogate-assisted evolutionary algorithms can be found in [9].

The surrogate-based approach to modelling and simulation has gained a great popularity as a tool for lowering the computational costs of complex simulation models, especially in the area of engineering design problems. In the case of optimal trajectory search or path planning problems, surrogate-based methods are less frequently used [19]. The selected ones are briefly discussed below.

A surrogate based on uniform design and radial basis functions was used for mechatronic systems trajectory planning in [19]. In the proposed method, a conventional continue-time problem of optimal control was transformed into a nonlinear programming problem. The developed surrogate-based approach was applied to trajectory planning of an unmanned electric shovel. The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) was used as optimization algorithm.

A surrogate-based optimization method for full space mission trajectory design was proposed in [8]. As a data sampling method, the Optimized Latin Hypercube Design was used. The surrogate model was developed with the use of Kriging approach. After replacing the accurate dynamical model of the entire trajectory by a surrogate model, the multi-objective optimization problem was solved using NSGA-II genetic algorithm.

A multi-fidelity surrogate-based framework for global trajectory optimization was proposed in [18]. In the proposed approach, the Latin hypercube sampling was used for data sampling. Trajectories with initial conditions were evaluated using GPU parallel computing. Several approaches to the development of surrogate models were applied: the quadratic response surface, artificial neural network, and Kriging. The impact of each decision variable on the output param-

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

eter was assessed with the use of variance-based global sensitivity analysis (a numerical procedure based on Sobol's variance decomposition was applied). The verification of the proposed framework was performed with the use of a mission scenario including orbital transfer from a near-rectilinear halo orbit to a low lunar orbit.

A flight path planning surrogate model based on stacking ensemble learning was introduced in [20]. The proposed surrogate-based approach allowed for accurate, real-time calculation of the flight waypoint coordinates. In the proposed approach, a path planning analytical model was first used to generate numerous samples, which were then grouped based on selected parameters from the analytical model. The samples were then used to train base-learners using radial basis function neural networks. Finally, a complete surrogate model was constructed from base-learners using a stacking method.

An adaptive surrogate model for fast optimal transfer paths planning for spacecraft formation reconfiguration on libration point orbits was proposed in [13]. The uniform design of experiments was used to obtain the initial and updated samples. To construct the surrogate model, the Kriging and radial basis functions methods were used. The ant colony optimization algorithm was used for adaptive surrogate model optimization.

All the above-mentioned surrogate-based approaches for path planning or optimal trajectory search used approximation-based surrogates. Only a few of them applied GPU-based parallel computing. *The distinctive features of the optimal path search algorithm proposed in this paper include: the use of physics-based surrogate, using dynamic programming with integrated refinement of the solution space and simulation model fidelity, and* SPMD*-parallelization.*

3 Problem formulation

Consider a sailboat going from point $A(q_A, y_A)$ to $B(q_B, y_B)$, where (q_i, y_i) are the coordinates of the corresponding point in either the Cartesian or polar system [3,4]. The true wind vector field is given in the following way (see Fig. 1):

$$\boldsymbol{v}_t(q, y, t) = T_q(q, y, t)\ \hat{\boldsymbol{q}} + T_y(q, y, t)\ \hat{\boldsymbol{y}}, \tag{1}$$

where: $T_q(q, y, t)$, $T_y(q, y, t)$ are scalar functions, and $\hat{\boldsymbol{q}}$, $\hat{\boldsymbol{y}}$ are the unit vectors representing the axes of the corresponding coordinate system.

The problem domain consists of C^1-continuous $\widetilde{AB}$ paths which cover the given sailing area S_A (see Fig. 1). We assume that the explicit formula for the objective functional is unknown. Hence, each path $(y^{(i)})$ can be evaluated only through simulation, i.e.:

$$J[y^{(i)}] = \texttt{performSimulation}\ [y^{(i)}, \mathrm{cfg}\,(\boldsymbol{v}_t, \ldots)\ ], \tag{2}$$

where: J represents the given performance measure and $\mathrm{cfg}\,(\boldsymbol{v}_t, \ldots)$ – the simulator configuration.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

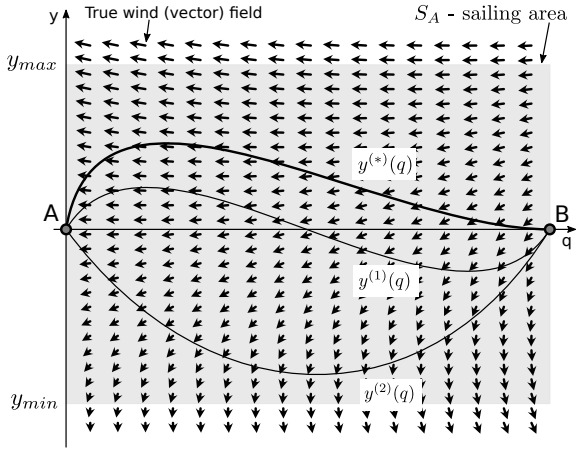

Fig. 1. Optimal sailboat path search problem: example admissible paths connecting points A and B, with $y^{(*)}(q)$ representing the optimal path [3,4].

Problem statement. The optimal sailboat path search problem under consideration can be defined as follows ([3,4]):

- find, among all admissible paths, the one with the best value of performance measure J;
- the values of J can be found only through simulation;
- only on-board, off-line computers can be used.

In the special case, when $J[y^{(i)}] = \Delta t[y^{(i)}]$, with Δt being the sailing duration, we get the *minimum-time problem.*

4 Proposed solution

The approach proposed in this paper is a "surrogate-accelerated" version of the one introduced in [3] and then extended in [4]. It is based on the following two main steps:

1. transformation of the continuous optimisation problem into a (discrete) search problem over a specially constructed finite graph (*multi-spline* [3]);
2. application of surrogate-assisted dynamic programming to find the approximation of the optimal path represented as a C^1-continuous cubic Hermite spline.

These two steps repeated several times form an adaptive version of the algorithm. Its key elements are:

- multi-spline based solution space and the SPMD-parallel computational topology it generates [3];

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

– effective (fast and accurate) surrogate model;
– integrated refinement of the solution space and simulation model fidelity that significantly reduces the time complexity of the reference algorithm.

They are discussed in the following subsections.

4.1 The solution space representation

A discretisation of the original problem leads to a grid, G, whose structure can be fitted into the problem domain [3]. An example of such a grid is shown on the left of Fig. 2. The grid is based on equidistant nodes grouped in rows and columns: four regular rows plus two special ones (containing the start (A) and the end (B) points) and four columns. The number of nodes in such a grid is equal to

$$|G| = n_c\,(n_r - 2) + 2 \tag{3}$$

where n_c and n_r are the numbers of columns and rows (including the two special ones), respectively.

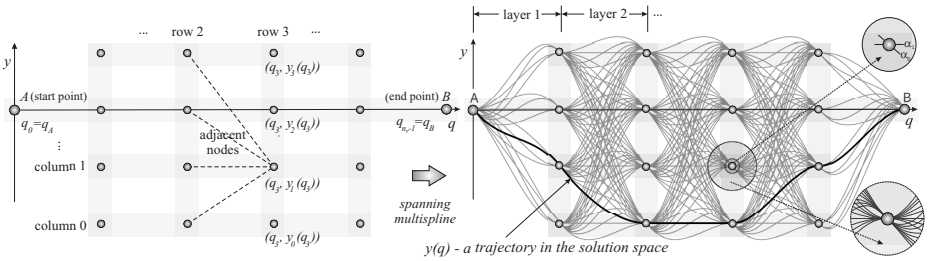

Fig. 2. Solution space representation: multi-spline spanned on regular grid G_{ex} [3]

After assigning n_{ts} additional values to every node of grid G, we obtain a new structure, G_{ex}, that can store not only the coordinates of each node but also the n_{ts} slopes (angles) of path segments which start/end in that particular node (see the right part of Fig. 2).

Joining the nodes from subsequent rows of G_{ex} by using cubic Hermite spline segments, we get a *multi-spline* which forms a discrete space of C^1-continuous functions (see Fig. 2). A detailed description of the multi-spline concept can be found in [3].

4.2 Surrogate-based performance measure

The proposed performance measure approximator (*low-fidelity* model, LFM) is an extended version of the estimator introduced in [4]. It is inspired by the *work-energy principle* that states that the work done by the forces on an object, $W_{AB} = \int_A^B \vec{F} \cdot d\vec{r}$, equals the change in its kinetic energy, $0.5\ m(v_B^2 - v_A^2)$, where

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

m stands for the sailboat mass. This principle can be used to transform the original problem from *time domain* to *spatial domain*, i.e.,

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

In our case, $F = F(s, v)$, or even, $F = F(t, s, v)$, but since we are building an approximation model, we can safely assume that:

$$\begin{cases} s_0 = s_A, \\ v_0 = v_A, \\ v_i^2 = v_{i-1}^2 + \frac{2}{m}F(s_{i-1}, v_{i-1})\Delta s_i \end{cases} \tag{4}$$

where: $i = 1, 2, \ldots, i_B$, $\Delta s_i = s_i - s_{i-1}$, and $s_{i_B} = s_B$. Having found the distribution of the (tangent) velocity along the sailing line, and assuming a constant value of F in each sub-interval, we can find the sailing duration. The details are shown in Algorithm 1.

Remark 1. Operating in the spatial domain is one of the key properties of the proposed approximator because the sailing duration can be calculated in a predefined number of steps N_s (e.g., 15) stemming from the (spatial) discretization of a path. In the original problem [3,4] – 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.

4.3 Surrogate-assisted optimal path search algorithm

The graph G_{ex}, on which the solution space (multi-spline) is spanned, is directed, acyclic (DAG) and has a layered structure. Since, at the beginning of the search process, the performance measure of each path segment is unknown, it has to be obtained from simulation. The *cost matrix* corresponding to the graph can be then computed using the *Principle of Optimality* [2].

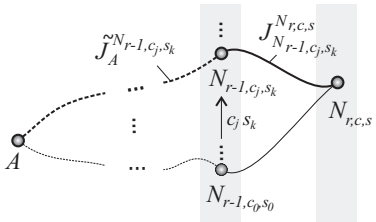

Fig. 3. Principle of Optimality in Dynamic Programming (see Eq. 5).

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

Algorithm 1: Surrogate model based approximation of a path segment performance measure

Input:
- s: the segment to be evaluated,
- v_S: the initial velocity of the sailboat (at the start point of s),
- t_S: the time of reaching the start point of s,
- m_S: the mass of the sailboat,
- t_{min}: the best (approximated) performance measure found so far,
- L_F: the "safety factor" for turning-on the (very) low-fidelity approximation,
- v_{min}: minimum non-zero velocity.

Output: the surrogate-based approximation of the performance measure of s

function `v_2`(v_1, dl, F, m_S)**:**
  $s_v \leftarrow v_1^2 + 2\,\frac{F\ dl}{m_S}$
  if $s_v > 0$ **then return** $\sqrt{s_v}$ **else return** 0

function `LFM_eval`(s, v_S, t_S, t_{min}, m_S, L_F)**:**
  $v_1 \leftarrow v_S$
  $v_{nz} \leftarrow$ `max`(v_S, v_{min}) `// last non-zero velocity`
  $dt \leftarrow 0$
  foreach *sub-segment* s_j *of segment* s **do**
    foreach x_i *in Gauss nodes for sub-segment* s_j **do**
      $dl_i \leftarrow$ the length of the i-th part of s_j
      $F_i \leftarrow$ the net-force for the current position and velocity
      $v_2 \leftarrow$ `v_2` (v_1, dl_i, F_i, m_S)
      if $v_2 > 0$ **then** $v_{nz} \leftarrow v_2$
      $\bar{v} \leftarrow 0.5\ (v_1 + v_2)$
      if $\bar{v} > 0$ **then** $v_{inv} \leftarrow \bar{v}^{-1}$ **else** $v_{inv} \leftarrow v_{nz}^{-1}$
      $dt \leftarrow dt + v_{inv}\ dl_i$
      if $(t_S + dt)\ t_{min}^{-1} > L_F$ **then return** (`length` (s) $(\sum_{k=0}^{i} dl_k)^{-1}\ dt$)
      $v_2 \leftarrow v_1$
  return $t_S + dt$

This principle can be expressed for an example path A-$N_{r,c,s}$ (see Fig. 3) in the following way [3]:

$$\widetilde{J}_A^{N_{r,c,s}} = \min_{c_j, s_k} \left(\widetilde{J}_A^{N_{r-1,c_j,s_k}} + J_{N_{r-1,c_j,s_k}}^{N_{r,c,s}} \right) \tag{5}$$

where: $c_j = (0, \ldots, n_c - 1)$, $s_k = (0, \ldots, n_{ts} - 1)$, $J_{N_s}^{N_e}$ is the cost corresponding to the path N_s-N_e (N_s - start node, N_e - end node), $\widetilde{J}$ represents the optimal value of J, and $N_{r,c,s}$ is the node of G_{ex} with "graph coordinates" $\langle row, column, tangent\_slope \rangle = \langle r, c, s \rangle$.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

Fig. 3 (a visualization of Eq. 5) presents the computation state in which the optimal costs of reaching all nodes in row $(r-1)$ are known – they were calculated in previous stages of this multi-stage process. The optimal cost of path A-$N_{r,c,s}$ is calculated by performing simulations for all spline segments that join node $N_{r,c,s}$, which is located in layer/row r, with nodes from the previous (i.e. $(r-1)^{\text{th}}$) row.

The multi-spline generated computational topology is reflected in the SPMD-structure of Algorithm 2 (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 path and its performance measure.

Algorithm 2: Adaptive, SPMD-parallel, surrogate-assisted optimal sailboat path search

Input:
- g_{AB}: initial (layered) grid with the start point, A, and the target point, B,
- $\boldsymbol{v}_t$: (true) wind vector field (see Eq.1),
- HFM: the sailboat movement simulator (high-fidelity model),
- LFM: the surrogate-model (low-fidelity model),
- C_M: "promising" segments (according to LFM) cut-off threshold.

Output: t_{min} - the minimum-time path

foreach *refinement* $g_{AB}^{(ref)}$ *of grid* g_{AB} **do**
  foreach *layer in* $g_{AB}^{(ref)}$ **do**
    @parallel foreach *entry point* e_p *of* $g_{AB}^{(ref)}$ *nodes* **do**
      if *not final refinement of* g_{AB} **then**
        $\mathbb{S}_{\text{ep}}^{(r)} \leftarrow$ representative subset of segments ending in e_p
        $(t_{min}^{(\texttt{LFM})}, s_{best}) \leftarrow \langle \min_{s\in\mathbb{S}_{\text{ep}}^{(r)}}, \arg\min_{s\in\mathbb{S}_{\text{ep}}^{(r)}} \rangle$ (`LFM_eval` (s)) `// using LFM`
        $t_{best} \leftarrow$ `HFM_eval` (s_{best}) `// using HFM`
      else
        $\mathbb{S}_{\text{ep}} \leftarrow$ all segments ending in e_p
        $\mathbb{K}_b \leftarrow k$ best from $\mathbb{S}_{\text{ep}}$, according to LFM
        $(t_{best}, s_{best}) \leftarrow \langle \min_{s\in\mathbb{K}_b}, \arg\min_{s\in\mathbb{K}_b} \rangle$ (`HFM_eval` (s))
        $\Delta t_{max}^{(k)} \leftarrow \max_{s\in\mathbb{K}_b} |\frac{\texttt{LFM\_eval}(s)}{t_{best}} - 1|$
        $\mathbb{R}_s \leftarrow \{\ \mathbb{S}_{\text{ep}} \setminus \mathbb{K}_b\ \} \bigcap \{\ s : |\ \frac{\texttt{LFM\_eval}(s)}{t_{best}} - 1\ | < C_M\ \Delta t_{max}^{(k)}\ \}$
        $(t'_{best}, s'_{best}) \leftarrow \langle \min_{s\in\mathbb{R}_s}, \arg\min_{s\in\mathbb{R}_s} \rangle$ (`HFM_eval` (s))
        if $t'_{best} < t_{best}$ **then** $(t_{best}, s_{best}) \leftarrow (t'_{best}, s'_{best})$
      save (t_{best}, s_{best}) `// save the best segm and its performance`

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

The next key element of the algorithm – *the integrated refinement of the solution space and simulation model fidelity* – is reflected by the conditional statement (lines 4-15). As the computation progresses, the search strategy changes from *mostly exploration* (lines 5-7) to *mostly exploitation* (lines 9-15). In the exploration phase, only the *representative subset* of segments is evaluated (line 5) and it is done with a *coarse grid* (i.e., $N_s = 15$). The exploitation phase is more complex. In the first step (line 9), *all* segments ending in a particular node are evaluated using a *fine grid* (i.e., $N_s = 30$). Following that (line 10), the k-best[2] candidate segments are evaluated using the HFM-model (i.e. the simulator). As a final step (lines 12-15), using the accuracy measure $\Delta t_{max}^{(k)}$, additional "promising" segments are selected (if there are any) and evaluated.

Complexity analysis. Algorithm 2 average-case *time complexity* is determined by the number of solution space refinements, n_i, the average number of force evaluations[3] for a single path segment, $\bar{n}_F$, and the number of such segments, $n_c n_{ts}^2 \left[(n_r - 3)\, n_c + 2 \right]$ (see Section 4.1). For the sequential version of the algorithm it can be expressed as:

$$T_s = \Theta \left(n_i \; n_r \; n_c^2 \; n_{ts}^2 \; \bar{n}_F \right). \tag{6}$$

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:

$$T_p = \Theta \left(n_i \; n_r \; n_c \; n_{ts} \left\lceil \frac{n_c \; n_{ts}}{p} \right\rceil \; \bar{n}_F \right). \tag{7}$$

In the same way as in reference algorithm [3], the Algorithm 2 *space complexity* formula, $\Theta \left(n_r \; n_c \; n_{ts} \right)$, arises from the solution space representation.

5 Results and discussion

To demonstrate the effectiveness of the algorithm, a series of experiments was carried out using a MacBook Pro[4] with macOS 12.2 and OpenCL 1.2. This system had two (operational) OpenCL-capable devices: Intel Core i5 @ 2.7 GHz (the CPU) and 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 surrogate-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 significant impact, especially in the exploration phase of the search process (detection of all potentially good segments). The results of its experimental evaluation are given in the form of a violin plot in Fig. 4.

[2] the value of k can be a constant or auto-adaptive variable
[3] 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
[4] Retina, 13-inch, Early 2015, with 16GB of DDR3 1867 MHz RAM

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

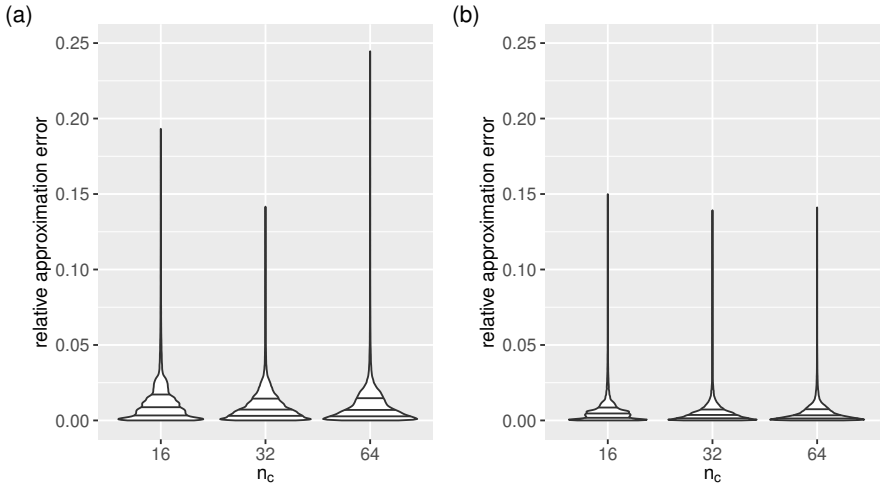

Fig. 4. Accuracy of the two surrogate models in use in the forms of their relative approximation errors (with reference to the high-fidelity model) for different numbers of multi-spline nodes (n_c): coarse model **(a)** vs. finer model **(b)**.

The plot shows the distributions of approximation relative errors, $|\frac{t_{\text{approx}}-t_{\text{sim}}}{t_{\text{sim}}}|$, for path segments from different search spaces. The highest recorded medians (0.83% for the coarse model and 0.45% for the finer model) confirm the very high accuracy of the proposed surrogate-model[5].

Efficacy of the surrogate-assisted search. This element was verified using the reduction of the number of force computations, $(\bar{n}_F^{(base)} - \bar{n}_F^{(sur)})/\bar{n}_F^{(base)}$, as the measure. To test the accuracy of $\bar{n}_F$ as the measure of the algorithm time complexity (see Eqns. 6 and 7), the duration of sequential computations, t_{sim}, was also measured, and then the Pearson correlation coefficient ($\rho(\bar{n}_F, t_{sim})$) was calculated. The result was equal to 0.999999997, which means (almost) perfect linear dependence between the two variables. The corresponding experimental results are given in Table 1 and Fig. 5. The observed $\bar{n}_F$ was reduced by 98%, which gives a significant improvement (more than 2 times) when compared to the results presented in [4].

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.

[5] the samples sizes (i.e., the number of segments) used to compute the distributions of errors were large: from 85 745 for $n_c = 16$ to 1 390 047 for $n_c = 64$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

Table 1. Efficacy of surrogate-assisted search: average numbers of force evaluations, $\bar{n}_F$, and execution times, t_{sim} (in seconds), for different n_c. The solution space (two refinements) with $n_r = 32$, $n_{ts} = 8$. Statistics from ten runs.

n_c	BASE MODEL					SURROGATE-ASSISTED MODEL				
	$\bar{n}_F$	t_{sim}				$\bar{n}_F$	t_{sim}			
		min	max	avg	sd		min	max	avg	sd
16	748.8	214.6	215.2	214.9	0.17	17.9	5.5	5.5	5.5	0.01
32	735.8	841.4	844.1	842.7	0.76	17.4	21.4	21.5	21.5	0.01
64	731.4	3344.0	3347.6	3345.8	1.20	17.0	83.9	84.0	84.0	0.02
128	731.7	13364.1	13413.4	13383.8	18.79	14.8	294.8	295.4	295.0	0.21

The execution times for different sizes of the solution space and the corresponding parallel-speedups are presented in Table 2 and Fig .5. Its maximum recorded

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	t_{sim}				speedup
	min	max	avg	sd	
16	0.973	0.985	0.980	0.003	5.6
32	1.854	1.865	1.859	0.003	11.5
64	3.360	3.379	3.368	0.006	24.9
128	5.432	5.445	5.439	0.004	54.2

value was 54.2 (see Table 1 and Fig. 5). With the reference point as the sequential search based on the full simulation, it gives in total 2461 times faster execution. Additionally, when compared to the results presented in [4], it gives us an execution time more than 3 times shorter.

6 Conclusions

It has been shown that the surrogate-assisted dynamic programming based optimal sailboat path planning algorithm can be both effective and (energy) efficient. The key elements in achieving this have been the fast and accurate physics-based surrogate model, the integrated refinement of the solution space (multi-spline) and simulation model fidelity, and the OpenCL-based SPMD-parallelisation of the algorithm.

The numerical results show the high accuracy of the surrogate model (the medians of relative approximation errors were smaller than 0.85%, see Fig. 4), its efficacy in terms of the reduction of computing time (from 39.2 to 45.4 times, see Table 1 and Fig. 5), and the high speedup of the parallel algorithm (its maximum observed value was 54.2, see Fig. 5). Combining these effects has given (up to) 2461 times faster execution time (see Fig. 5).

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

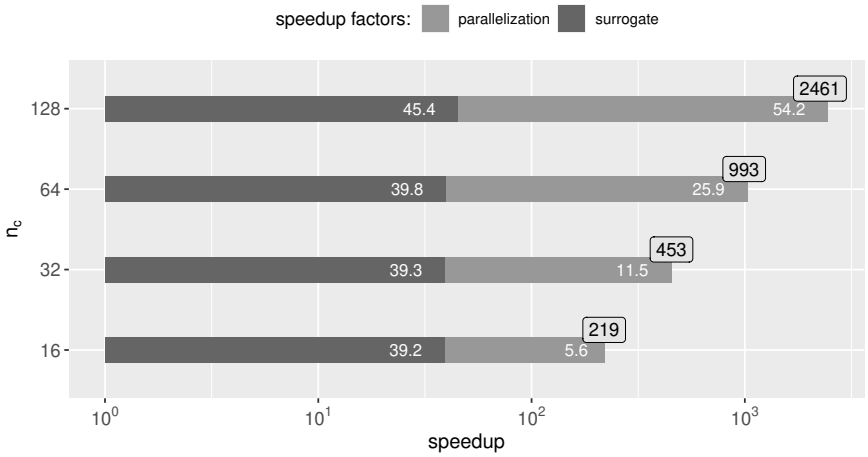

Fig. 5. Total speedup (boxed numbers at the end of each bar) and its factors: surrogate-model application (dark-grey part) and SPMD-parallelisation (light-grey part) for different n_c. The remaining parameters as in Table 1.

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 (surrogate). Further exploration of this idea could be the first possible future research direction. Another could be the algorithm space complexity reduction.

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.

References

1. Angiulli, G., Cacciola, M., Versaci, M.: Microwave devices and antennas modelling by support vector regression machines. IEEE Transactions on Magnetics **43**(4), 1589–1592 (2007). https://doi.org/10.1109/TMAG.2007.892480
2. Bellman, R., Dreyfus, S.: Applied Dynamic Programming. Princeton University Press, Princeton, New Jersey (1962)
3. Dębski, R.: An adaptive multi-spline refinement algorithm in simulation based sailboat trajectory optimization using onboard multi-core computer systems. Int. J. Appl. Math. Comput. Sci. **26**(2), 351–365 (2016). https://doi.org/10.1515/amcs-2016-0025
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

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30

5. Giunta, A., Watson, L.: A comparison of approximation modeling techniques-polynomial versus interpolating models. In: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, p. 4758. American Institute of Aeronautics and Astronautics (1998). https://doi.org/10.2514/6.1998-4758
6. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. Journal of Geophysical Research **76**(8), 1905–1915 (1971)
7. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, NJ (1998)
8. He, X., Zuo, X., Li, Q., Xu, M., Li, J.: Surrogate-based entire trajectory optimization for full space mission from launch to reentry. Acta Astronautica **190**, 83–97 (2022). https://doi.org/10.1016/j.actaastro.2021.09.030
9. Jin, Y., Wang, H., Chugh, T., Guo, D., Miettinen, K.: Data-driven evolutionary optimization: An overview and case studies. IEEE Transactions on Evolutionary Computation **23**(3), 442–458 (2019). https://doi.org/10.1109/TEVC.2018.2869001
10. 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
11. 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
12. Myers, R.H., Montgomery, D.C., Anderson-Cook, C.M.: Response Surface Methodology: Process and Product Optimization Using Designed Experiments. John Wiley & Sons (2016)
13. Peng, H., Wang, W.: Adaptive surrogate model-based fast path planning for spacecraft formation reconfiguration on libration point orbits. Aerospace Science and Technology **54**, 151–163 (2016). https://doi.org/10.1016/j.ast.2016.04.017
14. Pošík, P., Huyer, W., Pál, L.: A comparison of global search algorithms for continuous black box optimization. Evolutionary Computation pp. 1–32 (2012)
15. Shaker, G., Bakr, M.H., Sangary, N., Safavi-Naeini, S.: Accelerated antenna design methodology exploiting parameterized cauchy models. Progress In Electromagnetics Research B **18**, 279–309 (2009). https://doi.org/10.2528/PIERB09091109
16. Smola, A., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing **14**, 199–222 (2004)
17. Szłapczyński: Customized crossover in evolutionary sets of safe ship trajectories. Int. J. Appl. Math. Comput. Sci **22**(4), 999–1009 (2012)
18. Ueda, S., Ogawa, H.: Multi-fidelity approach for global trajectory optimization using GPU-based highly parallel architecture. Aerospace Science and Technology **116**, 106829 (2021). https://doi.org/10.1016/j.ast.2021.106829
19. 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
20. Yang, X.Z., Cui, Z.X., Qiu, X.Y.: Flight path planning surrogate model based on stacking ensemble learning. IOP Conference Series: Materials Science and Engineering **751**(1), 012038 (2020). https://doi.org/10.1088/1757-899x/751/1/012038

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_30