Balancing Between Time Budgets and Costs in Surrogate-Assisted Evolutionary Algorithms

Cedric J. Rodriguez[1], Peter A.N. Bosman[2], and Tanja Alderliesten[1]

Leiden University Medical Center, 2300 RC Leiden, The Netherlands
{c.j.rodriguez,t.alderliesten}@lumc.nl
Centrum Wiskunde & Informatica, 1090 GB Amsterdam, The Netherlands
peter.bosman@cwi.nl

Abstract. For many real-world multi-objective optimisation problems, function evaluations are computationally expensive, resulting in a limited budget of function evaluations that can be performed in practice. To tackle such expensive problems, multi-objective surrogate-assisted evolutionary algorithms (SAEAs) have been introduced. Often, the performance of these EAs is measured after a fixed number of function evaluations (typically several hundreds) and complex surrogate models are found to be the best to use. However, when selecting an SAEA for a real-world problem, the surrogate building time, surrogate evaluation time, function evaluation time, and available optimisation time budget should be considered simultaneously. To gain insight into the performance of various surrogate models under different conditions, we evaluate an EA with and without four surrogate models (both complex and simple) for a range of optimisation time budgets and function evaluation times while considering the surrogate building and surrogate evaluation times. We use 55 BBOB-BIOBJ benchmark problems as well as a real-world problem where the fitness function involves a biomechanical simulation. Our results, on both types of problems, indicate that a larger hypervolume can be obtained with SAEAs when a function evaluation takes longer than 0.384 seconds (on the hardware we used). While we confirm that state-of-the-art complex surrogate models are mostly the best choice if up to several hundred function evaluations can be performed, we also observe that simple surrogate models can still outperform non-surrogate-assisted EAs if several thousand function evaluations can be performed.

Keywords: Expensive optimisation · Surrogate-Assisted Evolutionary Algorithms · Real-world problems · Biomechanical simulations.

1 Introduction

Function evaluations of real-world multi-objective (MO) optimisation problems are often computationally expensive, costing multiple seconds to several hours. While evolutionary algorithms (EAs) are known to be effective at optimizing MO problems, they typically require well over a hundred thousand function evaluations to get (near-)optimal solutions (for non-trivial problems). This is not feasible for many expensive real-world problems. For this reason, MO surrogate-assisted EAs (SAEAs) have been introduced [20] in which many of the expensive-to-evaluate objective functions are replaced with quick-to-evaluate models (surrogates). The surrogate is typically built on all expensive function evaluations performed so far and is continuously updated during optimisation.

MO SAEAs, such as K-RVEA [10], AB-SAEA [35], CSEA [25], MOEA/D-EGO [39], or ADSAPSO [24], are often evaluated on DTLZ [13] benchmark problems with a fixed function evaluation budget (typically a few hundred). The efficacy of these SAEAs in scenarios that permit a "medium" budget of thousands of function evaluations has not been thoroughly investigated. Moreover, the time required for building and evaluating surrogate models is often not considered. This places non-surrogate-assisted EAs increasingly at a disadvantage as the expensiveness of function evaluations decreases, making it hard to assess the value of a surrogate model. For real-world problems, where optimisation time is frequently a limiting factor, it is crucial to assess surrogate models across various total optimisation time budgets instead of solely based on function evaluations.

SAEAs have been used for various real-world problems [19], e.g., energy-efficient design [6, 8, 14, 23, 33, 36, 38], motor manufacturing [29, 31], ship design [1, 17], automobile design [2, 30, 34], satellite design [28], antenna design [21, 37, 41], and energy and power [16]. In these works, total optimisation time is often disregarded as comparisons are performed at a fixed number of function evaluations or generations, possibly overestimating the *general* practical usability of SAEAs compared to non-surrogate-assisted EAs. Some works do provide the total optimisation time of different algorithms [23, 28, 36] and others do consider different optimisation time budgets [37]. No previous work has, however, varied *both* the optimisation time budget and the function evaluation time to systematically evaluate the performance of SAEAs for real-world problems.

We therefore emphasize comparing surrogate models under different total optimisation time budgets, following the strategy in [42], where surrogates based on Gaussian processes and polynomial regression were compared across varied time budgets. This however concerned single-objective optimisation, whereas we focus on MO optimisation which changes the balance between optimisation time and (surrogate) evaluation time budgets. By varying both the total optimisation time and the function evaluation time, we aim to cover the full spectrum of scenarios encountered in real-world applications, offering a more comprehensive evaluation of MO SAEAs so as to ultimately consider the following question: Given a spectrum of function evaluation times and optimisation time budgets, should an MO SAEA be considered and if yes, what type of surrogate model should be used?

The remainder of this paper is organized as follows: the MO SAEA and the surrogate models considered are described in Section 2. Our methodology and experimental setup are described in sections 3 and 4, respectively. In Section 5, the results on the benchmark problems are presented. In the sections 6 and 7, we apply the methodology to a real-world application. Finally, we present our conclusions in Section 8.

2 Background

2.1 MO SAEA

The MO SAEA that we consider is outlined in Algorithm 1. This algorithm is similar to [27] where each objective is modelled by a separate surrogate. In the initialization phase, a set of solutions is sampled using Latin Hypercube Sampling (LHS). These solutions are then evaluated using the expensive (true) function, and are subsequently used to build an initial surrogate model. The generational process can be decomposed into an inner cycle and an outer cycle. In the inner cycle, a population is initialized that contains all solutions previously evaluated with the true function. Selection and variation operators are applied, and offspring solutions are evaluated using the surrogate model instead of the true function. When the maximum number of surrogate evaluations for an inner cycle is reached, the inner cycle is terminated. All surrogate-evaluated solutions are then considered in the outer cycle as potential candidate solutions for evaluation with the true function. The most promising solutions are then selected as described in Section 4.2 and evaluated using the true function. Subsequently, a surrogate model is rebuilt based on all true-evaluated solutions so far, and the next iteration of the inner cycle commences.

Algorithm 1 MO SAEA

```
Input:  $N$ = Number of initial solutions; $FE_{max}^{true}$ = maximum number of true
        function evaluations; $FE_{max}^{surrogate}$ = maximum number of surrogate
        evaluations in each inner cycle; $\mu$ = number of new selected solutions
        in the outer cycle
Output: $A$ = Archive of all true evaluated solutions
Initialize population $P$ of size $N$ using random sampling
Evaluate all solutions in $P$ using the true function
Initialize archive $A$ of all true evaluated solutions with all solutions in $P$
Set number of true function evaluations $FE^{true}$ to $N$
while $FE^{true} < FE_{max}^{true}$ do                                  ▷ Outer Cycle
    Build surrogate model $M$ based on $A$
    Continue to optimize $P$ using surrogate                           ▷ Inner Cycle
    model $M$ in EA with $FE_{max}^{surrogate}$
    Select $\mu$ solutions from all surrogate evaluated solutions
    Evaluate $\mu$ selected solutions with true function
    $FE^{true} = FE^{true} + \mu$
    Add true evaluated solutions to archive $A$
end while
Return $A$
```

2.2 Surrogates

We consider two commonly adopted surrogates of varying complexity as well as two surrogates that are not considered often, but are of very low complexity.

Gaussian processes The most popular surrogate, and often considered to be state-of-the-art for expensive optimisation, is Gaussian process regression (e.g., Kriging, also much used in Bayesian optimisation) [35]. It is defined as follows:

$$\hat{y}^{kriging}(x) = \beta + r^T(x)R^{-1}(y - F(\beta, x)\beta) \tag{1}$$

where β are the coefficients, R is the correlation matrix between all the training samples, $r^T(x)$ represents the correlation vector between the solution x and the training samples, y represents the fitness values of the training set of solutions, and $F(\beta, x)$ is the regression model. [10] This surrogate model requires maximizing a likelihood function for the hyperparameters which is computationally expensive.

Radial basis functions This surrogate is a weighted aggregation of basis functions $\Psi(\cdot)$:

$$\hat{y}^{rbf}(x) = \sum_{i=1}^{n} w_i \Psi_{i,j} \tag{2}$$

In this paper, we utilize Gaussian radial basis functions, where the centre of each Gaussian corresponds to each solution that the surrogate model is built upon. The influence of each basis function is calculated using the squared Euclidean distance from the point being evaluated, i.e., $\Psi_{i,j} = e^{-\gamma d_{i,j}^2}$ where $d_{i,j}$ represents the Euclidean distance between the i-th and j-th solutions, and γ is a scalar that determines the spread of the basis functions. The weights are determined through generalized linear regression, which involves computing the pseudo-inverse of the Ψ matrix. [7] This is much faster than what is needed to build the Kriging model.

K-nearest neighbours linear regression We also consider a simpler surrogate that leverages linear regression. The local structure of the problem landscape is however captured by using only the k-nearest neighbours (*knn*) of the solution of interest.

$$\hat{y}^{lr\text{-}knn}(x) = \text{mode}\,\{y_i \mid (x_i, y_i) \in \text{Nearest}_k(x, \mathcal{D})\} \tag{3}$$

where $\mathcal{D}$ is the training dataset consisting of pairs (x_i, y_i), where x_i are the decision vectors and y_i are the fitness values. The function $\text{Nearest}_k(x, \mathcal{D})$ returns the decision vectors and the fitness values of k-nearest neighbours using Euclidean distance in the decision space of x in $\mathcal{D}$. The mode function determines the desired interpolation function between these neighbours. This model, we use linear regression as the interpolation function defined as:

$$\hat{y}^{lr\text{-}knn}(x) = \beta_0 + \sum_{q=1}^{D} \beta_q x_q \tag{4}$$

where β_0 is the intercept, β_q are the coefficients (slopes) in each problem dimension, D is the number of problem dimensions, and x_q are the decision variables.

Nearest neighbour Finally, we also consider one of the simplest surrogates - solely considering the nearest neighbour. This surrogate model is defined as:

$$\hat{y}^{nn}(x) = \text{Nearest}(x, \mathcal{D}) \tag{5}$$

where $\mathcal{D}$ is the training dataset and the function $\text{Nearest}(x, \mathcal{D})$ returns the fitness of the nearest neighbour of x in $\mathcal{D}$ in terms of Euclidean distance in decision space.

2.3 Reference Vector Guided EA

The baseline MOEA that we use in this work, is a Reference Vector Guided Evolutionary Algorithm (RVEA) [9]. In RVEA, offspring solutions are generated using simulated binary crossover [11] and polynomial mutation, comparable to what is used in other MOEAs like NSGA-III [12]. The selection of the parent population for the subsequent generation of offspring, is performed using a technique known as angle penalized distance, which is guided by reference vectors. RVEA is often used within a surrogate-assisted approach, including some state-of-the-art expensive optimisation approaches for MO optimisation [10,35], which is why we consider this EA here as well.

2.4 MAMaLGaM

For the real-world application, we additionally consider an estimation-of-distribution algorithm (EDA), in particular iMAMaLGaM-X with a multivariate joint Gaussian distribution [4]. In part, this is because an EDA can be considered a different type of model-based EA, but also, this algorithm was previously used to optimize a simpler version of the real-world problem [26] that we also consider in this work. In iMAMaLGaM-X, solutions are selected from the parent population and elitist archive, and are organized into a user-defined number of clusters (K). A Gaussian distribution is estimated and adaptively scaled for each cluster, from which new offspring solutions are sampled. For more details, see [4].

3 Methodology

To get insights into the impact of the computational expensiveness of the true fitness function as well as the computational expensiveness associated with the surrogate model, we run EAs and SAEAs using benchmark problems. For benchmark problems, the true function evaluation time is in the order of milliseconds and is thus negligible. We use this fact to subsequently estimate the time an algorithm would have taken if the true fitness function had been more expensive.

To this end, for each combination of problem instance and algorithm, we run an optimisation process. After each true function evaluation FE^{true}, the measured execution time t^m_{FE-1} is recorded. Because the time taken to evaluate a benchmark problem is negligible, the execution time t^m_{FE-1} represents time taken by the algorithm in between true function evaluations. For EAs, this is the time required for variation and selection and for SAEAs, this is the time for surrogate building and optimisation using the surrogate model.

We can now estimate the total optimisation time $\hat{t}_{FE}$ of an algorithm using

$$\hat{t}_{FE} = t^m_{FE-1} + \delta FE \tag{6}$$

where the execution time t^m_{FE-1} is the surrogate building time and surrogate evaluation time, FE is the number of true function evaluations and δ is the true function evaluation time.

In this paper, we consider $\delta = 0.0001 \cdot 2^i$ minutes for $i \in \{0, 1, \ldots, 20\}$ and a total optimisation time budget of $16 \cdot 2^j$ minutes for $j \in \{0, 1, \ldots, 11\}$.

4 Experimental setup

In this section, we describe the considered benchmark problems, algorithm settings used, and details regarding how we analysed the results.

4.1 Benchmark problems

We consider the bi-objective BBOB functions (BBOB-BIOBJ) [5] with 20 continuous problem variables from the COCO framework [18]. We only consider the first instance of the 55 problems, since we are not exploring rotation and translational invariances of algorithms. Each BBOB-BIOBJ problem is composed of a pair-wise combination of 10 single objective BBOB functions. The algorithms are initialized in the default COCO ranges: −100 to 100. The termination criteria are set to 10,000 true function evaluations or a maximum runtime of 3 days, whichever comes first. We execute 15 repeats per optimisation. A high-performance computing cluster is used, composed of AMD EPYC 7H12 nodes, each containing 2 GB of memory and 64 cores with a 2.6GHz clock speed per node. Octave version 7.3.0 and the COCO framework [18][1] are used to perform all experiments.

4.2 Algorithm settings

In this work, we evaluate four SAEAs where two SAEAs utilise state-of-the-art surrogates, namely *Kriging* from [10] and *rbf* from [22] as implemented in [32]. We also introduce two quick-to-compute surrogates namely *nn* and *lrknn* as described in Section 2.2. As a baseline, we also consider *rvea* without any surrogate assistance. This leads to five different algorithms in total: *kriging-rvea*, *rbf-rvea*, *nn-rvea*, *lrknn-rvea*, and *rvea*.

[1] `https://github.com/numbbo/coco`

All algorithms start with an initial population of 32 solutions, sampled using LHS. Each repeat is completely independent, using a different seed number. This is the first multiple of 16 (available parallel cores for the real-world application) larger than the problem's dimension of 20. After a surrogate optimisation cycle of 2000 surrogate evaluations using *rvea*, for all SAEA variants except *kriging-rvea*, the potential solutions for true evaluation are determined by pre-selecting the non-dominated solutions in terms of the surrogate fitness values. For *Kriging*, we utilise the acquisition function for the pre-selection as was described in [35] and was implemented for Bayesian optimisation in the PlatEMO framework [32]. This acquisition function prioritizes solutions with uncertain fitness estimation at the beginning of the optimisation and solutions with good fitness estimations at the end of the optimisation. This is reported in [35] to provide higher hypervolume on the DTLZ [13] and UF [40] bi-objective benchmark problems compared to [10]. As suggested in [35], we also use an archive management method for the *Kriging* surrogate, limiting the archive size to 320 solutions.

For all SAEA variants, we then randomly select μ solutions from this pre-selection. In this work, we have used $\mu = 1$ so that the model is updated as frequently as possible (every new true evaluation). Finally, *rvea* is used as implemented by [9] in PlatEMO [32]. In this implementation, the parameter α prioritizing convergence is set to $100,000$ and the population size is set to 32.

For the *lrknn* surrogate, $k = 32$ is used to fit a linear slope through the training samples at the beginning of the optimisation and to capture the local structure of the problem landscape at the end of the optimisation. Finally, *nn-rvea* has no additional parameters to set.

4.3 Evaluation of results

For each (estimated) optimisation time budget and expensive function evaluation time, the two algorithms are selected that exhibit the highest and second-highest median hypervolume over the 15 repeats on a BBOB-BIOBJ problem. The reference point is set to $[1, 1]$, as suggested by [18]. We also evaluate whether there is a statistically significant difference between the hypervolume of the best and second-best performing algorithms using a Wilcoxon test. When considering 55 BBOB-BIOBJ problems, 12 optimisation time budgets, and 21 expensive function evaluation times, this leads to 13,860 statistical tests. We consider statistical significance to be $p \leq 0.05$ and correct this using the Bonferroni correction. To get an aggregated view of an algorithm's performance over all BBOB-BIOBJ problems, we estimate, for each combination of optimisation time budget and expensive function evaluation time, the frequency that a particular algorithm has been selected as the best-performing algorithm.

5 Results on Benchmark Problems

The best performing algorithms on the BBOB-BIOBJ problems 1 to 19 are visualised using a heatmap, see Figure 1. The BBOB-BIOBJ problems 20 to 55 can be found in the Supplementary Material[2]. Because the first 32 function evaluations correspond to the initial randomly sampled solutions in the population, we shaded the regions where the total optimisation time is smaller than 32 times the function evaluation time. The regions with high optimisation times and low function evaluation times are also shaded because due to the termination criteria of 10,000 evaluations, no data was gathered for this region.

It can be observed in Figure 1 and Figure 2 that *rvea* typically has the highest median hypervolume for function evaluation times below 0.0064 minutes (0.384 seconds). Furthermore, *rvea* only has a higher median hypervolume if the total optimisation time is below 32 minutes. This is mostly due to the limit of 10,000 evaluations. If this budget was set larger, *rvea* is expected to also outperform all other algorithms for the longer total optimisation times.

As expected, the current state-of-the-art surrogate models such as *kriging* and *rbf* lead to higher median hypervolumes obtained with the SAEA for smaller budgets of function evaluations, which corresponds to the left-upper envelope of the graphs, see Figure 2. In particular, for each row on the diagonal of the left-upper envelope, the first three columns represent up to 80 function evaluations. Interestingly, for some BBOB-BIOBJ problems (e.g., 7, 8, 13, 15) in Figure 1, *rvea* still obtained a higher median hypervolume in these regions. A likely reason for this is that the considered surrogates are not effective at modelling the problem with such a limited number of function evaluations. Finally, and interestingly, on almost all problems the use of "simple" surrogate models in our SAEA, i.e., *nn-rvea* and *lrknn-rvea*, obtains a higher median hypervolume for "medium"-range expensive settings in which up to 10,000 function evaluations can be performed, with 10,000 evaluations corresponding to the bottom-right envelope in Figure 2. Only in a few cases does *rvea* become the dominating algorithm again when 10,000 evaluations are used.

6 Real-world Application: A Biomechanical Simulation

While the BBOB-BIOBJ problems are certainly of importance and value, they may still differ from a real-world application in which a true expensive, and often complex, optimisation problem needs to be solved.

We therefore also consider a real-world optimisation problem in this work that is computationally demanding: biomechanical simulation optimisation for the purpose of deformable medical image registration in radiation treatment of cervical cancer. In radiation treatment, multiple medical images of the same patient are acquired, see Figure 3. Given that organs in the pelvic region are highly deformable for various reasons, there is often an inherent mismatch between the shapes of organs in any pair of medical images. Deformable image registration is aimed at finding the physically correct spatial mapping between two medical image pairs to transfer spatial information, such as dosimetric data.

In this work, we consider a biomechanical approach to deformable image registration in which the organs in one image are contoured by a medical pro-

[2] `https://zenodo.org/records/10992139`

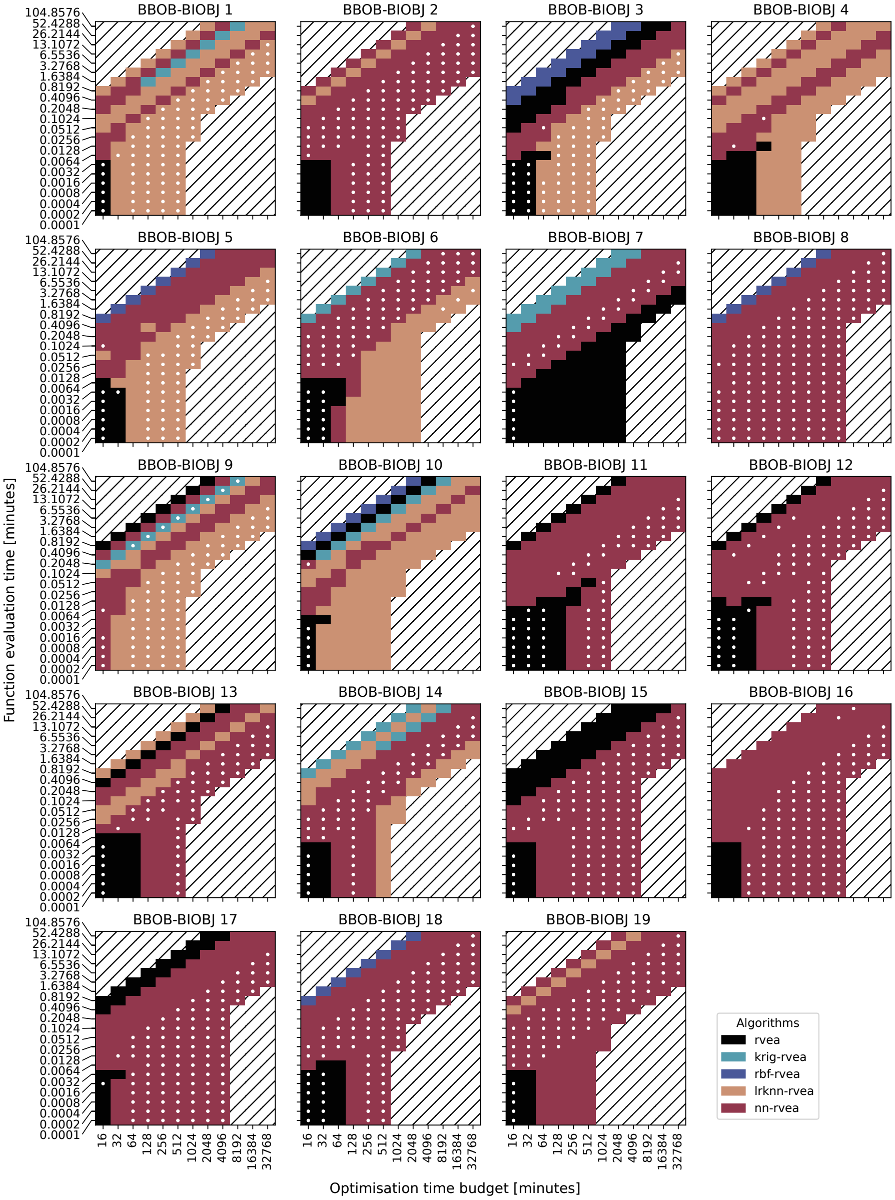

Fig. 1. Heatmap per BBOB-BIOBJ problem, visualising the algorithm with the highest median hypervolume given an optimisation time and function evaluation time. The white dots indicate if there is a statistical difference between the best and second-best performing algorithm.

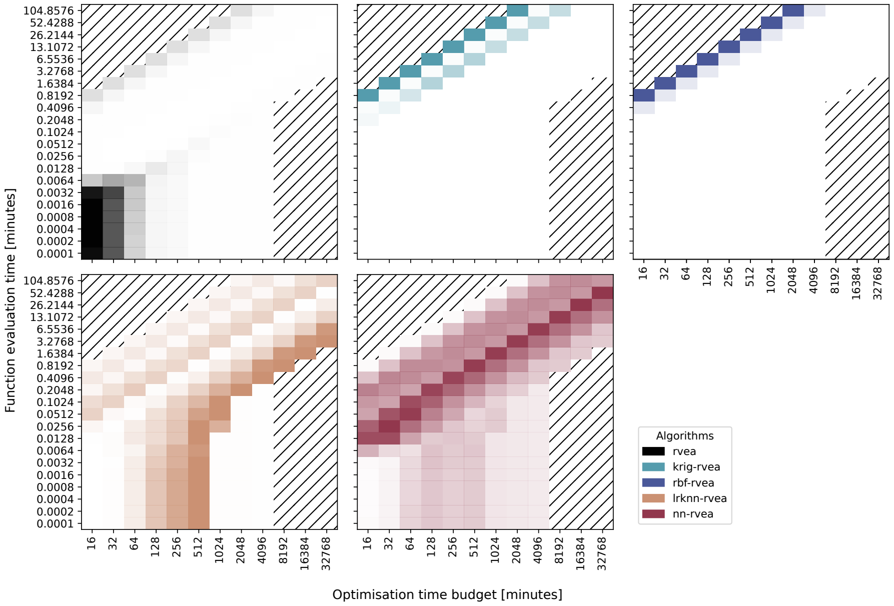

Fig. 2. Heatmap per algorithm, where a higher colour intensity indicates, given an optimisation time and function evaluation time, a higher frequency of highest median hypervolume over all BBOB-BIOBJ problems.

fessional and are subsequently used to construct a mesh representation that can be used inside a finite element method (FEM) simulation. Tissue characteristics, particularly related to elasticity, are adhered to different parts of the mesh. Subsequently, forces are defined that are applied to specific regions in the mesh. A FEM simulation then adjusts the mesh nodes according to the forces applied. Consequently, the underlying organ contours are deformed.

The optimisation task consists of finding free simulation parameters such that FEM simulation results in a mesh transformation that ensures the transformed organ contours are aligned with those in the second image. Each function evaluation necessitates a full FEM simulation, which is a computationally expensive operation.

In this work, we focus on the deformable image registration of 3D pelvic computed tomography (CT) scans for patients undergoing external beam radiation treatment for cervical cancer. The treatment planning involves acquiring two 3D CT scans: one with an empty bladder and another with a full bladder. In the following section, we provide an overview of the process. For more details, we refer the interested reader to the Supplementary Material.

Contouring and mesh generation Initially, a 3D CT scan with an empty bladder is used for contouring important structures such as the body, bladder, bones, cervix-uterus, bowel, and rectum. Subsequently, the contoured images are converted into triangular surface meshes for each organ, which are then inte-

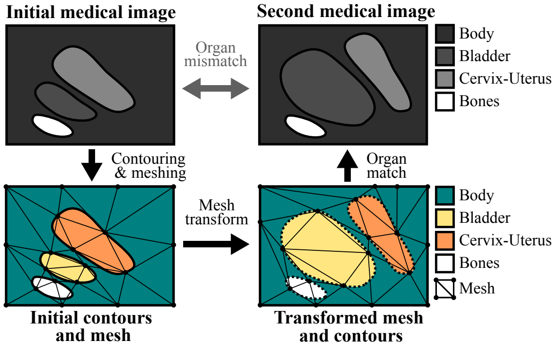

Fig. 3. An illustrative representation of deformable image registration, where the top two images represent medical images of the same patient with a difference in bladder volume and the bottom two images depict the contoured organs overlayed with the finite element method mesh that is deformed to align the two images.

grated into a unified tetrahedral mesh for each patient. Each tetrahedron within this mesh is associated with a specific organ, facilitating a detailed anatomical representation of the patient with an empty bladder.

Mesh transformation - biomechanical FEM simulation The organs in the empty bladder scan are deformed according to the FEM simulation. To optimize the match between the so-deformed organs and those in the full bladder CT scan, the parameters corresponding to various forces to be applied to the mesh are optimized. Specifically, 19 continuous simulation parameters are considered.

Organ comparison To determine the quality of the resulting FEM-based deformation, we utilize the Dice similarity coefficient (DSC) score [15]. The DSC score evaluates the overlap between the two volumes, with scores ranging from 0 (no volumetric overlap) to 1 (perfect volumetric overlap). However, relying solely on DSC scores could still result in large and unrealistic organ deformations. Therefore, a second objective, namely deformation energy, is considered that is related to the magnitude of the resulting deformation. Specifically, this measure quantifies the energy required for organ deformation, calculated in Joules by summing the energy needed to deform each tetrahedral element in the patient's mesh based on its biomechanical properties. This bi-objective approach ensures that we obtain results representing different trade-offs between accurate organ registration and physical realism in the deformation process which can subsequently be inspected by a medical professional so as to ultimately decide which deformation is the preferred one to use.

Problem settings, algorithm settings and result evaluation Optimizing biomechanical simulations for seven patients involves stopping at either $10,000$ simulations or after five days. Since this function evaluation time is not negligible compared to the surrogate optimisation time, we directly record the time duration of the surrogate optimisation cycles (excluding the function evaluation times). We run 16 simulations in parallel using 16 cores and repeat each optimisation 10 times. The same algorithms are used as in Section 4.2 as well as iMAMaLGaM-X. The settings for iMAMaLGaM-X are based on the guidelines in literature [3], except for the population size, which is set to 32 (equal to *rvea*). A high-performance computing system is used, composed out of nodes, each containing 2 GB of memory per core and 16-128 Intel E5/Gold series cores. The optimisation uses Octave version 7.3.0 and the simulation uses SOFA[3] v.21.12. The results are evaluated using the same procedure as in Section 4.3. The hypervolume is calculated using the reference point of $[0, 5000]$, which has been established empirically.

7 Results on Real-world Application

The best performing algorithms are visualised per patient using a heatmap, see Figure 4. Here, we see, similar to the benchmarks, that *rvea* performs best for a function evaluation time under 0.0064 minutes (0.384 seconds). Furthermore, *kriging* and *rbf* surrogates seem to only show a benefit for very small evaluation budgets. Interestingly, iMAMaLGaM-X is consistently effective for intermediate evaluation budgets which constitutes a large portion of the heatmap and the use of the *nn* surrogate in RVEA is best around the 10,000 evaluations budget limits. No statistically significant differences were found.

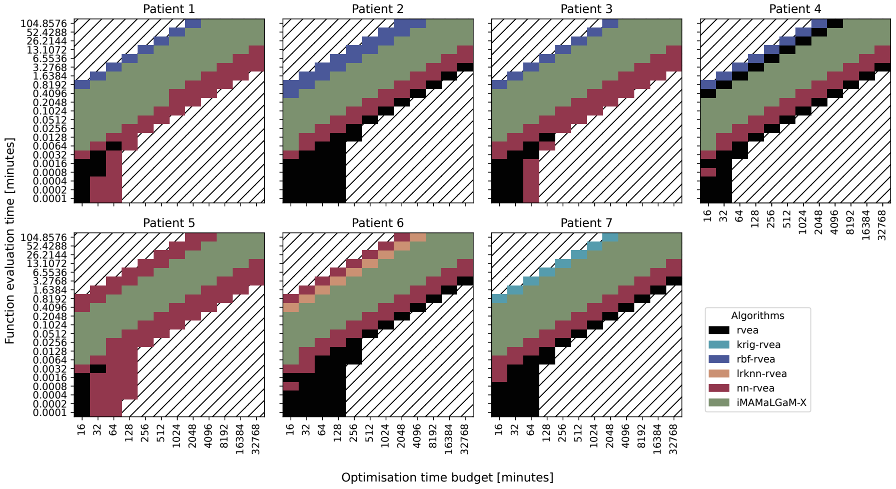

Fig. 4. Heatmap per patient, visualising the algorithm with the highest median hypervolume given an optimisation time and function evaluation time.

[3] https://github.com/sofa-framework/sofa

The average simulation in the experiments took approximately 48 seconds. With 16 parallel cores, this results in an effective simulation time of 3 seconds (0.05 minutes). Figure 4 and 5 indicate that, for this function evaluation time and optimisation budget of 4 hours (240 minutes), *nn-rvea* leads to the highest hypervolume on all 7 patients. Nevertheless, the border to iMAMaLGaM-X and *rvea* often is one block away, meaning that a slight change in function evaluation time or optimisation budget, could change the best choice of optimizer. For example, say we expect the simulation to increase in simulation time due to increased simulation complexity. This would mean, although *nn-rvea* shows a higher hypervolume now, it could then be better to select iMAMaLGaM-X as optimizer for this real-world application. When looking at the approximation front for Patient 1 in Figure 6, we see that both *nn-rvea* and iMAMaLGaM-X, find evenly distributed solutions along the front with good DSCs of up to 0.8.

While results for different patients do not differ hugely, for different evaluation budgets and different patients, sometimes different algorithms appear as the best choice. The general approach presented in this work can therefore not be used to make patient-specific recommendations about the best SAEA to use, but rather gives generic insight into which algorithms are of interest for a certain total optimisation budget (i.e., a pre-selection). However, if desired, further refinement of algorithm selection could be done using (online) landscape analysis of both the true and surrogate fitness landscapes, as for instance in [27].

8 Conclusions

In this work, we addressed a mostly overlooked aspect of different types of time considerations in evaluating multi-objective surrogate-assisted evolutionary algorithms (MO SAEAs), by incorporating surrogate building and surrogate evaluation times alongside true function evaluation times and total optimisation time budgets. We compared four surrogate models across 55 benchmark problems and a real-world medical application. We demonstrated that MO EAs seemed to be more effective than MO SAEAs for functions that have an evaluation time that is shorter than 0.384 seconds. Of course, this particular number is strongly related to the used hardware and types of surrogate models and their implementation. However, the approach used here is a general one that can be repeated on other hardware and with different implementations within a different context (e.g., with more parallel computing power) to obtain problem- and computational-setup specific results if need be.

Our results showed that while state-of-the-art surrogate models like Kriging and radial basis functions excel up to several hundred function evaluations, simpler models such as nearest neighbour and linear regression emerge as effective options in case more function evaluations can be performed (i.e., less expensive problems or larger available computational power/budget). This work also motivates that more research should be done on reducing surrogate building and evaluation time (especially as the number of training samples increase) or the use of sparse surrogate models.

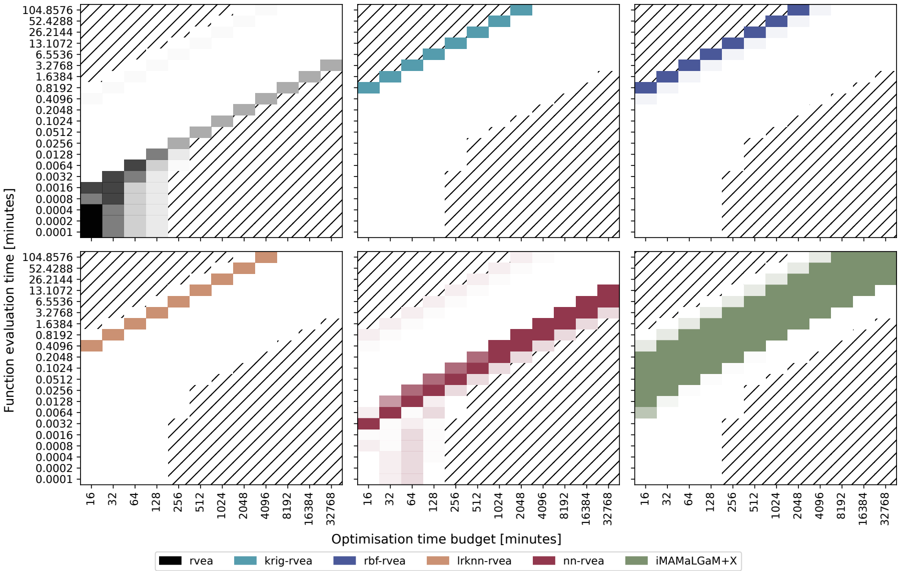

Fig. 5. Heatmap per algorithm, where a higher colour intensity indicates a higher frequency of highest median hypervolume over all patients.

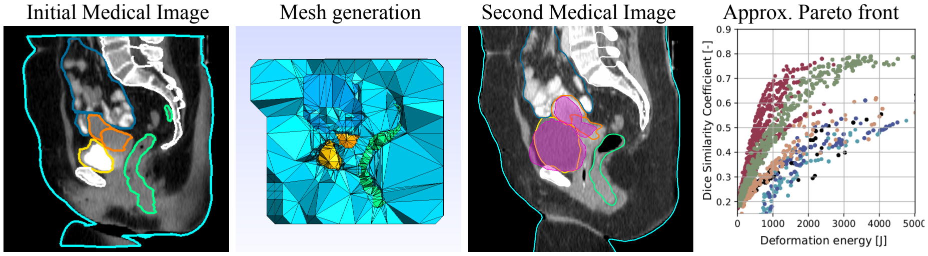

Fig. 6. Optimisation process and results of patient 1, where pink visualizes the deformed bladder and cervix-uterus corresponding to the solution with the highest DSC score after optimisation.

Acknowledgments. The research is part of the research programme Open Technology Programme with project number 15586, which is financed by the Dutch Research Council (NWO), Elekta (Elekta Solutions AB, Stockholm, Sweden), and Xomnia (Xomnia B.V., Amsterdam, The Netherlands). Further, the work is co-funded by the public-private partnership allowance for top consortia for knowledge and innovation (TKIs) from the Ministry of Economic Affairs.

Disclosure of Interests. All authors are involved in one or more projects supported by Elekta AB, Stockholm, Sweden. Elekta had no involvement in the study design, the data collection, analysis and interpretation, or the writing of the paper.

References

1. Ayob, A.F.M., Ray, T., Smith, W.F.: Beyond hydrodynamic design optimization of planing craft. Journal of Ship Production **27**(1), 1–13 (2011)
2. Bhattacharjee, D., Ghosh, T., Bhola, P., Martinsen, K., Dan, P.K.: Data-driven surrogate assisted evolutionary optimization of hybrid powertrain for improved fuel economy and performance. Energy **183**, 235–248 (2019)
3. Bosman, P.A.: On empirical memory design, faster selection of bayesian factorizations and parameter-free gaussian edas. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. pp. 389–396 (2009)
4. Bosman, P.A.: The anticipated mean shift and cluster registration in mixture-based edas for multi-objective optimization. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation. pp. 351–358 (2010)
5. Brockhoff, D., Auger, A., Hansen, N., Tušar, T.: Using well-understood single-objective functions in multi-objective black-box optimization test suites. Evolutionary Computation **30**(2), 165–193 (2022)
6. Chegari, B., Tabaa, M., Simeu, E., Moutaouakkil, F., Medromi, H.: Multi-objective optimization of building energy performance and indoor thermal comfort by combining artificial neural networks and metaheuristic algorithms. Energy and Buildings **239**, 110839 (2021)
7. Chen, G., Zhang, K., Xue, X., Zhang, L., Yao, C., Wang, J., Yao, J.: A radial basis function surrogate model assisted evolutionary algorithm for high-dimensional expensive optimization problems. Applied Soft Computing **116**, 108353 (2022)
8. Chen, X., Yang, H.: Integrated energy performance optimization of a passively designed high-rise residential building in different climatic zones of china. Applied energy **215**, 145–158 (2018)
9. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation **20**(5), 773–791 (2016)
10. Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., Sindhya, K.: A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **22**(1), 129–142 (2016)
11. Deb, K., Agrawal, R.B., et al.: Simulated binary crossover for continuous search space. Complex Systems **9**(2), 115–148 (1995)
12. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2013)
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). vol. 1, pp. 825–830. IEEE (2002)
14. Dhariwal, J., Banerjee, R.: An approach for building design optimization using design of experiments. In: Building Simulation. vol. 10, pp. 323–336. Springer (2017)
15. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)
16. Dong, Q., Wang, C., Peng, S., Wang, Z., Liu, C.: A many-objective optimization for an eco-efficient flue gas desulfurization process using a surrogate-assisted evolutionary algorithm. Sustainability **13**(16), 9015 (2021)

17. Habib, A., Singh, H.K., Ray, T.: A multiple surrogate assisted evolutionary algorithm for optimization involving iterative solvers. Engineering Optimization **50**(9), 1625–1644 (2018)
18. Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., Brockhoff, D.: COCO: A platform for comparing continuous optimizers in a black-box setting. Optimization Methods and Software **36**, 114–144 (2021). `https://doi.org/https://doi.org/10.1080/10556788.2020.1808977`
19. He, C., Zhang, Y., Gong, D., Ji, X.: A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. Expert Systems with Applications p. 119495 (2023)
20. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation **1**(2), 61–70 (2011)
21. Koziel, S., Bekasiewicz, A.: Scalability of surrogate-assisted multi-objective optimization of antenna structures exploiting variable-fidelity electromagnetic simulation models. Engineering Optimization **48**(10), 1778–1792 (2016)
22. Li, J., Wang, P., Dong, H., Shen, J.: Multi/many-objective evolutionary algorithm assisted by radial basis function models for expensive optimization. Applied Soft Computing **122**, 108798 (2022)
23. Li, K., Pan, L., Xue, W., Jiang, H., Mao, H.: Multi-objective optimization for energy performance improvement of residential buildings: A comparative study. Energies **10**(2), 245 (2017)
24. Lin, J., He, C., Cheng, R.: Adaptive dropout for high-dimensional expensive multiobjective optimization. Complex & Intelligent Systems **8**(1), 271–285 (2022)
25. Pan, L., He, C., Tian, Y., Wang, H., Zhang, X., Jin, Y.: A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **23**(1), 74–88 (2018)
26. Rodriguez, C.J., de Boer, S.M., Bosman, P.A., Alderliesten, T.: Bi-objective optimization of organ properties for the simulation of intracavitary brachytherapy applicator placement in cervical cancer. In: Medical Imaging 2023: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 12466, pp. 114–125. SPIE (2023)
27. Rodriguez, C.J., Thomson, S.L., Alderliesten, T., Bosman, P.A.: Temporal true and surrogate fitness landscape analysis for expensive bi-objective optimisation. arXiv preprint arXiv:2404.06557 (2024)
28. Shi, R., Liu, L., Long, T., Wu, Y., Wang, G.G.: Multidisciplinary modeling and surrogate assisted optimization for satellite constellation systems. Structural and Multidisciplinary Optimization **58**, 2173–2188 (2018)
29. Silva, R.C., Li, M., Rahman, T., Lowther, D.A.: Surrogate-based moea/d for electric motor design with scarce function evaluations. IEEE Transactions on Magnetics **53**(6), 1–4 (2017)
30. Su, S., Li, W., Li, Y., Garg, A., Gao, L., Zhou, Q.: Multi-objective design optimization of battery thermal management system for electric vehicles. Applied Thermal Engineering **196**, 117235 (2021)
31. Taran, N., Ionel, D.M., Dorrell, D.G.: Two-level surrogate-assisted differential evolution multi-objective optimization of electric machines using 3-d fea. IEEE Transactions on Magnetics **54**(11), 1–5 (2018)
32. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB platform for evolutionary multi-objective optimization. IEEE Computational Intelligence Magazine **12**(4), 73–87 (2017)

33. Tresidder, E., Zhang, Y., Forrester, A.: Acceleration of building design optimisation through the use of kriging surrogate models. Proceedings of building simulation and optimization **2012**, 1–8 (2012)
34. Wang, N., Li, C., Li, W., Chen, X., Li, Y., Qi, D.: Heat dissipation optimization for a serpentine liquid cooling battery thermal management system: An application of surrogate assisted approach. Journal of Energy Storage **40**, 102771 (2021)
35. Wang, X., Jin, Y., Schmitt, S., Olhofer, M.: An adaptive bayesian approach to surrogate-assisted evolutionary multi-objective optimization. Information Sciences **519**, 317–331 (2020)
36. Xu, W., Chong, A., Karaguzel, O.T., Lam, K.P.: Improving evolutionary algorithm performance for integer type multi-objective building system design optimization. Energy and Buildings **127**, 714–729 (2016)
37. Yu, M., Li, X., Liang, J.: A dynamic surrogate-assisted evolutionary algorithm framework for expensive structural optimization. Structural and Multidisciplinary Optimization **61**(2), 711–729 (2020)
38. Zemella, G., De March, D., Borrotti, M., Poli, I.: Optimised design of energy efficient building façades via evolutionary neural networks. Energy and Buildings **43**(12), 3297–3302 (2011)
39. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by moea/d with gaussian process model. IEEE Transactions on Evolutionary Computation **14**(3), 456–474 (2009)
40. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S., et al.: Multiobjective optimization test instances for the cec 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report **264**, 1–30 (2008)
41. Zhang, Z., Chen, H.C., Cheng, Q.S.: Surrogate-assisted quasi-newton enhanced global optimization of antennas based on a heuristic hypersphere sampling. IEEE Transactions on Antennas and Propagation **69**(5), 2993–2998 (2020)
42. Zhou, Z., Ong, Y.S., Nguyen, M.H., Lim, D.: A study on polynomial regression and gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In: 2005 IEEE congress on evolutionary computation. vol. 3, pp. 2832–2839. IEEE (2005)