Benchmarking Constrained Surrogate-based Optimization on Low Speed Airfoil Design Problems

Pramudita Satria Palar
Faculty of Mechanical and Aerospace Engineering
Institut Teknologi Bandung
Bandung, Indonesia
pramsp@ftmd.itb.ac.id

Yohanes Bimo Dwianto
Department of Aeronautics and Astronautics
University of Tokyo
Tokyo, Japan
dwianto@flab.isas.jaxa.jp

Rommel G. Regis
Department of Mathematics
Saint Joseph's University
Philadelphia, Pennsylvania, United States of America
rregis@sju.edu

Akira Oyama
Institute of Space and Astronautical Science, JAXA
Kanagawa, Japan
oyama@flab.isas.jaxa.jp

Lavi Rizki Zuhal
Faculty of Mechanical and Aerospace Engineering
Institut Teknologi Bandung
Bandung, Indonesia

ABSTRACT

In this paper, we investigated and compared the performance of various constrained surrogate-based optimization (SBO) techniques on solving low-fidelity, low-speed airfoil design problems. We aim to better understand the strengths and weakness of various constrained SBO algorithms on handling non-algebraic real-world problems. Low-fidelity problems are chosen since they allow us to perform multiple independent runs of optimization algorithms, but still in the domain of non-algebraic real-world problems. To be specific, the optimization methods that we compared are Kriging-based efficient global optimization (EGO) with the probability of feasibility (PoF), ConstrLMSRBF, COBRA, and COBYLA. Results on four airfoil design problems show that ConstrLMSRBF is the most robust method in terms of convergence and consistency of performance. On the other hand, EGO-PoF found high-quality solutions on two airfoil problems, but its robustness decreases as the dimensionality increases. We also observe that COBRA is significantly better than EGO-PoF on one high-dimensionality problem, but its general performance is not as good as that of ConstrLMSRBF. Finally, COBYLA is the worst performer, which implies that methods based on linear interpolation are not accurate for the problems considered in this paper.

CCS CONCEPTS

• **Applied computing** → *Computer-aided design*;

KEYWORDS

Surrogate-based optimization, low speed, airfoil design.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07...$15.00
https://doi.org/10.1145/3319619.3326829

ACM Reference Format:
Pramudita Satria Palar, Yohanes Bimo Dwianto, Rommel G. Regis, Akira Oyama, and Lavi Rizki Zuhal. 2019. Benchmarking Constrained Surrogate-based Optimization on Low Speed Airfoil Design Problems. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3319619.3326829

1 INTRODUCTION

Real-world engineering design problems frequently employ computationally expensive partial differential equations (PDE) solvers to evaluate objective functions. Especially in aerospace engineering, tools such as computational fluid dynamics (CFD) and finite element method (FEM) are routinely used to aid design and optimization processes. The crude use of metaheuristics such as evolutionary algorithms is not recommended due to the limited budget of function evaluations. To handle such problems, it is now common to deploy surrogate models incorporated into the framework of surrogate-based optimization (SBO). SBO, in essence, utilizes approximation models of black-box functions to assist the optimization process. Surrogate models can be coupled with metaheuristics [8, 18, 31] or used as stand-alone methods (i.e., the main driver of the optimization methods are surrogate models themselves) as in Bayesian optimization [28] and radial basis function (RBF)-based methods [9].

Most problems in engineering, including aerospace engineering, feature one or more constraints that should be satisfied while seeking the optimal solution. An example is aerodynamic shape optimization that typically involves constraints in pitching moment and lift coefficient. In topology optimization of aerospace structures, it is necessary to handle structural constraints such as stress and frequency constraint [16, 32]. It is worth noting that gradient information is not always available. Especially when function evaluations are expensive, SBO methods are probably the most viable methods for handling such problems. However, constrained SBO is still not widely studied in contrast to its unconstrained counterparts. Some successful examples of constrained SBO in real-world computationally expensive aerospace design are wing design [30] and composite structures design [17].

The efficient global optimization (EGO) [15], as one form of Bayesian optimization that uses the expected improvement (EI) metric can be equipped with the probability of feasibility (PoF) [26] to handle constraints. EGO-PoF is one of the most popular constrained SBOs due to its ability to obtain infill points that balance both exploration and exploitation in the optimization process; it is also often used as the baseline algorithm for other methods. Due to its drawback, i.e., it does not work well when one or more feasible solution is not provided in the samples, many new algorithms are developed based on EGO-PoF. For example, methods such as EGO-cons [10], SOCU [1], and the utilization of EI for constraints [13] have been proposed to improve the performance of EGO-PoF when there are no feasible solutions in the initial experimental design. EGO-PoF is also one of the methods that are utilized in surrogate model-based codes such as SurroOpt [11]. Another alternative to EGO-PoF is SEGOKPLS [4], which is an extension of SuperEGO [24] that can be used for high-dimensional constrained problems. It uses the KPLS (Kriging with Partial Least Squares) model [3] and the WB2 (locating the regional extreme) criterion [25] where the Kriging surrogate is minimized while also maximizing EI subject to Kriging surrogates of the constraints.

There also exist constrained SBO methods that rely on the non-probabilistic radial basis function (RBF) surrogates such as Constrained Local Metric Stochastic RBF (ConstrLMSRBF) [21], Constrained Optimization By RAdial basis function interpolation (COBRA) [22], and Self-Adjusting COBRA (SACOBRA) [2]. In addition, RBF surrogates have been used to assist metaheuristics for constrained optimization such as PSO (e.g., [23]).

Comparison of constrained SBOs, to the best of our knowledge, is rarely addressed in the context of non-algebraic problems (e.g., PDE-based problems). One reason is that not all PDE-based problems, e.g. those that are evaluated using CFD, feature characteristics that can be fully replicated by algebraic problems. Methods that perform as the best in algebraic problems are not necessarily the best method for handling PDE-based problems. Therefore, we think that it is also highly necessary to perform benchmarking study on PDE-based problems. In that case, to allow statistical analysis of the result, benchmarking can be performed with low-fidelity PDE-based problems.

In this paper, we have an interest in comparing the capability of four constrained SBO methods, i.e., constrained EGO with PoF, ConstrLMSRBF, COBRA, and linear interpolation-based COBYLA [20]. The surrogate models that we used in this paper follow the implementations of the original papers. In this respect, Kriging is used for EGO-PoF, RBF is used for ConstrLMSRBF and COBRA, and linear approximation is used for COBYLA. Here, constrained EGO with PoF was included since it is one of the most widely used methods when Kriging is used as the surrogate model. COBRA and ConstrLMSRBF were selected to represent SBO methods that use RBF. COBYLA was included since we want to observe how much gain in improvement can be achieved when we use RBF- and Kriging-based methods compared to a simple method (COBYLA uses linear approximation). This paper is an early attempt on the benchmarking of various optimization algorithms for solving non-algebraic constrained problems that hopefully will be useful for future algorithmic development of constrained SBO algorithms. Furthermore, the insight obtained from this research will also be useful for the development of surrogate-assisted metaheuristics methods in general, especially in the context of constrained optimization. For example, the stand-alone methods that are compared in this research can be used later as a local search mechanism to boost the performance of memetic algorithms.

2 METHODS

In this section, we briefly explain the four optimization methods that we compared in this paper. All methods model the objective and constraint functions by using surrogate models. At each iteration, all methods add a new sample by optimizing a specific updating criterion.

2.1 EGO with probability of feasibility

EGO relies on Kriging models to approximate the objective and constraints functions. Here, Kriging approximates a black-box function $f(\boldsymbol{x})$, where $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]^T$ and d is the number of decision variables, with

$$\hat{f}(\boldsymbol{x}) = \mu + \mathcal{GP}(\boldsymbol{x}), \tag{1}$$

where μ is the constant mean and $\mathcal{GP}(\boldsymbol{x})$ is a Gaussian process with mean 0 and variance $\sigma^2(\boldsymbol{x})$. For a Kriging model, one can readily obtain the prediction $\hat{f}(\boldsymbol{x})$ and mean-squared error $\hat{s}^2(\boldsymbol{x})$ at arbitrary points.

After the Kriging models for the objective and constraint functions have been constructed, one can sequentially update the sampling point by evaluating the point that maximizes the expected improvement (EI), reads as

$$EI(\boldsymbol{x}) = (f_{min} - \hat{f}(\boldsymbol{x}))\Phi\left(\frac{(f_{min} - \hat{f}(\boldsymbol{x}))}{\hat{s}(\boldsymbol{x})}\right) + \hat{s}(\boldsymbol{x})\phi\left(\frac{(f_{min} - \hat{f}(\boldsymbol{x}))}{\hat{s}(\boldsymbol{x})}\right), \tag{2}$$

where f_{min} is the best value observed and Φ and ϕ are cumulative and probability density function of a normal distribution, respectively.

One method to handle constrained optimization with EGO is by multiplying $EI(\boldsymbol{x})$ with the probability of feasibility for each constraint, i.e., $P(g_i(\boldsymbol{x}) \leq 0) = \Phi(-\hat{g}_i(\boldsymbol{x})/\hat{s}_{g_i}(\boldsymbol{x}))$ (by assuming that the constraint limit is zero), reads as

$$CEI(\boldsymbol{x}) = EI(\boldsymbol{x}) \prod_{i=1}^{K} P(g_i(\boldsymbol{x}) \leq 0), \tag{3}$$

where K is the number of constraints. In the computation of CEI, note that the EI uses the best feasible solution as the reference solution. In that case, $f_{min,feas}$ replaces f_{min} in Eq. 2. In this paper, we use EGO-PoF since it is the most widely used implementation of constrained EGO.

In this paper, we tune the Kriging model by using a multi-start gradient-based *fmincon* with 10 restarts. The constrained expected improvement is optimized by using CMA-ES [12] with five restarts.

2.2 ConstrLMSRBF

The Constrained Local Metric Stochastic RBF (ConstrLMSRBF) algorithm [21] is a constrained SBO method that uses RBF surrogates. It can be used for high-dimensional problems with many constraints and has been successfully applied to the 124-D MOPTA08 benchmark problem with 68 black-box inequality constraints from the auto industry [14].

Given n distinct sample points $x^{(1)}, \ldots, x^{(n)}$ where the objective and constraint functions have been evaluated, ConstrLMSRBF uses an interpolating RBF model of the form

$$s_n(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\boldsymbol{x} - x^{(i)}\|) + p(\boldsymbol{x}),\ x \in \mathbb{R}^d, \tag{4}$$

where $\|\cdot\|$ is the 2-norm, $\lambda_i \in \mathbb{R}$ for $i = 1, \ldots, n$ and $p(\boldsymbol{x})$ is a polynomial in d variables. Here, we use the *cubic* form ($\phi(r) = r^3$) with a linear polynomial tail $p(\boldsymbol{x})$. However, ϕ can also take the thin plate spline, multiquadric and Gaussian forms [19]. Details of how to fit this model can be found in [21].

ConstrLMSRBF uses the above type of RBF model for the objective and constraint functions, but it is possible to use other types of surrogates such as Kriging models. It treats each inequality constraint individually instead of combining them into a penalty function and builds and maintains separate surrogates, one for each constraint. As with other SBO methods, ConstrLMSRBF begins by evaluating the objective and constraint functions at the points of a space-filling design over the search space. Then, in every iteration, RBF surrogates for the objective and for each constraint function are constructed (or updated) based on all available sample points. Next, the algorithm generates a large number of random candidate points from a Gaussian distribution centered at the current best point. The algorithm then collects the *valid candidate points* Ω_n^{valid}, which are the candidate points with the minimum number of predicted constraint violations according to the RBF surrogates of the constraints. The next sample point $x^{(n+1)}$ is then chosen to be the point $x \in \Omega_n^{\text{valid}}$ that minimizes

$$\mathcal{W}_n(x) = w_n^{\text{RBF}} V_n^{\text{RBF}}(x) + w_n^{\text{Dist}} V_n^{\text{Dist}}(x), \tag{5}$$

where $V_n^{\text{RBF}}(x)$ and $V_n^{\text{Dist}}(x)$ are the scores of $x \in \Omega_n^{\text{valid}}$ for the RBF and distance criteria, respectively, and w_n^{RBF} and w_n^{Dist} are the weights for these criteria. Here, the scores for each criterion are between 0 and 1 with the more desirable points having scores closer to 0. In the default implementation of ConstrLMSRBF, $w_n^{\text{RBF}} = 0.95$ and $w_n^{\text{Dist}} = 0.05$, giving more focus on local search.

2.3 COBRA

The COBRA algorithm [22] uses the same type of RBF model as ConstrLMSRBF to approximate the objective and constraint functions. Again, it is possible to use other types of surrogates with its infill strategy. It uses a two-phase approach where Phase I finds a feasible point if none are available among the initial sample points while Phase II improves on the feasible point found in Phase I. In Phase I, the next sample point is typically a minimizer of the sum of the squares of the predicted constraint violations (as predicted by the RBF surrogates) subject only to the bounds. In Phase II, the next sample point is a minimizer of the RBF surrogate of the objective subject to RBF surrogates of the inequality constraints within some small margin while also satisfying a distance requirement from previous iterates. More precisely, after n sample points $x^{(1)}, \ldots, x^{(n)}$ have been evaluated, the next sample point $x^{(n+1)}$ solves the optimization subproblem:

$$\begin{aligned} \min_{\boldsymbol{x}}\ & s_n^{(0)}(\boldsymbol{x}) \\ \text{s.t.}\ & \boldsymbol{x} \in \mathbb{R}^d,\ \ell \le \boldsymbol{x} \le u \\ & s_n^{(i)}(\boldsymbol{x}) + \epsilon_n^{(i)} \le 0,\ i = 1, 2, \ldots, m \\ & \|\boldsymbol{x} - x^{(j)}\| \ge \rho_n,\ j = 1, \ldots, n \end{aligned} \tag{6}$$

Here, $s_n^{(0)}(\boldsymbol{x})$ is the RBF surrogate of objective while $s_n^{(i)}(\boldsymbol{x})$ is the RBF surrogate of the ith constraint given the first n sample points. Moreover, $\epsilon_n^{(i)} > 0$ is the margin for the ith constraint and ρ_n is the distance requirement after n sample points. The margins are meant to facilitate the generation of feasible iterates [22].

The distance requirements are given by $\rho_n = \gamma_n L([\ell, u])$, where $L([\ell, u])$ is the side length of the hypercube $[\ell, u]$ and γ_n can take larger values to promote global search and smaller values to facilitate local search. Here, we use the *local variant* of COBRA, the default implementation, where γ_n cycles through small values $\langle 0.01, 0.001, 0.0005\rangle$ [22]. However, other settings for the distance requirements are possible such as a more global variant described in [22] and COBRA can be quite sensitive to these settings. Moreover, the optimization subproblem in (6) is solved using Matlab's gradient-based *fmincon* solver from a good starting point obtained via a global search scheme, but this can also be replaced with a multistart approach. In this paper, we used the local variant of COBRA that is focused on local search, which is the default implementation.

2.4 COBYLA

The COBYLA (Constrained Optimization BY Linear Approximation) algorithm [20] is a derivative-free trust region method that uses linear models for the objective and constraint functions. In each iteration, linear approximations to the objective and constraint functions are constructed by interpolation at the vertices of a simplex. The next sample point $x^{(n+1)}$ is typically obtained by solving the subproblem:

$$\begin{aligned} \min_{\boldsymbol{x}}\ & \widehat{f}_n(\boldsymbol{x}) \\ \text{s.t.}\ & \widehat{g}_n^{(i)}(\boldsymbol{x}) \le 0,\ i = 1, \ldots, m \\ & \|x - x^*\| \le \Delta_n, \end{aligned} \tag{7}$$

where $\widehat{f}_n(x)$ and $\widehat{g}_n^{(i)}(x),\ i = 1, \ldots, m$ are the linear models of the objective and constraints, Δ_n is the trust region radius, and x^* is the current best sample point according to the merit function

$$\Phi_n(x) = \widehat{f}_n(x) + \mu_n[\max\{\widehat{g}_n^{(i)}(x) : i = 1, \ldots, m\}]_+. \tag{8}$$

Here, μ_n is a parameter that is automatically adjusted and the subscript + denotes the positive part of the given expression. However, in some cases, the sample point is chosen to improve the geometry of the interpolation points for the linear models. In this study, we ran COBYLA through the interface provided by the OPTI toolbox [6].

3 EMPIRICAL EXPERIMENTS

3.1 Problem definition

We used a set of low-speed airfoil design problems that use the NLF(1)-0416 airfoil as the baseline design. NLF(1)-0416 is an airfoil with natural laminar flow characteristic that was developed for general aviation applications [27], thus, it is suitable for our study.

Variable	l_b	u_b
r_{le}	0.0193	0.0236
X_{up}	0.2814	0.3440
Y_{up}	0.0942	0.1152
$Y_{xx_{up}}$	-1.1967	-0.9791
X_{lo}	0.3243	0.3964
Y_{lo}	-0.0613	-0.0501
$Y_{xx_{lo}}$	0.5552	0.6785
α_{te}	-0.2458	-0.2011
α_{te}	0.1065	0.1302

Table 1: Upper and lower bounds of the PARSEC problem.

Here, a PARSEC parameterization with 9 variables [29] and a B-spline technique with 16 tunable control points [5] are used to change the baseline airfoil; these parameters are then disturbed to act as the decision variables of the optimization problem.

The PARSEC parameterization works by representing an airfoil shape with polynomials, in which the parameters are designed in a way such that they are intuitive to engineers (e.g., radius of leading edge and trailing edge angle). The upper and lower bounds of the PARSEC problems are shown in Table 1 with the trailing edge thickness and ordinate were set to zero. Details regarding the PARSEC parameterization can be found in the original paper [29].

On the other hand, B-spline parameterization works by attaching control points close to the airfoil. The location of these control points are then systematically altered to change the airfoil shape. Figure 1 shows the baseline airfoil and the control points for the B-spline parameterization. We attached 20 control points but only 16 points are allowed to vary in the z_2 direction. For both B-spline and PARSEC-based problems, we use XFOIL [7] to evaluate the objective and the constraint functions. XFOIL is a low-fidelity solver, yet accurate for low-speed problems, that couples a boundary layer solver and a panel method. One XFOIL simulation takes less than one second in a personal computer; hence, performing independent runs of optimization with multiple methods is doable.

The objective function for all problems is to maximize the lift-to-drag ratio (L/D) as the performance measure of aerodynamic design subjects to constraints in lift coefficient (C_l), absolute pitching moment coefficient ($|C_m|$), and laminar to turbulent flow transition point. We vary the problem by enforcing two and three constraints, where the two-constraint problems use C_l and C_m as the constraints. The maximization problem can be transformed into a minimization problem by simply multiplying the L/D with negative one. The pitching moment constraint for the B-spline problem is set to a different value so as to create problems with feasible domain while still minimizing the number of failed simulations. The transition point is set as one constraint in order to maintain laminar flow over the upper surface, which is beneficial for airfoil design.

The four problems are summarized in the following:

(1) NLF(1)-0416, PARSEC parameterization ($n_{var} = 10$). Constraints: $C_l > 0.5$, $|C_m| < 0.10$.
(2) NLF(1)-0416, PARSEC parameterization ($n_{var} = 10$). Constraints: $C_l > 0.5$, $|C_m| < 0.10$, transition point > 0.4
(3) NLF(1)-0416, B-Spline parameterization ($n_{var} = 17$). Constraints: $C_l > 0.5$, $|C_m| < 0.11$.
(4) NLF(1)-0416, B-spline parameterization ($n_{var} = 17$). Constraints: $C_l > 0.5$, $|C_m| < 0.11$, transition point > 0.4

Notice that the angle of attack α is also set as one design variable, which is set to vary from 0 to 4 degrees; thus, the number of actual design variables is the number of airfoil parameters plus one. The number of initial and new samples for the PARSEC problem are 30 and 100, respectively. For the B-spline-problem, due to its higher dimensionality, the number of initial and new samples are set to 50 and 100, respectively. We set these numbers of initial new samples by primarily considering the expensive construction time of Kriging, thus, we cannot afford too many iterations if the number of initial samples is already high. The training time and acquisition function optimization of EGO are significantly more expensive than the others and this limits the number of EGO runs that we can perform. To that end, also for a fair comparison, we evaluate the performance of the optimization algorithms by using 15 independent runs for all problems.

It is worth noting that it is very difficult to design a test problem without failed simulations (i.e., simulations that do not converge). Here, failed simulations are not those that violate the constraint but simulations that yield no value of the objective and constraint functions. If there are too many failed solutions, it becomes difficult to assess the difference in the performance of various optimization methods. Therefore, we minimized the number of failed simulations by carefully choosing the upper and lower bounds of the design variables. The number of failed simulation for each problem, according to a random sampling with 2000 samples, is about 2%. Although this number is reasonably low, we imputed the objective and constraint values for failed simulations with $\max(Y) + \mathrm{std}(Y)$, where (Y) is the vector of responses from 2000 random simulations.

To assess the difficulty of the problem in terms of feasibility, We estimated the percentage of feasible region by 2000 random sampling and we found that the feasible region for problem (1), (2), (3), and (4) are 12.3%, 4.25%, 50.2%, 4.4%, respectively. In addition, we estimated the percentage of the feasible region with each and all constraints considered, as shown in Table 2. From the table, we can see that the lift constraint is relatively easy to fulfill, followed by the pitching moment constraint, and the transition point constraint; the latter is obviously the hardest for both B-Spline and PARSEC problems. The lift constraint is relatively easy to fulfill because it is sensitive to the change of angle of attack. The moment constraint of the B-spline problem was relaxed, which is the reason why the percentage of feasible of region is higher than that of the PARSEC problem. The transition point constraint is not so sensitive to the change in design variables considered in this paper, which is the reason why it creates a small feasible space.

The PARSEC problem experiments were executed with a personal computer with the specifications of Intel(R) Core(TM) i5-6200U CPU @2.30 GHz 2.40 GHz and 4 GB RAM. On the other hand, the B-spline problems were performed in a personal computer with Intel Xeon E5 2620 v4, with 8 cores, 16 threads, 2.10 Ghz (base clock), and RAM 16 Gb 2400 MHz.

Problem	C_l	$\lvert C_m \rvert$	x_{trs}	C_l and $\lvert C_m \rvert$	All
PARSEC	93.35%	16.6%	24.5%	12.3%	4.25%
B-Spline	95.3%	52.4%	10.9%	50.2%	4.4%

Table 2: Percentages of feasible region for all problems due to the single and combined constraints.

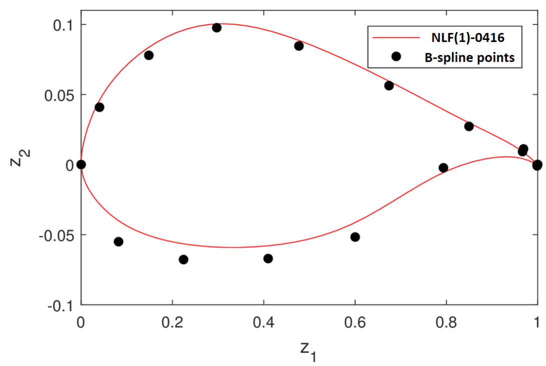

Figure 1: The NLF(1)-0416 airfoil and locations of B-spline points used to alter the shape.

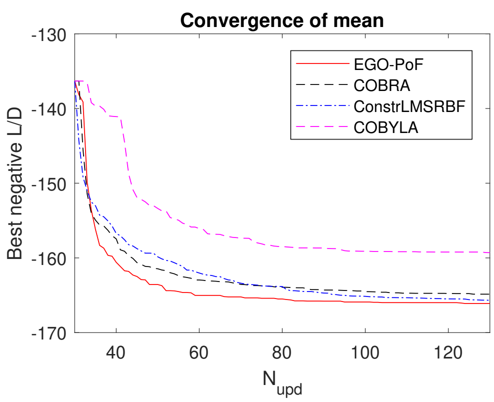

Figure 2: Mean convergence plots of best feasible objective functions for the PARSEC problem with two constraints.

3.2 Experimental results

We show the convergence of the mean of the best negative L/D during the course of optimization. In addition, we show the boxplot of solutions at the final iteration to depict the distribution of final best feasible solutions. We judge the performance of the algorithms primarily from these two aspects, thus, we analyze both the mean convergence and the distribution of final best feasible solutions. This is because there might be cases when an algorithm yields a fast convergence but the distribution of final solutions is not so satisfactory. Notice that we defined a feasible solution as a solution that strictly fulfills the constraints without any threshold. The results for the PARSEC problems are shown in Figs. 2, 3, 4, and 5, while the results for the B-spline problems are shown in Figs. 6, 7, 8, and 9.

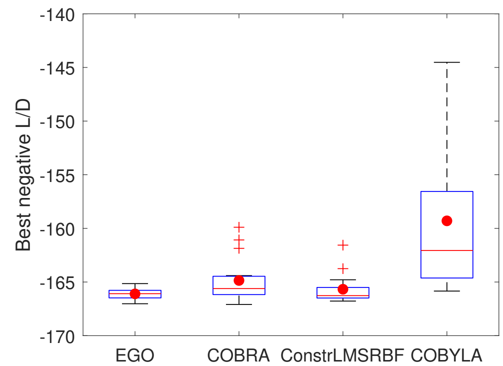

Figure 3: Boxplots of final best feasible objective functions for the PARSEC problem with two constraints.

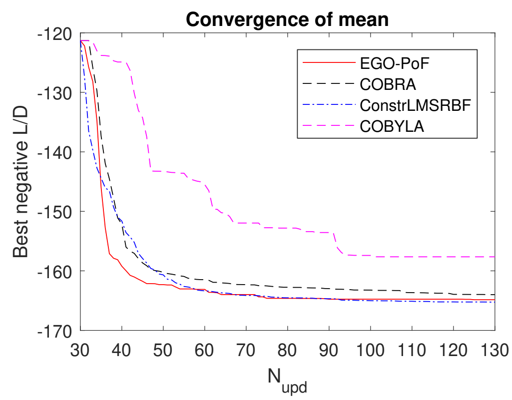

Figure 4: Mean convergence plots of best feasible objective functions for the PARSEC problem with three constraints.

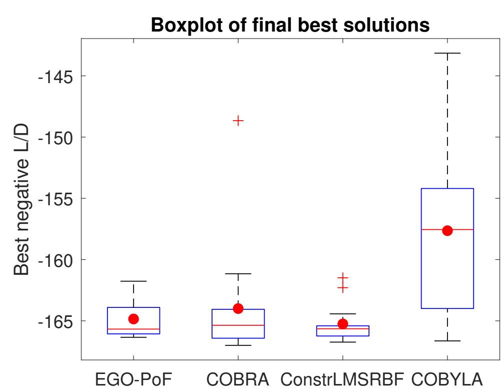

Figure 5: Boxplots of final best feasible objective functions for the PARSEC problem with three constraints.

First, we notice that COBYLA is the worst performer on all problems. This clearly implies that linear approximations are not sufficient to solve the constrained aerodynamic optimization problems considered in this paper. Instead, RBF (as used by COBRA and ConstrLMSRBF) and Kriging (as used by EGO-PoF), which

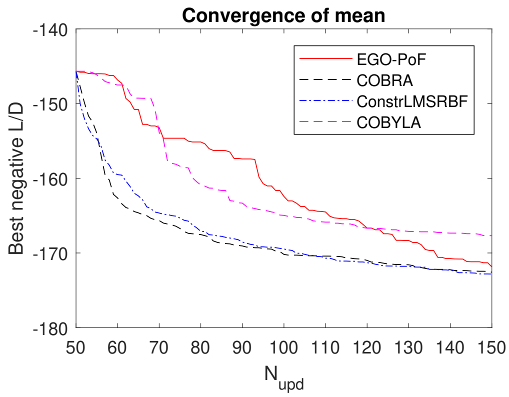

Figure 6: Mean convergence plots of best feasible objective functions for the B-spline problem with two constraints.

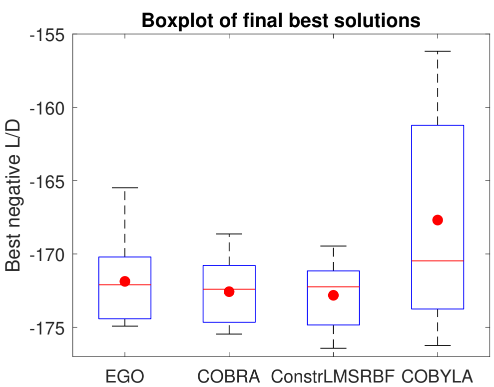

Figure 7: Boxplots of final best feasible objective functions for the B-Spline problem with two constraints.

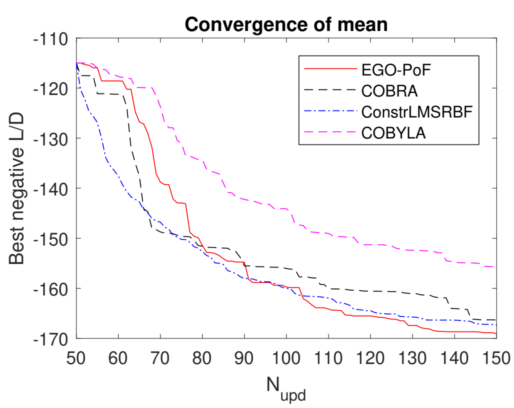

Figure 8: Mean convergence plots of best feasible objective functions for the B-spline problem with three constraints.

are capable of capturing nonlinear functions, are more suitable for handling this kind of problem; we can see that they converged faster

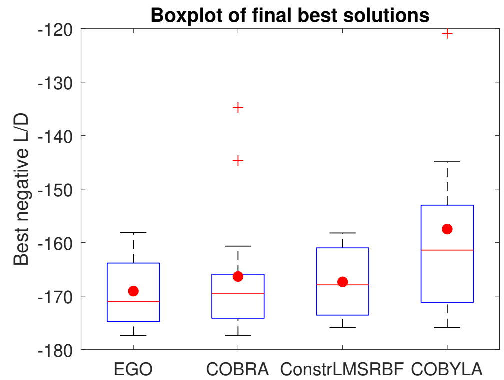

Figure 9: Boxplots of final best feasible objective functions for the B-Spline problem with three constraints.

and yielded significantly better best solutions at the end of the optimization (one interesting case is on the B-spline problem with two constraints as we discuss below).

The results show that, in general, ConstrLMSRBF outperformed COBRA on all problems; here, we can see that the former yielded a better convergence property (although only slightly) and better distribution of final best feasible solutions than the latter. On the PARSEC problem with two constraints, COBRA briefly outperformed ConstrLMSRBF at early iterations but was soon outpaced by ConstrLMSRBF. However, despite its volatile performance, the observed best minimum solution of the two PARSEC problems was found by COBRA. EGO-PoF yielded the best convergence on the two PARSEC problems, in fact, it even outperformed ConstrLMSRBF from the viewpoint of both convergence and best final solutions. Nevertheless, ConstrLMSRBF was still competitive with EGO-PoF on these two problems. Such a fact indicates that EGO-PoF identified the feasible regions quicker than the other methods, at least for the PARSEC problems. Despite the fast convergence of EGO-PoF, ConstrLMSRBF yielded better distribution of final solutions on the PARSEC problem with three constraints. By remembering that the feasible region of the PARSEC problems with three constraints is very low, we infer that EGO-PoF (at least with the current setting) is not so good at the exploitation phase.

We observe an interesting phenomenon in the results of the B-spline problem with two and three constraints (see Figs. 6, 7, 8 and 9). For the two-constraint problem, ConstrLMSRBF started to outperform COBRA in terms of the mean convergence at about the 100th iteration, although the difference is actually slight; we can also see that the former is better from the viewpoint of final distributions. Such an observation is in line with the results of the PARSEC problems which demonstrate the robustness of ConstrLMSRBF. What interesting here is that the mean convergence of EGO-PoF is even worse than that of COBYLA before the 120th iteration on the two-constraint PARSEC problem. This means that for the B-spline problem with two constraints, EGO-PoF spent too much time in exploring the design space. EGO-PoF finally exceeded COBYLA starts from about iteration 120 and its distribution of final solutions is also much better than COBYLA. However, this indicates that one should be careful when using EGO-PoF. For example, in a situation where one could only afford 50 new solutions, EGO-PoF could find worse solutions compared to when we use COBRA and ConstrLMSRBF.

Surprisingly, EGO-PoF performed well on the B-spline problem with three constraints. EGO-PoF, COBRA, and ConstrLMSRBF are indeed competitive with each other on this problem. The performance of COBRA is not so robust as indicated by poor values of two outliers. We can also see that EGO-PoF was slow at early iterations on the B-Spline problem with three constraints; however, if we pay a good attention to the mean convergence of the two PARSEC problems, it can also be seen that EGO-PoF needed a time to warm-up itself. However, the fact that EGO-PoF does not perform well on the B-spline problem with two constraints signifies that EGO-PoF is not a really robust method. It is also perplexing that EGO-PoF performed better on a problem that was supposed to be more difficult (i.e., the problem with three constraints). The best guess that we have now is that the performance of EGO-PoF highly depends on the initial sampling points, which makes sense if we take into account the fact that EGO-PoF primarily exploits locations near the current feasible solutions.

3.3 Discussion on the run time

We plot the average actual computation time of the algorithms on the PARSEC problem with two constraints and the B-spline problem with three constraints in Figs. 10 and 11, respectively, which represent the case with the shortest and the fastest run time. In terms of the run time, we can see that COBRA, ConstrLMSRBF, and COBYLA are methods that run very quick even on a personal computer. This means that these three algorithms are also suitable for solving problems with a moderate budget of function evaluations (say, 500-1000 function evaluations). For all algorithms but EGO, the minor difference in the run time is attributable to the algorithm implementation rather than the surrogate model building. It is worth noting that the short run time of these three algorithms stems from the fast model building. ConstrLMSRBF and COBRA will also consume a significant amount of time if expensive surrogates are used. Nevertheless, one advantage of ConstrLMSRBF and COBRA compared to EGO is that they can use non-probabilistic surrogates in the loop since only the prediction structure is needed. The calculation time of EGO-PoF is significantly longer than the other algorithms, which is primarily due to the construction time of the Kriging model. It is worth noting that, just like other algorithms, EGO-PoF needs to construct $1+n_{con}$ Kriging models for one function evaluation, where n_{con} is the number of constraints. The problem is that the calculation time of EGO significantly increases as the number of constraint increases, which implies that EGO-PoF is not suitable for problems with many constraints.

3.4 Discussion on the percentage of feasible regions

Figures 12, 13, 14, and 15 show the average percentage of feasible solutions for the PARSEC problem with two constraints, PARSEC problem with three constraints, B-spline problem with two constraints, and B-spline problem with three constraints, respectively. Basically, we observe that the percentage of feasible solutions typically increases as we add more updating samples, although the percentage got stalled in some problems (see Fig. 13 for example).

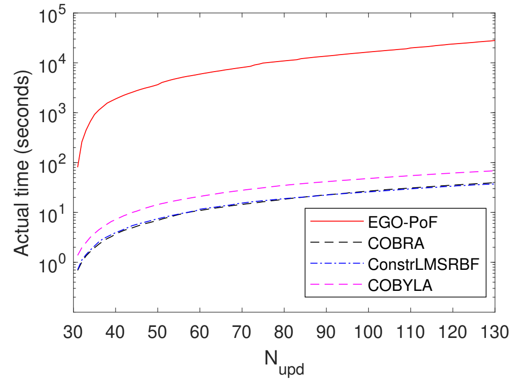

Figure 10: The actual computation time (in seconds) of all algorithms averaged from 15 independent runs for the PARSEC problem with two constraints.

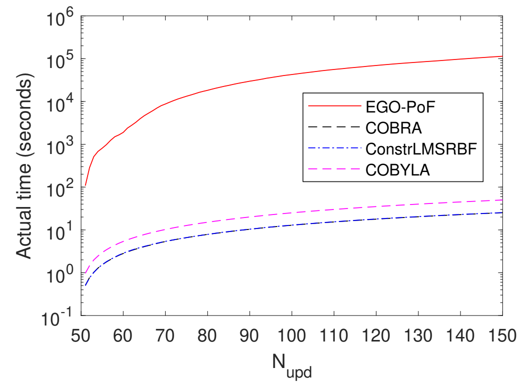

Figure 11: The actual computation time (in seconds) of all algorithms averaged from 15 independent runs for the B-spline problem with three constraints.

The results show that the two RBF-based methods (i.e., COBRA and ConstrLMSRBF) tend to generate a higher percentage of feasible solutions as the optimization progresses when compared to the other two methods. On the other hand, EGO-PoF and COBYLA tend to generate a similar percentage of feasible solutions. The exception is on the PARSEC problem with three constraints, where the percentage of feasible solutions of EGO-PoF is even lower than 10%. Despite this, EGO-PoF performed better than COBYLA, which indicates that the former is more efficient on finding good feasible solutions than the latter. It is also interesting to note that EGO-PoF performed well on both PARSEC problems although it generated a higher percentage of infeasible solutions compared to ConstrLMSRBF and COBRA; however, this is not the case in the B-spline problem with two constraints. This might mean one of two things, that is, either EGO-PoF primarily explored infeasible regions after it found a near global optimum location or it over-explored infeasible regions (the latter is the case for the B-spline problem with two constraints). Compared to EGO-PoF, it seems like COBRA and ConstrLMSRBF maintain a more balanced search between infeasible and feasible parts of the design space.

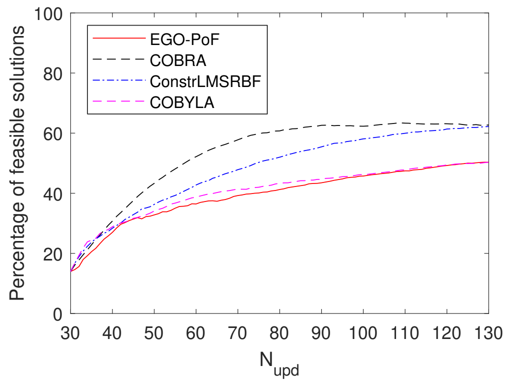

Figure 12: Percentage of feasibility averaged from 15 independent runs for the PARSEC problem with two constraints.

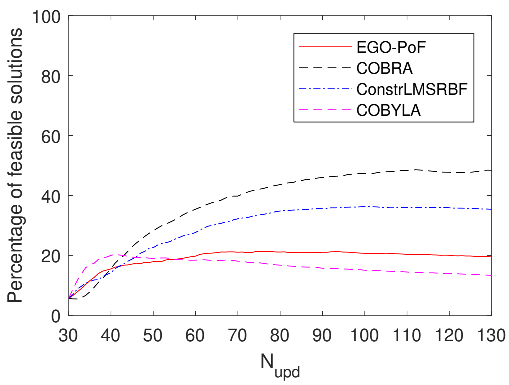

Figure 13: Percentage of feasibility averaged from 15 independent runs for the PARSEC problem with three constraints.

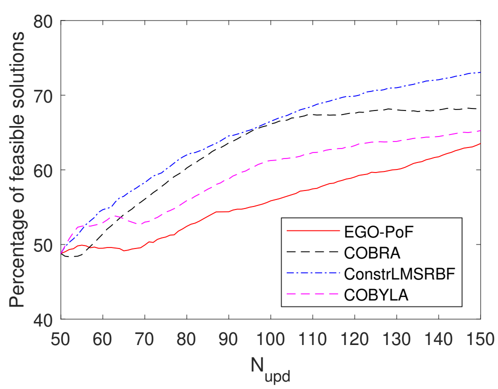

Figure 14: Percentage of feasibility averaged from 15 independent runs for the B-spline problem with two constraints.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we compared the performance of four constrained SBO methods (i.e., EGO with the probability of feasibility, COBRA, ConstrLMSRBF, COBYLA) on four aerospace engineering

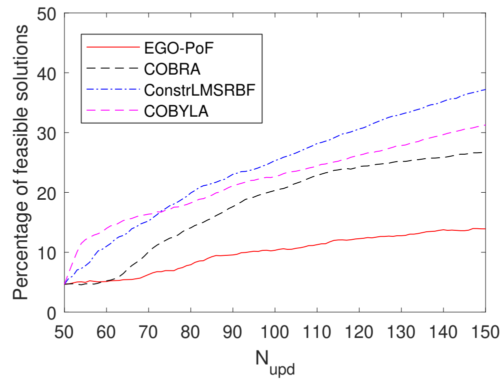

Figure 15: Percentage of feasibility averaged from 15 independent runs for the B-spline problem with three constraints.

problems, that is, low-speed airfoil design optimization cases with a varying number of constraints and dimensionality. The primary objective of this paper is to investigate the strengths and weaknesses of various constrained SBO methods on solving PDE-based problems (represented by low-speed airfoil cases). It is hoped that the insight obtained from this paper will be useful for future algorithmic development to handle real-world constrained optimization problems, especially in the aerospace domain.

The results show that ConstrLMSRBF is the most consistent and well-performing method, in the sense that it is relatively robust to the various number of constraints and dimensionality. The performance of COBRA, although still much better than COBYLA, is still not as robust as ConstrLMSRBF. However, in this paper, it is worth noting that we only used the local variant of COBRA; comparison with adaptive version of COBRA is one subject of future works. EGO-PoF performed very well on the PARSEC problems but it performed poorly when it was used to solve the high-dimensional problem with two constraints. The poor performance of COBYLA relative to the others indicates that linear approximations are not adequate enough to solve the current low-fidelity low-speed airfoil design problems.

For future works, we are planning to completely separate the effect of the baseline algorithm (including constraint handling techniques) and the surrogate models. In this regard, all algorithms considered so far, with the exception of COBYLA, should be compared by using identical surrogate models (e.g., Gaussian RBF). It is also worth noting that the insight obtained from the experiment this paper applies in the context of low-speed airfoil design. Therefore, we think that experiments with other types of problems and different CFD solvers are necessary to complement the results of this paper.

ACKNOWLEDGEMENT

Pramudita Satria Palar and Lavi Rizki Zuhal were funded in part through the Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT) administered by Direktorat Riset dan Pengabdian Masyarakat - Direktorat Jenderal Penguatan Riset dan Pengembangan - Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia.

REFERENCES

[1] Samineh Bagheri, Wolfgang Konen, Richard Allmendinger, Jürgen Branke, Kalyanmoy Deb, Jonathan Fieldsend, Domenico Quagliarella, and Karthik Sindhya. 2017. Constraint handling in efficient global optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 673–680.

[2] Samineh Bagheri, Wolfgang Konen, Michael Emmerich, and Thomas Bäck. 2017. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing* 61 (2017), 377–393.

[3] Mohamed Amine Bouhlel, Nathalie Bartoli, Abdelkader Otsmane, and Joseph Morlier. 2016. Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. *Structural and Multidisciplinary Optimization* 53, 5 (May 2016), 935–952.

[4] Mohamed Amine Bouhlel, Nathalie Bartoli, Rommel G. Regis, Abdelkader Otsmane, and Joseph Morlier. 2018. Efficient global optimization for high-dimensional constrained problems by using the Kriging models combined with the partial least squares method. *Engineering Optimization* 50, 12 (2018), 2038–2053.

[5] Edwin Catmull and James Clark. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-aided design* 10, 6 (1978), 350–355.

[6] Jonathan Currie and David I. Wilson. 2012. OPTI: Lowering the Barrier between Open Source Optimizers and the Industrial MATLAB User. In *Foundations of Computer-Aided Process Operations*, Nick Sahinidis and Jose Pinto (Eds.). FOCAPO, Savannah, Georgia.

[7] Mark Drela. 1989. XFOIL: An analysis and design system for low Reynolds number airfoils. In *Low Reynolds number aerodynamics*. Springer, 1–12.

[8] Wenyin Gong, Aimin Zhou, and Zhihua Cai. 2015. A multioperator search strategy based on cheap surrogate models for evolutionary optimization. *IEEE Transactions on evolutionary computation* 19, 5 (2015), 746–758.

[9] H-M Gutmann. 2001. A radial basis function method for global optimization. *Journal of global optimization* 19, 3 (2001), 201–227.

[10] Ahsanul Habib, Hemant Kumar Singh, and Tapabrata Ray. 2016. A study on the effectiveness of constraint handling schemes within Efficient Global Optimization framework. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1–8.

[11] Zhong-Hua Han. 2016. SurroOpt: a generic surrogate-based optimization code for aerodynamic and multidisciplinary design. In *Proceedings of the 30th Congress of the International Council of the Aeronautical Sciences*.

[12] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation* 11, 1 (2003), 1–18.

[13] Ruwang Jiao, Sanyou Zeng, Changhe Li, Yuhong Jiang, and Yaochu Jin. 2019. A complete expected improvement criterion for Gaussian process assisted highly constrained expensive optimization. *Information Sciences* 471 (2019), 80–96.

[14] Donald R. Jones. 2008. Large-scale multi-disciplinary mass optimization in the auto industry. Presented at the MOPTA (Modeling and Optimization: Theory and Applications) 2008 Conference. Ontario, Canada.

[15] Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.

[16] Mark K Leader, Ting Wei Chin, and Graeme Kennedy. 2018. High Resolution Topology Optimization of Aerospace Structures with Stress and Frequency Constraints. In *2018 Multidisciplinary Analysis and Optimization Conference*. 4056.

[17] Ang Li. [n. d.]. *Optimization of Composite Structures for Crashworthiness*. Master's thesis. Delft University of Technology.

[18] Pramudita Satria Palar, Takeshi Tsuchiya, and Geoffrey Thomas Parks. 2016. A comparative study of local search within a surrogate-assisted multi-objective memetic algorithm framework for expensive problems. *Applied Soft Computing* 43 (2016), 1–19.

[19] Michael J. D. Powell. 1992. The theory of radial basis function approximation in 1990. In *Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions*, W. Light (Ed.). Oxford University Press, Oxford, U.K., 105–210.

[20] Michael J. D. Powell. 1994. A direct search optimization methods that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*, S. Gomez and J.P. Hennart (Eds.). Kluwer, Dordrecht, 51–67.

[21] Rommel G Regis. 2011. Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Computers & Operations Research* 38, 5 (2011), 837–853.

[22] Rommel G Regis. 2014. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization* 46, 2 (2014), 218–243.

[23] Rommel G. Regis. 2018. Surrogate-Assisted Particle Swarm with Local Search for Expensive Constrained Optimization. In *Bioinspired Optimization Methods and Their Applications*, Peter Korošec, Nouredine Melab, and El-Ghazali Talbi (Eds.). Springer International Publishing, Cham, 246–257.

[24] Michael J. Sasena. 2002. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. Ph.D. Dissertation. University of Michigan, Michigan, USA.

[25] Michael J Sasena, Panos Papalambros, and Pierre Goovaerts. 2002. Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering optimization* 34, 3 (2002), 263–278.

[26] Matthias Schonlau. 1997. Computer experiments and global optimization. (1997).

[27] Michael S Selig, Mark D Maughmer, and Dan M Somers. 1995. Natural-laminar-flow airfoil for general-aviation applications. *Journal of aircraft* 32, 4 (1995), 710–715.

[28] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2016. Taking the human out of the loop: A review of bayesian optimization. *Proc. IEEE* 104, 1 (2016), 148–175.

[29] Helmut Sobieczky. 1999. Parametric airfoils and wings. In *Recent development of aerodynamic design methodologies*. Springer, 71–87.

[30] Ke-Shi Zhang, Zhong-Hua Han, Zhong-Jian Gao, and Yuan Wang. 2019. Constraint aggregation for large number of constraints in wing surrogate-based optimization. *Structural and Multidisciplinary Optimization* 59, 2 (2019), 421–438.

[31] Zongzhao Zhou, Yew Soon Ong, Meng Hiot Lim, and Bu Sung Lee. 2007. Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Computing* 11, 10 (2007), 957–971.

[32] Ji-Hong Zhu, Wei-Hong Zhang, and Liang Xia. 2016. Topology optimization in aircraft and aerospace structures design. *Archives of Computational Methods in Engineering* 23, 4 (2016), 595–622.