HAL
open science

An efficient application of Bayesian optimization to an industrial MDO framework for aircraft design

Rémy Priem, Hugo Gagnon, Ian Chittick, Stephane Dufresne, Youssef Diouane, Nathalie Bartoli

To cite this version:

Rémy Priem, Hugo Gagnon, Ian Chittick, Stephane Dufresne, Youssef Diouane, et al.. An efficient application of Bayesian optimization to an industrial MDO framework for aircraft design. AIAA AVIATION 2020 FORUM, Jun 2020, Inconnu, United States. 10.2514/6.2020-3152 . hal-02861588

HAL Id: hal-02861588

https://hal.science/hal-02861588

Submitted on 9 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An efficient application of Bayesian optimization to an industrial MDO framework for aircraft design

Rémy Priem*
ONERA, DTIS, Université de Toulouse, Toulouse, France
ISAE-SUPAERO, Université de Toulouse, Toulouse, 31055 Cedex 4, France

Hugo Gagnon,† Ian Chittick,‡ and Stéphane Dufresne§
Bombardier Aviation, Dorval, Québec, H9P 1A2, Canada

Youssef Diouane¶
ISAE-SUPAERO, Université de Toulouse, Toulouse, 31055 Cedex 4, France

Nathalie Bartoli‖
ONERA, DTIS, Université de Toulouse, Toulouse, France

The *multi-level*, *multi-disciplinary* and *multi-fidelity* optimization framework developed at Bombardier Aviation has shown great results to explore efficient and competitive aircraft configurations. This optimization framework has been developed within the Isight software, the latter offers a set of ready-to-use optimizers. Unfortunately, the computational effort required by the Isight optimizers can be prohibitive with respect to the requirements of an industrial context. In this paper, a *constrained Bayesian optimization* optimizer, namely the *super efficient global optimization with mixture of experts*, is used to reduce the optimization computational effort. The obtained results showed significant improvements compared to two of the popular Isight optimizers. The capabilities of the tested *constrained Bayesian optimization* solver are demonstrated on *Bombardier research aircraft configuration* study cases.

Preprint

I. Nomenclature

d	=	the number of design variables
m	=	the number of inequality constraints
Ω	=	the domain of the design variables, i.e., $\Omega \subset \mathbb{R}^d$
f	=	the objective function, i.e., $f : \mathbb{R}^d \mapsto \mathbb{R}$
$\boldsymbol{g}$	=	the constraints function, i.e., $\boldsymbol{g} : \mathbb{R}^d \mapsto \mathbb{R}^m$
$\mu_s^{(l)}$	=	the prediction of the Gaussian process of a given function $s : \mathbb{R}^d \mapsto \mathbb{R}$ build with l samples
$\sigma_s^{(l)}$	=	the uncertainty of the Gaussian process of a given function $s : \mathbb{R}^d \mapsto \mathbb{R}$ build with l samples
$\alpha_f^{(l)}$	=	the acquisition function of the objective function $f : \mathbb{R}^d \mapsto \mathbb{R}$
$\alpha_{\boldsymbol{g}}^{(l)}$	=	the feasibility criterion of the constraint functions $\boldsymbol{g} : \mathbb{R}^d \mapsto \mathbb{R}^m$
$\Omega_{\boldsymbol{g}}^{(l)}$	=	the approximated feasible domain defined by the feasibility criterion $\alpha_{\boldsymbol{g}}^{(l)} : \mathbb{R}^d \mapsto \mathbb{R}^m$
$C_{L_{max}}$	=	the low speed maximum lift coefficient

*Ph.D. Candidate, Information Processing and Systems Department & Complexes Systems Engineering Department, remy.priem@onera.fr, Student AIAA Member

†Engineering Professional, Advanced Design, hugo.gagnon@aero.bombardier.com.

‡Engineering Specialist, Advanced Aerodynamics, ian.chittick@aero.bombardier.com.

§Senior Engineering Specialist, Specialized Aircraft, stephane.dufresne@aero.bombardier.com.

¶Associate Professor, Complexes Systems Engineering Department, youssef.diouane@isae-supaero.fr.

‖Senior researcher, Information Processing and Systems Department, nathalie.bartoli@onera.fr, AIAA Member.

II. Introduction

BOMBARDIER aviation *multi-disciplinary optimization* (MDO) framework has been continuously evolving over the past decade to ensure that future development programs result in highly efficient and competitive aircraft [1]. In order to achieve this vision, Bombardier has developed a multi-level, multi-fidelity MDO framework to mature an aircraft configuration from early design stage to detailed design [2].

The first element of this framework is called *Conceptual MDO* (CMDO). The main objective of the CMDO framework is to enable the design team to explore the design space assuming a set of realistic constraints established from previous aircraft development programs and marketing requirements. The main deliverable of this process is an aircraft configuration, which includes the initial sizing of the major structural components and systems. This conceptual aircraft configuration then generates a set of design requirements that will be validated using high fidelity frameworks, namely the *Preliminary MDO* (PMDO) and the *Detailed MDO* (DMDO). The second element corresponds to the PMDO framework analyzing the aerodynamic and structural characteristics of the wing. Its main objectives are to refine the wing sizing and to validate the wing related conceptual requirements, such as drag polar and wing weight. The two deliverables of this process are (1) a detailed wing geometry, external and internal, and (2) an initial definition of the wing structure.

The context of this research is about the design process of a new aircraft configuration in an industrial setting. The main inputs to initiate the design of a new aircraft platform, as well as inputs to the MDO framework, are a set of requirements from the marketing team (e.g. cruise mach number, design range, field performance and critical airport operations). For each of these sets of requirements, or constraints to the MDO framework, there needs to be a convergence between the CMDO and PMDO framework to ensure a robust initial design point. Since there could be multiple sets of requirements to be analyzed, the design team becomes rapidly limited by the time and number of CPU available.

The CMDO framework is built on a large set of low fidelity modules representing the aircraft disciplines. All these modules include their own set of design variables, which increases the number of dimensions of the optimization problem. Consequently, it makes the thorough exploration of the design space expensive in terms of function calls and CPU time, in the order of days. Similarly, the PMDO framework also includes a large number of design variables required to define the wing geometry and the aero-structural models are time consuming to converge, in the order of weeks. The motivation of this research is then to apply Bayesian Optimization (BO) methods [3] to reduce the number of functions calls and accelerate the convergence time for both MDO frameworks. To do so, the SEGOMOE python tool-box is investigate against historical optimizers used at Bombardier aviation. The main objective of this paper is thus to demonstrate that all the optimizers of the tool-box has good convergence properties and performs well on an industrial test case.

This paper is organized into four sections: (1) detailed description of the models used in the CMDO and PMDO frameworks, (2) introduction to the Bombardier Research Aircraft Configuration (BRAC) case study, (3) definition of the optimizer selected for the analysis of the case study and finally (4) the analysis of the results with the emphasis on the convergence in terms of CPU time and number of evaluations.

Preprint

III. An industrial multi-level, multi-fidelity and multi-disciplinary optimization process

The MDO framework developed at Bombardier Aviation consists of three levels, corresponding to the three traditional stages of aircraft design: conceptual, preliminary, and detailed. Only a brief description of the first two levels is given here. Refer to [2] for more information.

A. Conceptual multi-disciplinary optimization framework

Conceptual MDO (CMDO) is the first level of the multi-level MDO framework. It accounts for all major aircraft-level disciplines, such as aerodynamics, structures, systems, weight and balance, performance, stability and control, and economics. Due to the high number of disciplines, the level of fidelity is low and varies from empirical to simple physics-based methods. For example, in the aerodynamics module, an in-house method is used to predict the low-speed maximum lift coefficient $C_{L_{max}}$ while a vortex-lattice method coupled with a 2D computational fluid dynamics code is used to compute high-speed trim drag. Similarly, other disciplines can combine mixed-fidelity analyses depending on the desired accuracy and turnaround times. Example usages of CMDO are design space exploration, optimization of marketing requirements and objectives, validation of business cases, assessment of technology insertion, down selection of promising configurations, and definition of performance targets [2].

Typical design variables of CMDO include the wing area and planform as well as the engine scaling factor in the

case of a fixed engine architecture. It is also possible to vary the thickness-to-chord ratios of the wing and chordwise positions of the spars at various spanwise locations. Typical constraints include performance parameters such as the *balanced field length* (BFL), *approach speed* (V_{ref}), *initial cruise altitude* (ICA), range for different missions, and the ability to take-off and land at critical airports. Systems and geometric considerations such as landing gear integration and wingtip chord can also be included. Typical objectives include *maximum take-off weight* (MTOW), cost, and climate impact, or any combination thereof. For this study only MTOW is used since it combines the effects of cost and climate impact through fuel burn.

The CMDO framework is implemented as a workflow in Isight [4]. Isight is a process-integration, design-optimization software developed and distributed by Dassault Systemes. It comes with its own set of optimizers.

The output of CMDO is an input to *Preliminary MDO* (PMDO), discussed next.

B. Preliminary multi-disciplinary optimization framework

The *preliminary MDO* (PMDO) is the second level of the multi-level MDO framework. Compared to CMDO, the scope of the design space is narrowed, while the fidelity of the underlying analysis tools is increased. PMDO focuses primarily on the coupling and trade-off between aerodynamics and structures. The objective is to obtain wing *outer mold lines* (OML) that are aerodynamically efficient in both high-speed and low-speed flight, as well as to provide adequate performance at critical off-design conditions. At the same time, the OML must be structurally viable, yield low weight, and satisfy space allocation requirements.

Piperni et al. [2] previously described the PMDO process in detail. Although many of the components remain the same, the framework has continued to evolve, and a complete description falls outside the scope of this paper. In brief, the aircraft surfaces are updated in a parametric CATIA V5 model. Bombardier's in-house structured mesh generator, MBGRID [5], and Navier-Stokes flow solver, FANSC [6], are used to compute the flow solutions at several high-speed design and static load case conditions. The external loads are transferred to the wing structural mesh generated by AWSOM [7], Bombardier's in-house wing sizing tool, which minimizes the primary wing-box weight while respecting various margin of safety constraints. The low-speed characteristics are assessed using the semi-empirical Valarezo criteria [8] in conjunction with VSAERO, a 3D panel code [9]. Finally, the high-speed drag and structural weight are related through a fractional change equation, as described in [10].

The architecture of the PMDO framework has changed significantly with the maturation of BOOST, an adjoint-based optimization framework used at Bombardier [11, 12]. The adjoint method is an efficient tool for computing the gradients needed for aerodynamic shape optimization, which involves long running simulation times and large numbers of design variables. In order to integrate the aerodynamic adjoint in a multidisciplinary environment, a hybrid-adjoint MDO framework was developed [13], as depicted in Figure 1. In the first stage, an aero-structural optimization is performed in Isight by varying the wing plan-form, wing twist, and maximum thickness distribution (by scaling the wing profiles). This is followed by an aero-shape optimization in BOOST, where the drag is minimized by varying the airfoil shapes at fixed thickness to maximum chord ratio and wing span-load. This process is repeated in a sequential manner until convergence is achieved. This paper will examine the top-level optimizer used to drive the aero-structural optimization, highlighted in bold in Figure 1, which is the current bottleneck in the PMDO process.

C. Bombardier research aircraft configuration

The test case considered for this work is representative of a small business aircraft, and is referred to as the *Bombardier research aircraft configuration* (BRAC). It is based on an early design of the Challenger 300 platform, and was developed to be shared with academic partners and for publication purposes. An image of the baseline BRAC geometry is shown in Figure 2.

IV. Bombardier research aircraft configuration optimization case

In this section, the two level optimization process is introduced. First, we present the optimization of the BRAC model into the CMDO framework. Then, the previously obtained result is used in the PMDO of the BRAC model.

A. BRAC CMDO problem

The BRAC CMDO problem is based on a modified version of the initial Challenger 300 marketing requirements and objectives. It is thus representative of a typical industrial problem, yet here the design space is purposefully relaxed to test the global optimization capabilities of each optimizer. For confidentiality reasons the exact bounds on the design

Preprint

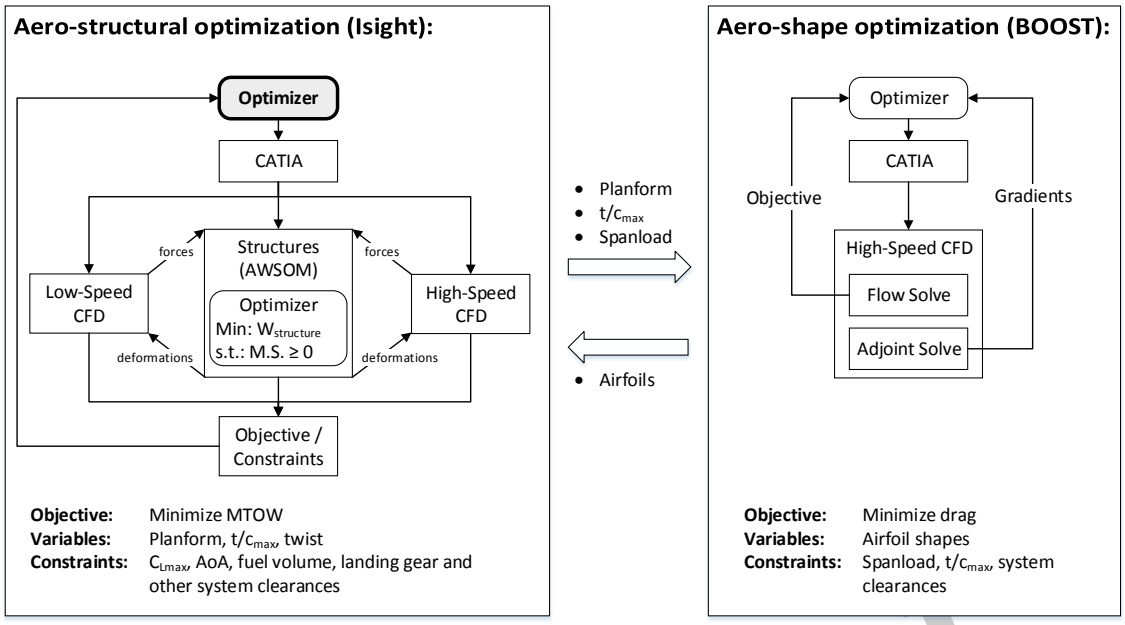


Fig. 1 Simplified schematic of the hybrid-adjoint PMDO workflow.

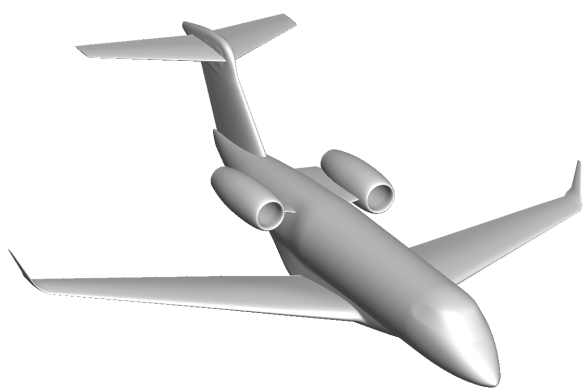


Fig. 2 The baseline BRAC configuration.

variables and constraints are not given here; instead, in the following sections their value will be normalized between 0 and 1.

Formally, the CMDO problem is:

$$\min_{\boldsymbol{x}\in[0,1]^d} \{\mathrm{MTOW}(\boldsymbol{x}) \text{ s.t. } \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{B}\}, \tag{1}$$

where $\boldsymbol{x}$ is the design variable vector of $[0,1]^d$ described in Table 1, d the number of design variables, $\boldsymbol{g}$ is the inequality constraints described in Table 2 and $\boldsymbol{B}$ the associated bounds. Due to confidentiality reason $\boldsymbol{B}$ is not given. Finally for the CMDO optimization problem (1), we consider here 12 design variables and 8 constraints.

Also note that the long-range cruise (i.e. nominal) mission range is satisfied implicitly by the workflow. The baseline BRAC (see Figure 2) as defined in the CMDO workflow is unfeasible. This is inconsequential, since that initial design is not used as a starting point by any of the optimizers. Section V explains which starting point each optimizer uses instead. For industrial reasons, the CMDO of BRAC cannot take more than 8 hours.

Table 1 Design optimization space definition.

Variable index	Variable Name	Variable symbol	Variable index	Variable Name	Variable symbol
$\boldsymbol{x}[0]$	Engine scaling factor	DV_0	$\boldsymbol{x}[4-5]$	Wing rear spar chord-wise locations	DV_4-5
$\boldsymbol{x}[1]$	Wing aspect ratio	DV_1	$\boldsymbol{x}[6]$	Wing sweep	DV_6
$\boldsymbol{x}[2]$	Wing area	DV_2	$\boldsymbol{x}[7]$	Wing taper	DV_7
$\boldsymbol{x}[3]$	Wing inboard trailing edge sweep	DV_3	$\boldsymbol{x}[8-11]$	Wing maximum thickness-to-chord ratios	DV_8-11

Table 2 Constraints definition.

Constraint index	Constraint Name	Bounds index	Constraint index	Constraint Name	Bounds index
$\boldsymbol{g}[0]$	BFL	$\boldsymbol{B}[0]$	$\boldsymbol{g}[4]$	Climb performance	$\boldsymbol{B}[4]$
$\boldsymbol{g}[1]$	ICA	$\boldsymbol{B}[1]$	$\boldsymbol{g}[5]$	High-speed mission range	$\boldsymbol{B}[5]$
$\boldsymbol{g}[2]$	Vref	$\boldsymbol{B}[2]$	$\boldsymbol{g}[6]$	Landing gear spacing	$\boldsymbol{B}[6]$
$\boldsymbol{g}[3]$	Excess fuel	$\boldsymbol{B}[3]$	$\boldsymbol{g}[7]$	Wingtip chord	$\boldsymbol{B}[7]$

B. BRAC PMDO problem

As mentioned earlier, the PMDO portion of this study will consider the aero-structural workflow, shown on the left half of Figure 1. Similar to CMDO, the objective is to minimize MTOW, as in problem (1). The wing and winglet design variables considered are listed in Table 3. The wing area and engine size are kept constant at the PMDO level, and the wing area of the CMDO optimum is used as an input to the plan-form generator. Similarly, the spar locations are fixed using the CMDO result, since the high-lift systems integration is not considered. The constraints imposed during PMDO are listed in Table 4. The volume availability factor is a measure of the fuel volume available in the wing compared to the volume required to meet the range requirements. The maximum wingtip twist deformation is a proxy for a dynamics constraint, and is enforced to prevent an overly flexible wing. Again, for confidentiality reasons, the variable and constraint bounds are omitted. Finally for the PMDO optimization problem (1), we consider here 19 design variables and 5 constraints which is representative of a typical PMDO industrial application.

Table 3 PMDO design optimization space definition.

Variable index	Variable Name	Variable symbol	Variable index	Variable Name	Variable symbol
$\boldsymbol{x}[0]$	Wing span	DV_0	$\boldsymbol{x}[8-13]$	Wing max thickness-to-chord ratios	DV_8-13
$\boldsymbol{x}[1]$	Wing leading edge sweep	DV_1	$\boldsymbol{x}[14]$	Wing inboard trailing edge sweep	DV_14
$\boldsymbol{x}[2]$	Wing break loation	DV_2	$\boldsymbol{x}[15]$	Winglet cant	DV_15
$\boldsymbol{x}[3]$	Wingtip chord	DV_3	$\boldsymbol{x}[16]$	Winglet span	DV_16
$\boldsymbol{x}[4-7]$	Wing twist distribution	DV_4-7	$\boldsymbol{x}[17-18]$	Winglet twist	DV_17-18

V. Optimizers

In this section, we introduce six optimizers to solve the following constrained problem:

$$\min_{\boldsymbol{x}\in\Omega} \{ f(\boldsymbol{x}) \ \text{ s.t. } \ \boldsymbol{g}(\boldsymbol{x}) \geq 0 \}, \tag{2}$$

Preprint

Table 4 PMDO constraints definition.

Constraint index	Constraint Name	Bounds index	Constraints index	Constraint Name	Bounds index
$\boldsymbol{g}[0]$	Volume availability factor	$\boldsymbol{B}[0]$	$\boldsymbol{g}[3]$	Landing gear spacing	$\boldsymbol{B}[3]$
$\boldsymbol{g}[1]$	$C_{L_{max}}$ retracted	$\boldsymbol{B}[1]$	$\boldsymbol{g}[4]$	Wingtip twist deformation	$\boldsymbol{B}[4]$
$\boldsymbol{g}[2]$	Cruise deck angle	$\boldsymbol{B}[2]$			

where $f : \mathbb{R}^d \mapsto \mathbb{R}$ is the objective function, $\boldsymbol{g} : \mathbb{R}^d \mapsto \mathbb{R}^m$ gives the inequality constraints which are expensive to evaluate and $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$ is the vector of design variables. Two of the optimizers are included in the Isight software, which is mandatory to use because of CMDO and PMDO frameworks. We developed an Isight interface for the SEGOMOE python tool-box in which the other optimizers are implemented.

A. Bayesian optimizers

This section introduces the *constrained Bayesian optimization* (CBO) framework [3, 14] that aims to solve the optimization problem (2) with a minimal number of calls. To do so, one uses *Gaussian Process* (GP) (also known as Kriging) [15, 16] trained with a pre-computed *design of experiments* (DoE) (i.e. set of designs evaluated on the objective and constraints functions) of l points. GPs are then used to provide, with a cheap computational cost, a prediction $\mu_s^{(l)} : \mathbb{R}^d \mapsto \mathbb{R}$ and an associated uncertainty $\sigma_s^{(l)} : \mathbb{R}^d \mapsto \mathbb{R}$ for each point of $\boldsymbol{x} \in \Omega$ where $s : \mathbb{R}^d \mapsto \mathbb{R}$ can be either f and g_i for a given constraint component i. Concerning the objective function, these information are combined in an acquisition function $\alpha_f^{(l)} : \mathbb{R}^d \mapsto \mathbb{R}$ [17–19] coding the trade-off between exploration of the highly uncertain domain that can hide a minimum and exploitation of the minimum of the GP prediction. For the constraints, these information are joined to produce a feasibility criterion $\alpha_{\boldsymbol{g}}^{(l)} : \mathbb{R}^d \mapsto \mathbb{R}^m$ [17, 20, 21] which is generally explicit. The point $\boldsymbol{x}^{(l+1)}$, solving the constrained maximization trade-off sub-problem:

$$\boldsymbol{x}^{(l+1)} = \arg\min_{\boldsymbol{x}\in\Omega} \left\{ \alpha_f^{(l)}(\boldsymbol{x}) \ \text{ s.t. } \ \boldsymbol{x} \in \Omega_{\boldsymbol{g}}^{(l)} \right\}, \tag{3}$$

where $\Omega_{\boldsymbol{g}}^{(l)}$ is the approximated feasible domain defined by the feasibility criterion $\boldsymbol{\alpha}_{\boldsymbol{g}}^{(l)}$, is thus iteratively added to the DoE until a maximum number of iterations max_nb_it is reached. The solution provided by CBO to the problem (2) is eventually the best point in the DoE (i.e with the minimal feasible value of f). The main steps of CBO are finally summarized in Algorithm 1.

Algorithm 1 The constrained Bayesian optimization framework.

input : Objective and constraints functions, initial DoE for objective and constraints, a maximum number of iterations max_nb_it

1: **for** $l = 1$ **to** max_nb_it **do**
2: Build the surrogate models using GPs
3: Find $\boldsymbol{x}^{(l+1)}$ a solution of the enrichment maximization sub-problem
4: Evaluate the objective and constraints functions at $\boldsymbol{x}^{(l+1)}$
5: Update the DoE
6: **end for**

output : The best point found in the DoE

In this context, Bartoli et al. [18] implemented a CBO python tool-box based on the *super efficient global optimization* (SEGO) algorithm of Sasena et al. [22] which has been enhanced by the use of *mixture of experts* (MOE) [23, 24], the *kriging with Partial Least Squares* (KPLS) for high dimensional problems [25], additional acquisition functions (e.g. WB2S [18]) for highly multimodal objective functions, supplementary feasibility criteria (e.g. the *upper trust bound* (UTB) [20, 21]) for non-linear constraint functions and a multiprocessing ability to speed-up the optimization process. The tool-box [18] is thus named the *super efficient global optimization with mixture of experts* (SEGOMOE) tool-box. We also developed an interface of the python tool-box to allow its use in the Isight software in which BRAC is implemented.

Preprint

In this paper, the four following variants of SEGO (available with the SEGOMOE tool-box) are tested to show their common ability on industrial MDO problems:

- SEGO [22]: it can be seen as a basic implementation of the super efficient global optimization framework, it was developed for standard constrained optimization problems.
- SEGO-UTB [21]: it includes a decreasing upper trust bound in the SEGO framework. In the context of this paper and to encourage the exploration of the feasible domain, we used an exponential decrease for the constraints learning rate, see [21, Fig. 3].
- SEGOMOE [18]: it combines the use of multiple models (local GPs) with the SEGO framework in order to mitigate the high non-linearity of the targeted optimization problem.
- SEGOMOE-UTB: similarly to SEGO-UTB, it combines the use of the UTB strategy and the SEGOMOE framework. Again, we choose to work with an exponential decrease for the constraints learning rate.

We note that for all the four variants we used the WB2S acquisition function combined with the KPLS models as the dimension of the design space is larger than 12.

B. Isight optimizers

The Isight software [4] is delivered with many ready-to-use constrained optimizers. In this paper, we focus on two of the popular derivative-free Isight constrained optimizers: the *evolutionary optimization algorithm* (Evol) [26] and a variant of the Pointerdog algorithm (Pointer-2) [4].

Evol is an evolutionary strategy that mutates iteratively the best known point by adding a normally distributed perturbation to the design variables. The standard deviation of the normal distribution is adapted during the optimization process to solve the regarded problem with a minimal number of evaluations. This simple algorithm is enhanced by different features such as: (1) a repeat calculation check to ensure that all the points computed are different, (2) a standard deviation expansion process when the same point is always evaluated, (3) a consecutive variable search allowing the exploration in a single canonical direction of the design space, and (4) a parallel execution accelerating the optimization process.

The Pointer-2 strategy is an optimization framework based on an Isight proprietary algorithm that managed a set of well-known optimizers: Evol [26], the Hooke-Jeeves direct search method [27], the *non-linear programming by quadratic Lagrangian* (NLPQL) algorithm [28], the downhill simplex algorithm [29], the *multi-function optimization system tool* (MOST) technique [4] and the *multi-objective particle swarm optimization* (MOPSO) algorithm [30]. The choice of the set of optimizers used in the optimization process is made with respect to a classification system using the information available on the problem (i.e. often given by the user). The best optimizer, and its settings, are then updated thanks to the information collected all along the optimization process. Furthermore, the optimization method can be used in two ways. First, the optimizers and their settings are selected to have the higher improvement leading to the best solution in the shortest time. On the contrary, one searches a robust solution to uncertainties which leads to a slower optimization process due to a higher number of calls. The Pointer-2 strategy periodically performs a surrogate based optimization to speed up the convergence. In fact, the Pointer-2 is able to solve a wide range of constrained optimization problems and thus allows non-specialist to solve an optimization problem only providing the objective and constraints functions, the design variables, and a targeted optimization time.

VI. Results

In this section, we introduce and comment the different tests realized on the BRAC CMDO and PMDO problems. We also recall the aim here is to find the best solution to the optimization problem, the fastest, and the different optimizers are tested in this scope. In the following, the six previously introduced optimizers (see Section V) are tested on the BRAC CMDO and PMDO. For the sake of comparison, we will produce convergence and parallel plots to assess the efficiency of the SEGOMOE tool-box solvers compared to the Isight optimizers.

A. BRAC CMDO results

1. Tests details

The optimizers, introduced in Section V, are compared with the following test plan. We perform 10 independent runs for each solver using 10 different initial DoEs build with the Latin hyper-cube sampling method. For each run, all the solvers are initiated with the same DoE. The size of the initial DoEs is set to $n_{start} = 13$ (i.e $d + 1$ where d is the

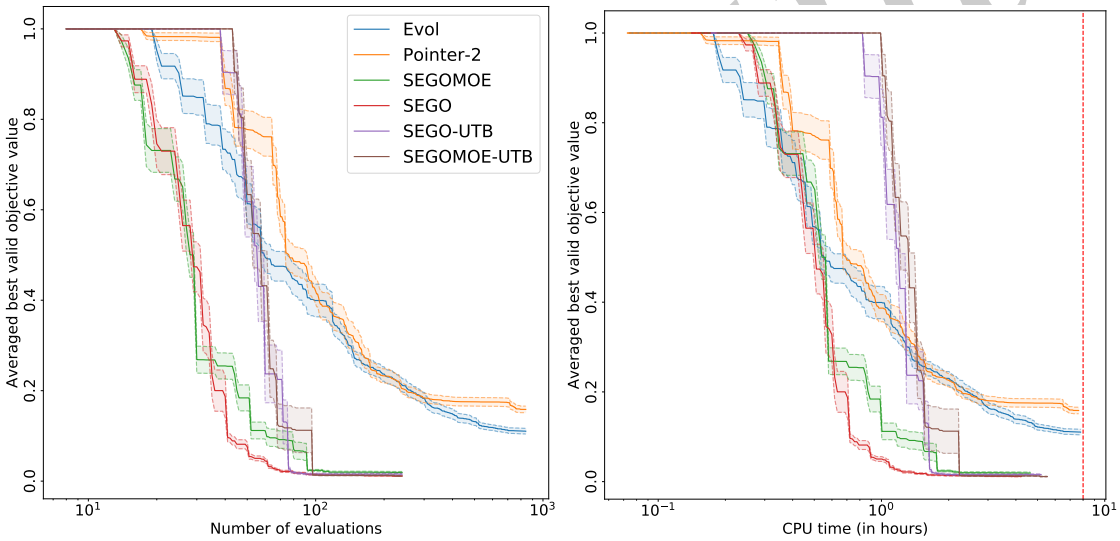
Preprint

dimension of BRAC CMDO problem). Note that Evol and Pointer-2 need a single point to launch the optimization process, we thus provide the point with the best valid value of the DoE. If there is no valid value in the DoE, the point with the minimal constraints violation is provided.

The maximum number of evaluations of the SEGO-like solvers is set to max_it_nb = 227 (i.e. $20d - n_{start}$ meaning a total of $20d = 240$ evaluations is performed). Concerning Evol and Pointer-2, we used the historical options settings of the CMDO framework. The maximum number of evaluations of Evol is set to max_it_nb = 960. For each run, Evol thus performed 973 function evaluations, which takes approximately 8 hours to perform. We respectively fix the maximum allowable job time and the topography type options of Pointer-2 to 8 hours and *nonlinear*. All the remaining options of Evol and Pointer-2 are kept as default. Finally, the best solution among the ten optimizations performed for each optimizer is kept to initiate the BRAC PMDO problem as explained in Section III.B.

2. Convergence plots

To assess the performance of the introduced optimizers, we build two kinds of convergence plots. The first one, named *evaluation convergence plot*, displays the average and the standard deviation of the best valid value over the 10 runs for increasing number of evaluations. The best valid value is defined as the best valid value if there is, at least, one valid point in the DoE, otherwise, a penalization replaces the obtained invalid value. The penalization is the highest feasible value of the objective function ever found. Because of confidentiality reasons, we scale the value of the convergence plots between 0 and 1. The second convergence plot, named *time convergence plot*, also shows the average and the standard deviation of the best valid value over the 10 runs along the optimization time. The penalization is kept the same as in the evaluation convergence plot.



(a) Evaluation convergence plot.

(b) Time convergence plot.

Fig. 3 Convergence plots for the CMDO of BRAC. In the time convergence plot, the vertical red-dashed line indicates the maximum CPU time allowed for an industrial optimization process (8 hours for this test case).

Figure 3 shows that SEGO-like solvers clearly outperform Evol and Pointer-2 in term of the optimum value. In term of number of evaluations, Figure 3a displays that SEGO is converging the fastest to the optimum value even if SEGOMOE has a fastest convergence rate at the beginning of the optimization. Furthermore, note that SEGO-UTB and SEGOMOE-UTB are not converging as fast as SEGO and SEGOMOE because of their extensive exploratory behaviour. When looking to the optimization time, Figure 3b reveals that the SEGO-like solvers converge in approximately 2 hours meanwhile Evol and Pointer-2 does not converge after 8 hours.

To conclude, these convergence plots have shown that the SEGO-like solvers provide a better solution in less running CPU-time than Evol and Pointer-2. We also note that the use of the UTB option within SEGO and SEGOMOE seems to not lead to any improvement on the obtained performance which suggests that the exploration of the feasible domain was not a difficult task to handle on this test case.

3. Parallel plots

The parallel plots introduced in the following depict the behaviour of the tested solvers along the optimization process in which targeted values are displayed (e.g., the explored design variables). Here, the plotted values are the number of iterations, the design variables values, the objective function value and the constraints violation. Furthermore, several colors are used to distinguish the reference design in red (i.e. the best feasible design found so far by all the tested optimizers, here SEGO), the optimum design found by the regarded solver in black, the feasible explored designs in green, and the unfeasible designs in blue. Due to the stochastic nature of our tests, we build our parallel plots using a median run in the following way. For each run of the optimizer, we store the best valid objective value; if none of the runs converges to a feasible point, we collect the minimal violation explored by the optimizer. The median run is then selected based on the stored values for all runs.

Figures 4 and 5 display the parallel plots of the median run of the tested optimizers. First, note that all the SEGO-like optimizers are converging to the optimum. On the contrary, Evol and Pointer-2 do not converge to the optimum, as depicted by Figures 4a and 4b. Furthermore, Evol is not able to explore the entire domain as implied by the grouped green lines of Figure 4a. Secondly, note the different solutions of the SEGO-like solvers even if they all converge approximately to the same objective function value. Indeed, the values for DV_8 and DV_11 (i.e. wing maximum thickness-to-chord ratios) are different from an optimizer to another. This can be due to either a local optimum or an inactive design variable (i.e. the objective function does not change along the design variable). From the expert point of view, the behavior observed on the BRAC CMDO problem is due to inactive design variables.

B. PMDO results

1. Tests details

The optimizers are again compared on the BRAC PMDO problem. Because of the computation time of the BRAC PMDO problem (ca. $25min$), we only performed one optimization for each solver. Each of the optimizers is initiated with the same DoE of $n_{start} = 20$ sample points using the Latin hyper-cube sampling method. To mimic the introduced MDO process, we also add to the initial DoE the point corresponding to the best solution $\boldsymbol{x}^*$ of all the BRAC CMDO process. As in Section VI.A.1, Evol and Pointer-2 only need a single point to launch the optimization, we thus provide the best solution x^* of all the BRAC CMDO cases.

The maximum number of evaluations of the 4 SEGO-like solvers is set to max_it_nb = 170 (i.e. $10d - n_{start} - 1$ meaning a total of $10d = 190$ evaluations). The maximum number of evaluations of Evol and Pointer-2 is set to 510 to follow historical values used in the PMDO framework, and the parallel batch size is set to 4 for both. In addition, a *smooth* topography type is selected for Pointer-2. All the other settings are kept to default values.

2. Convergence plots

The convergence plots defined in this Section are slightly different from the ones introduced in Section VI.A.2. Indeed, only one optimization is here performed for each of the 6 optimizers. Thus Figure 6 displays the best valid values of the regarded solvers for increasing time or number of evaluations. First, Figure 6a shows that none of the solvers converge to the same optimum value, with the best value obtained with SEGO. Similar values are also achieved with Evol and SEGOMOE. Furthermore, Pointer-2 provides the worst value of all the solvers with an optimum at 0.1. The other algorithms find better solution values than Pointer-2, but worse than SEGO, SEGOMOE and Evol. Note that SEGO and SEGOMOE are able to find a similar solution as Evol with 30 fewer evaluations.

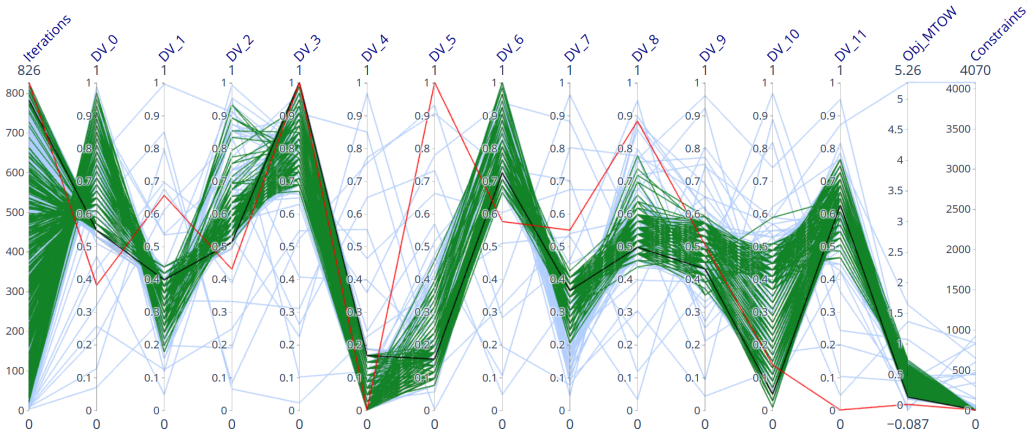
Secondly, in terms of convergence time, Evol and Pointer-2 benefit from the batch evaluation of the objective function which is not the case for the SEGO-like solvers. Indeed, the 220 necessary evaluations of Evol to converge are done in less than 40 hours, whereas the SEGO-like solvers need 80 hours for 190 evaluations.

To conclude, Figure 6 shows that the UTB feasibility criterion is not useful for the BRAC CMDO problem, whereas SEGO and SEGOMOE show great performances with a small budget. Moreover, the batch evaluation capability of Evol speeds up the convergence time.

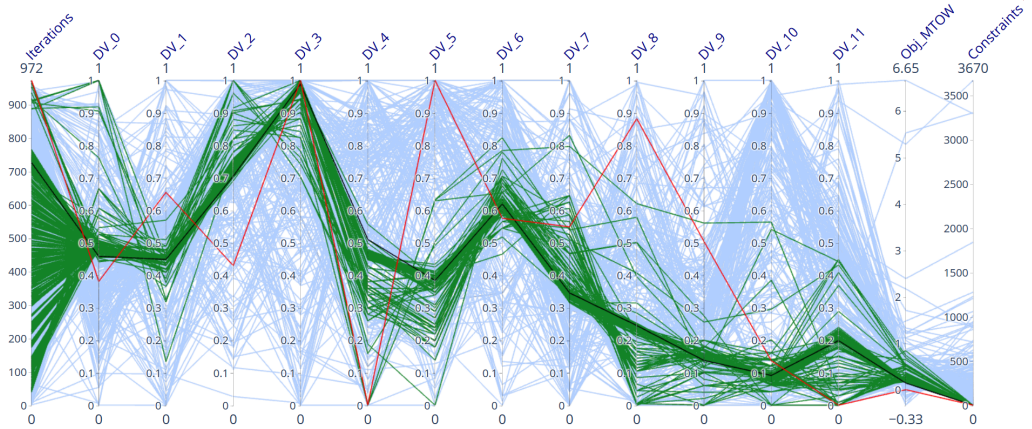
3. Parallel plots

Here, no run selections are needed for the parallel plots. Indeed, only a single run is performed for the BRAC PMDO problem for each solver. They are produced in the same way as in Section VI.A.3. Figures 7 and 8 show the parallel plots for Evol, Pointer-2, SEGO, SEGOMOE, SEGO-UTB and SEGOMOE-UTB.

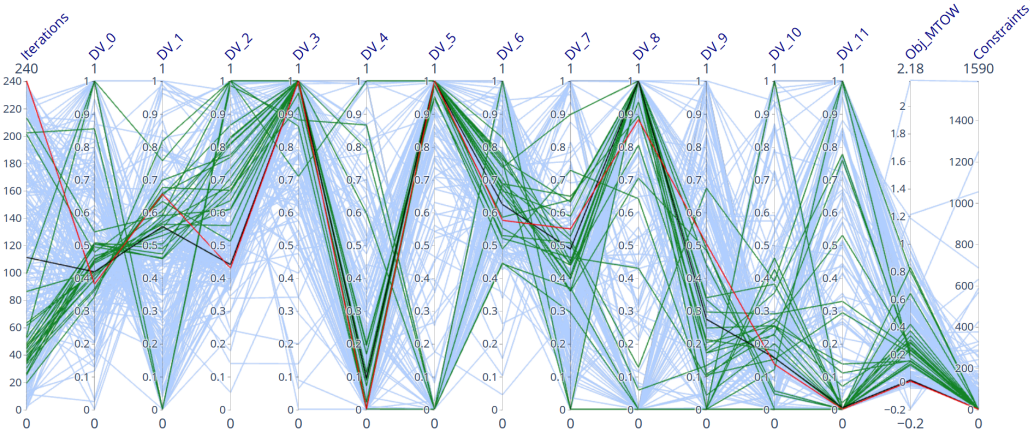
Preprint



(a) Evol.

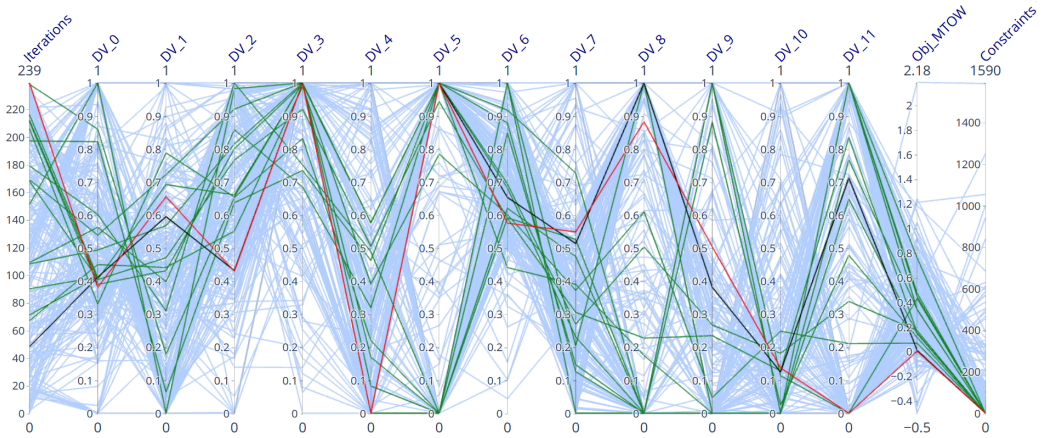


(b) Pointer-2.

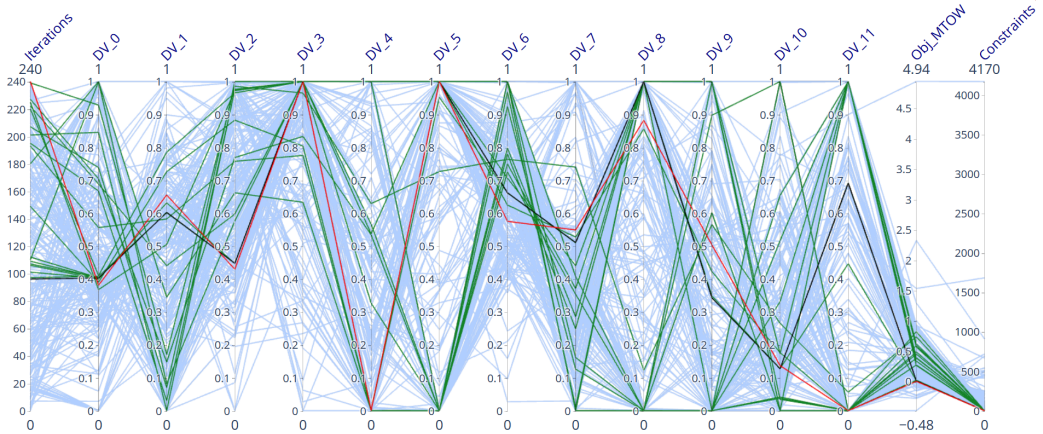


(c) SEGO.

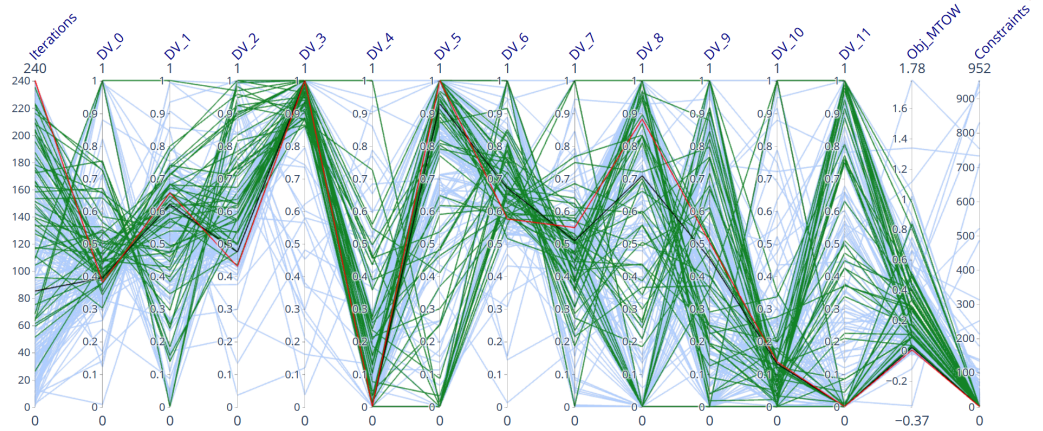
Fig. 4 Parallel plots using the median run for the CMDO of BRAC for Evol, Pointer-2 and SEGO. In blue: the unfeasible designs; in green: the feasible designs; in black: the optimum; in red: the reference design.



(a) SEGO-UTB.

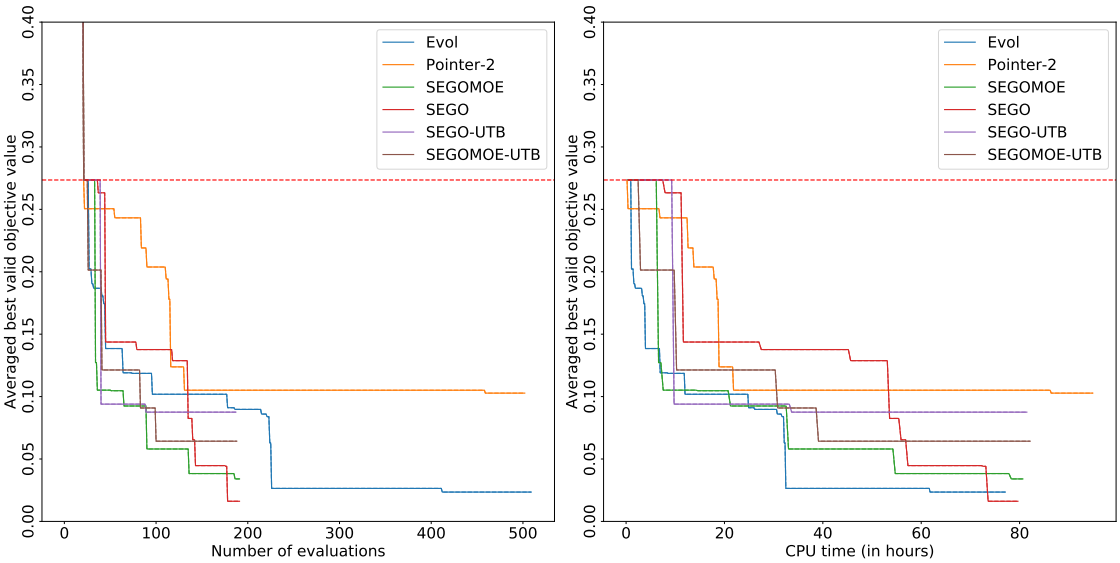


(b) SEGOMOE-UTB.



(c) SEGOMOE.

Fig. 5 Parallel plots using the median run for the CMDO of BRAC for SEGOMOE, SEGOMOE-UTB and SEGO-UTB. In blue: the unfeasible designs; in green: the feasible designs; in black: the optimum; in red: the reference design.



(a) Evaluation convergence plot.

(b) Time convergence plot.

Fig. 6 Convergence plots for the PMDO of BRAC. The red dashed line is the value of the BRAC CMDO design in the BRAC PMDO process.

Preprint

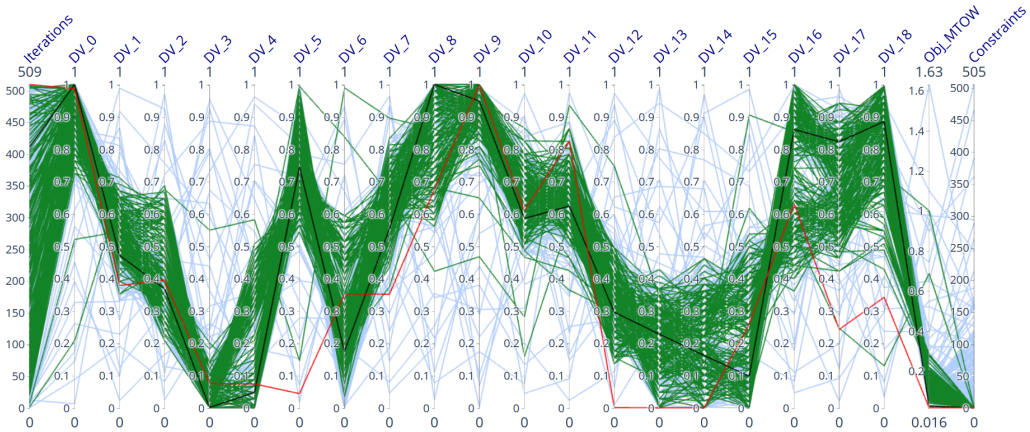
As mentioned, SEGO finds the best design, followed closely by Evol and SEGOMOE. These three solutions share similar main plan-form features, namely the wing span (DV_0) and sweep (DV_1), while there is some variation in the wing twist distribution (DV_4-7) and maximum thickness-to-chord ratios (DV_8-13), as shown in Figures 7a and 8c. Only SEGO converges to the lower thickness bound for the two most outboard profiles (DV_12 and DV_13). The winglet has a secondary impact on the MTOW, and as a result even greater variation is seen in the optimized winglet parameters (DV_15-18). The best solutions of the other three optimizers deviate even further from the SEGO optimum. It is possible that all the optimizers converge to different local optima, but the variability in the solutions is more likely due to incomplete convergence of the algorithms, since the number of iterations was limited to meet the time restrictions of an industrial application.

VII. Conclusion

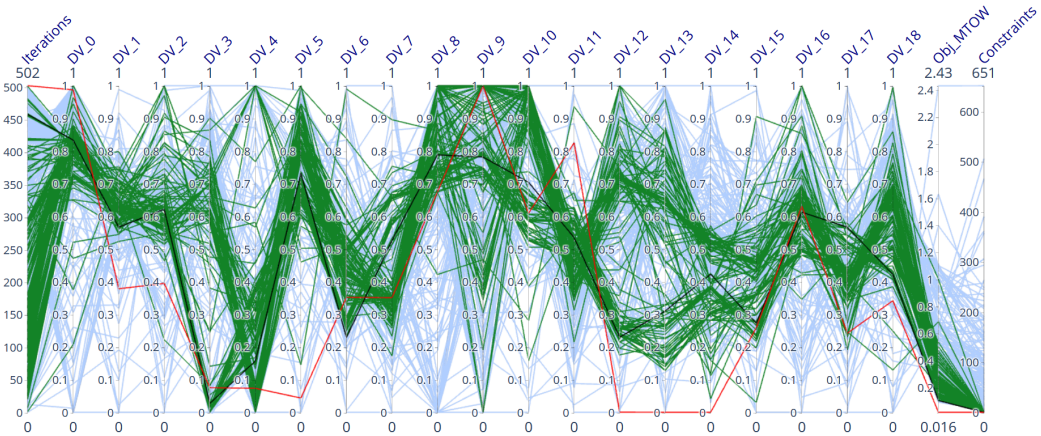
The multi-level optimization framework developed at Bombardier Aviation aims to help the design of efficient and competitive aircraft. The two optimization levels, implemented within the Isight software, were discussed in this paper: *conceptual and preliminary multi-disciplinary optimization* (CMDO and PMDO). The use of the ready-to-use Isight optimizers have shown great results in the past but within a minimum of 8 hours for the CMDO study case and a week for the PMDO one. The SEGOMOE python tool-box was investigated in this paper and led to significant improvements on Bombardier research aircraft configuration test cases. In particular, on the CMDO test case, we showed that all the tested solvers (implemented within the SEGOMOE tool-box) have outperformed two of the best Isight optimizers both in terms of the required computational effort as well as on the CPU running time. On the PMDO test case, the results are more contrasted. SEGO is able to provide a better solution than Evol in less evaluations but with more computational time. This is due to the batch evaluation capability of Evol. The batch evaluation capability of the SEGOMOE tool-box is a future improvement work to perform.

Acknowledgments

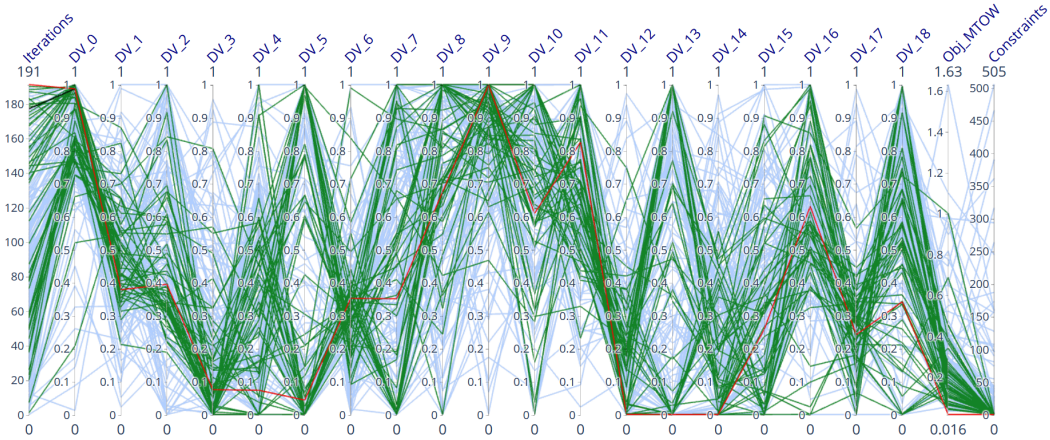
This work is part of the activities of ONERA - ISAE - ENAC joint research group in a context of a partnership between ONERA, ISAE-SUPAERO and Bombardier Aviation. In addition, we thank Jasveer Singh for his work on the BRAC CMDO specification.



(a) Evol.

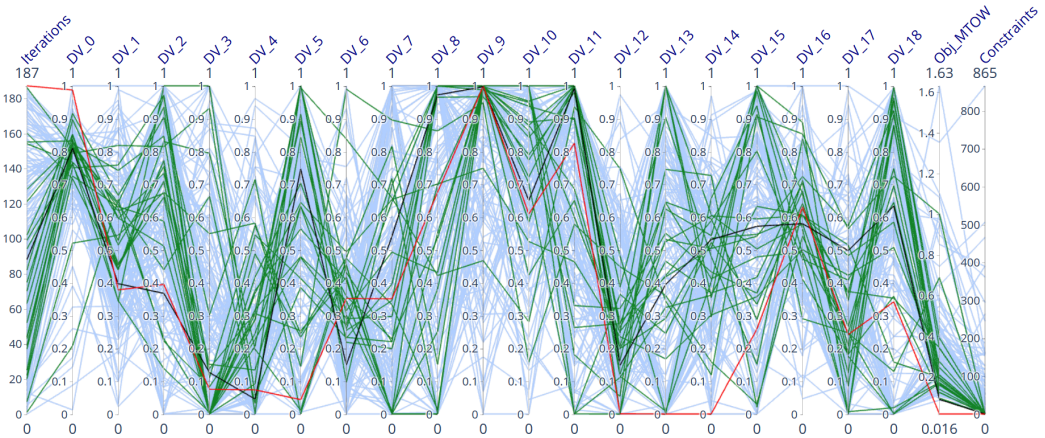


(b) Pointer-2.

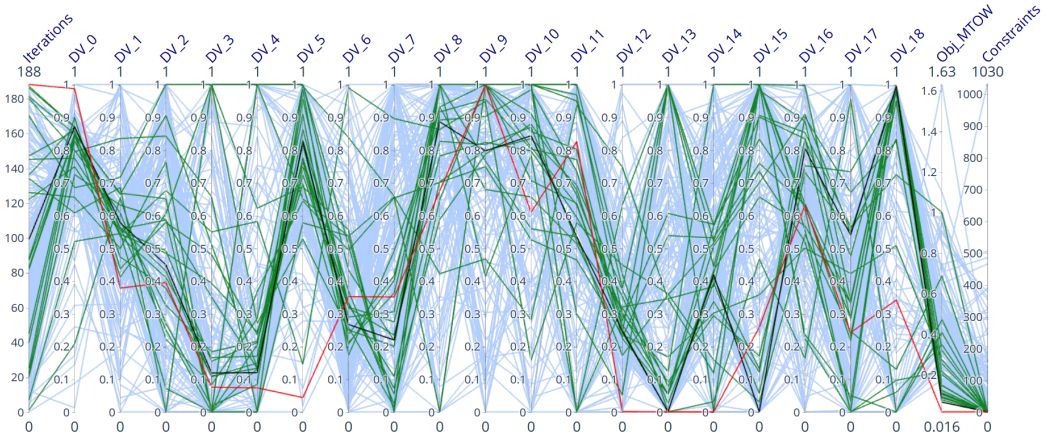


(c) SEGO.

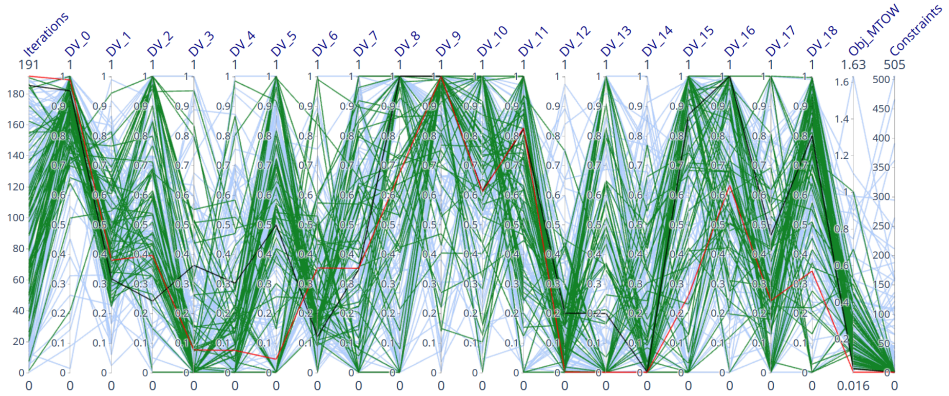
Fig. 7 Parallel plots for the PMDO of BRAC for Evol, Pointer-2 and SEGO. In blue: the unfeasible designs; in green: the feasible designs; in black: the optimum; in red: the reference design.



(a) SEGO-UTB.



(b) SEGOMOE-UTB.



(c) SEGOMOE.

Fig. 8 Parallel plots for the PMDO of BRAC for SEGO-UTB, SEGO-UTB and SEGOMOE. In blue: the unfeasible designs; in green: the feasible designs; in black: the optimum; in red: the reference design.

References

[1] Piperni, P., Abdo, M., and Kafyeke, F., "The application of multi-disciplinary optimization technologies to the design of a business jet," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004, p. 4370.

[2] Piperni, P., DeBlois, A., and Henderson, R., "Development of a Multilevel Multidisciplinary-Optimization Capability for an Industrial Environment," *AIAA Journal*, Vol. 51, 2013, pp. 2335–2352.

[3] Močkus, J., "On Bayesian Methods for Seeking the Extremum," *Optimization Techniques IFIP Technical Conference*, 1975, pp. 400–404.

[4] Van der Velden, A., and Koch, P., "Isight design optimization methodologies," *ASM handbook*, Vol. 22, 2010, p. 79.

[5] Piperni, P., and Boudreau, J., "The Evolution of Structured Grid Generation at Bombardier Aerospace," *50th Canadian Air and Space Institute Annual General Meeting and Conference*, 2003.

[6] Mohamed, K., Laurendeau, E., and Nadarajah, S., "Implementation and Validation of Detached Eddy Simulation in the Bombardier Aerospace Navier-Stokes Flow Solver," *56th Canadian Air and Space Institute Aeronautics Conference*, 2009.

[7] Deblois, A., and Abdo, M., "Multi-Fidelity Multidisciplinary Design Optimization of Metallic and Composite Regional and Business Jets," *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference 2010*, 2010.

[8] Valarezo, W., and Chin, V., "Method for the Prediction of Wing Maximum Lift," *Journal of Aircraft*, Vol. 31, 1994, pp. 103–109.

[9] Maskew, B., "PROGRAM VSAERO: A computer program for calculating the non-linear aerodynamic characteristics of arbitrary configurations: User's manual," 1982.

[10] Piperni, P., Abdo, M., Kafyeke, F., and Isikveren, A. T., "Preliminary Aerostructural Optimization of a Large Business Jet," *Journal of Aircraft*, Vol. 44, 2007, pp. 1422–1438.

[11] Sermeus, K., Mohamed, K., Laurendeau, E., and Nadarajah, S., "Development of an Industrial Aerodynamic Shape Optimization Strategy Using a Discrete Adjoint Method," *18th Annual Conference of the CFD Society of Canada*, 2010, p. 6.

[12] Reist, T. A., Koo, D., Zingg, D. W., Bochud, P., Castonguay, P., and Leblond, D., "Cross-Validation of High-Fidelity Aerodynamic Shape Optimization Methodologies for Aircraft Wing-Body Optimization," *AIAA Journal*, Submitted 2019.

[13] Deblois, A., "A New Hybrid-Adjoint MDO Framework for High-Fidelity Aerostructural Optimization," *SAE 2013 Aerotech Congress and Exhibition*, 2013.

[14] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global optimization*, Vol. 13, 1998, pp. 455–492.

[15] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[16] Krige, D. G., "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand," *Journal of the Southern African Institute of Mining and Metallurgy*, Vol. 52, 1951, pp. 119–139.

[17] Frazier, P. I., "A tutorial on Bayesian optimization," *aarXiv:1807.02811*, 2018.

[18] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Priem, R., Bons, N., Martins, J., and Morlier, J., "Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design," *Aerospace Science and technology*, Vol. 90, 2019, pp. 85–102.

[19] Wang, Z., and Jegelka, S., "Max-Value Entropy Search for Efficient Bayesian Optimization," *34th International Conference on Machine Learning*, Vol. 70, 2017, pp. 3627–3635.

[20] Lam, R., Allaire, D., and Willcox, K., "Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, p. 143.

[21] Priem, R., Bartoli, N., and Diouane, Y., "On the Use of Upper Trust Bounds in Constrained Bayesian Optimization Infill Criteria," *AIAA Aviation 2019 Forum*, 2019, p. 2986.

[22] Sasena, M. J., Papalambros, P., and Goovaerts, P., "Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization," *Engineering optimization*, Vol. 34, 2002, pp. 263–278.

[23] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J., "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, Vol. 27, 2005, pp. 83–85.

Preprint

[24] Bettebghor, D., Bartoli, N., Grihon, S., Morlier, J., and Samuelides, M., "Surrogate modeling approximation using a mixture of experts based on EM joint estimation," *Structural and multidisciplinary optimization*, Vol. 43, No. 2, 2011, pp. 243–259.

[25] Bouhlel, M. A., Bartoli, N., Otsmane, A., and Morlier, J., "Improving Kriging Surrogates of High-Dimensional Design Models by Partial Least Squares Dimension Reduction," *Structural and Multidisciplinary Optimization*, Vol. 53, 2016, pp. 935–952.

[26] Schwefel, H., "Evolutionsstrategie und numerische Optimierung [Evolution strategy and numerical optimization]," Ph.D. thesis, Fachbereich Verfahrenstechnik, Technische Universität Berlin, 1975.

[27] Hooke, R., and Jeeves, T. A., ""Direct Search" Solution of Numerical and Statistical Problems," *Journal of the ACM*, Vol. 8, 1961, pp. 212–229.

[28] Schittkowski, K., "NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems," *Annals of operations research*, Vol. 5, 1986, pp. 485–500.

[29] Nelder, J. A., and Mead, R., "A simplex method for function minimization," *The computer journal*, Vol. 7, 1965, pp. 308–313.

[30] Coello, C. A., Pulido, G. T., and Lechuga, M. S., "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on evolutionary computation*, Vol. 8, 2004, pp. 256–279.

Preprint