Multi-objective aerodynamic design exploration using multi-fidelity methods and pareto set identification techniques

Anand Amrit

Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Aerospace Engineering Commons

Recommended Citation

https://lib.dr.iastate.edu/etd/16782

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Multi-objective aerodynamic design exploration using multi-fidelity methods and pareto set identification techniques

by

Anand Amrit

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Program of Study Committee:
Leifur Leifsson, Major Professor
Anupam Sharma
Peng Wei
Thomas Ward
Alric P Rothmayer

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa

2018

Copyright © Anand Amrit, 2018. All rights reserved.
DEDICATION

I dedicate this thesis to my Father Mr. Ajaya Kumar Nayak and my Mother Mrs. Narayani Nayak, who supported me on this adventure.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>xi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>xiii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xiv</td>
</tr>
<tr>
<td>CHAPTER 1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>I. Motivation and Challenges</td>
<td>1</td>
</tr>
<tr>
<td>II. Research Goals and Questions</td>
<td>5</td>
</tr>
<tr>
<td>III. Research Contributions</td>
<td>6</td>
</tr>
<tr>
<td>A. List of Journal Articles</td>
<td>7</td>
</tr>
<tr>
<td>B. List of Conference Proceedings</td>
<td>8</td>
</tr>
<tr>
<td>C. List of Conference Talks</td>
<td>10</td>
</tr>
<tr>
<td>IV. Thesis Outline</td>
<td>11</td>
</tr>
<tr>
<td>References</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER 2. FAST MULTI-OBJECTIVE AERODYNAMIC OPTIMIZATION USING SERIAL</td>
<td>17</td>
</tr>
<tr>
<td>DOMAIN PATCHING AND MULTI-FIDELITY MODELS</td>
<td></td>
</tr>
<tr>
<td>Nomenclature</td>
<td>17</td>
</tr>
<tr>
<td>I. Introduction</td>
<td>19</td>
</tr>
<tr>
<td>II. Methodology</td>
<td>24</td>
</tr>
<tr>
<td>A. Definition of the Pareto Front</td>
<td>24</td>
</tr>
<tr>
<td>B. Pareto Front Exploration Using Sequential Domain Patching</td>
<td>25</td>
</tr>
<tr>
<td>C. Multi-Fidelity, Multi-Objective Aerodynamic Sequential Domain Patching</td>
<td>26</td>
</tr>
<tr>
<td>Algorithm</td>
<td></td>
</tr>
<tr>
<td>D. Automated Determination of Patch Sizes</td>
<td>31</td>
</tr>
<tr>
<td>E. Pareto Set Refinement</td>
<td>32</td>
</tr>
<tr>
<td>III. Numerical Examples</td>
<td>32</td>
</tr>
<tr>
<td>A. Analytical Problems</td>
<td>32</td>
</tr>
<tr>
<td>1. Zitzler–Deb–Thiele's function N. 1 (ZDT 1)</td>
<td>32</td>
</tr>
<tr>
<td>2. Fonseca and Fleming Function</td>
<td>34</td>
</tr>
<tr>
<td>B. Transonic Airfoil Design</td>
<td>35</td>
</tr>
<tr>
<td>1. Problem Description</td>
<td>35</td>
</tr>
<tr>
<td>2. Design Variables</td>
<td>36</td>
</tr>
<tr>
<td>3. High-Fidelity Viscous Aerodynamics Model</td>
<td>37</td>
</tr>
<tr>
<td>4. Low-Fidelity Viscous Aerodynamics Model</td>
<td>38</td>
</tr>
<tr>
<td>5. Single-Objective Optimization Results</td>
<td>38</td>
</tr>
<tr>
<td>6. The Pareto Front</td>
<td>42</td>
</tr>
</tbody>
</table>
IV. Conclusion......................................................................................................................... 44
References.......................................................................................................................... 45

CHAPTER 3. MULTI-FIDELITY AERODYNAMIC DESIGN TRADE-OFF
EXPLORATION USING POINT-BY-POINT PARETO SET IDENTIFICATION ...... 50
Nomenclature ....................................................................................................................... 50
I. Introduction ..................................................................................................................... 51
II. Methodology .................................................................................................................. 57
   A. Definition of the Pareto Front .................................................................................... 58
   B. Point-by-Point Exploration of the Pareto Front ....................................................... 58
   C. Multi-Fidelity, Point-by-Point Multi-Objective Optimization Algorithm .......... 60
III. Numerical Examples .................................................................................................... 64
   A. Analytical Problems .................................................................................................. 65
      1. Zitzler-Deb-Thiele’s function N. 1 (ZDT 1) ......................................................... 65
      2. Zitzler-Deb-Thiele’s function N. 2 (ZDT 2) ......................................................... 66
      3. Kursawe function ................................................................................................. 67
   B. Two-Dimensional Aerodynamic Design Problem .................................................. 68
      1. Problem Description .............................................................................................. 68
      2. Design Variables .................................................................................................. 69
      3. High-Fidelity Viscous Aerodynamics Model ...................................................... 70
      4. Low-Fidelity Viscous Aerodynamics Model ....................................................... 71
      5. Single-Objective Optimization Results .............................................................. 72
      6. Point-by-Point Pareto Front Exploration ............................................................ 75
   C. Algorithm Scalability................................................................................................. 77
   D. Algorithm Robustness .............................................................................................. 82
IV. Conclusion ..................................................................................................................... 83
References .......................................................................................................................... 83

CHAPTER 4. APPLICATIONS OF SURROGATE-ASSISTED AND MULTI-
FIDELITY MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS TO
SIMULATION-BASED AERODYNAMIC DESIGN.............................................................. 92
Nomenclature ....................................................................................................................... 93
I. Introduction ..................................................................................................................... 94
II. Methods .......................................................................................................................... 98
   A. Definition of the Pareto Front .................................................................................. 98
   B. Algorithm 1: Pareto Front Exploration Using Surrogate-Assisted Multi-
      Fidelity Based Multi-Objective Evolutionary Algorithm ........................................ 99
   C. Algorithm 2: Pareto Front Exploration Using Sequential Domain Patching
      Algorithm .................................................................................................................. 101
   D. Algorithm 3: Pareto Front Exploration Using the Point-by-Point
      Algorithm .................................................................................................................. 104
III. Numerical Examples .................................................................................................... 107
   A. Analytical Problem .................................................................................................. 108
   B. Transonic Airfoil Design .......................................................................................... 109
      1. Problem Description .............................................................................................. 110
      2. Design Variables .................................................................................................. 110
      3. High-Fidelity Viscous Aerodynamics Model ....................................................... 111
4. Low-Fidelity Viscous Aerodynamics Model ........................................ 112
5. Single-Objective Optimization (SOO) Results .................................. 113
6. The Pareto Front ............................................................................. 113
C. Subsonic Rectangular Wing Design .................................................. 117
  1. Problem Description .................................................................... 117
  2. Design Variables ......................................................................... 118
  3. Computational Model .................................................................. 119
  4. Single-Objective Optimization (SOO) Results ............................... 121
  5. Pareto-Front ................................................................................ 122
IV. Conclusion ..................................................................................... 128
References .......................................................................................... 130

CHAPTER 5. CONCLUSION .................................................................... 135
I. Main Contributions ........................................................................... 135
   A. Multi-Fidelity Aerodynamic Modeling .......................................... 135
   B. Multi-Fidelity Aerodynamic Design Exploration Using Sequential Domain Patching ................................................................. 135
   C. Multi-Fidelity Aerodynamic Design Exploration Using the Point-by-Point Method .............................................................. 136
   D. A Comparison of the Proposed Approaches ................................. 136
   E. Answers to the Research Questions ............................................... 137
II. Suggestions for Future Work ........................................................... 139
LIST OF FIGURES

Figure 2-1. Exploration of (a) the entire Pareto front and (b) only part of the Pareto front. ................................................................. 25

Figure 2-2. Pictorial representation of the sequential domain patching method: (a) patches in the design space (b) refinement of the Pareto front in the objective space. ......................................................... 26

Figure 2-3. Sequential domain patching based aerodynamic design exploration. .......... 29

Figure 2-4. Representation of high and low-fidelity evaluation values of 5 random designs. ........................................................................ 33

Figure 2-5. Pareto front obtained for the ZDT 1 function: (a) design space (b) feature space. ................................................................. 34

Figure 2-6. Pareto front obtained for the Fonseca and Fleming function: (a) Design space (b) Feature space. ......................................................... 35

Figure 2-7. Airfoil computational models: (a) airfoil shape parameterization, (b) hyperbolic C-mesh. ............................................................. 36

Figure 2-8. SOO results showing baseline and optimized (a) airfoil shapes, and (b) pressure distributions at \( M_{\infty} = 0.734, C_l = 0.824 \) and \( Re_{\infty} = 6.5 \times 10^6 \). ........ 41

Figure 2-9. SOO Mach contours at \( M_{\infty} = 0.734, C_l = 0.824 \) and \( Re_{\infty} = 6.5 \times 10^6 \) of (a) the baseline airfoil, (b) the SOO optimal design, \( x1^* \), and (c) the SOO optimal design, \( x2^* \). ................................................................. 41

Figure 2-10. The final refined Pareto front at \( M_{\infty} = 0.734, C_l = 0.824 \) and \( Re_{\infty} = 6.5 \times 10^6 \). .................................................................................. 43

Figure 2-11. MOO results showing (a) the airfoil shapes, and (b) the pressure distributions at \( M_{\infty} = 0.734, C_l = 0.824 \) and \( Re_{\infty} = 6.5 \times 10^6 \) ........... 43

Figure 2-12. Mach contours at \( M_{\infty} = 0.734, C_l = 0.824, Re_{\infty} = 6.5 \times 10^6 \) of (a) point 1, (b) point 2. ................................................................. 44
Figure 3-1. Hypothetical representation of the point-by-point Pareto front exploration progression shown for (a) two design variables in the design and (b) two objectives in the feature space. ........................................... 59

Figure 3-2. Representation of Pareto optimal solutions obtained by traversing in any direction .......................................................................................................................... 60

Figure 3-3. Flowchart of the point-by-point aerodynamic design exploration algorithm .......................................................................................................................... 60

Figure 3-4. Representation of high and low-fidelity evaluation values of few random designs. .......................................................................................................................... 66

Figure 3-5. ZDT 1 results: Comparison of Pareto obtained with the actual Pareto. ........ 66

Figure 3-6. ZDT 2 results: Comparison of Pareto obtained with the actual Pareto. ........ 67

Figure 3-7. Kursawe function results: comparison of Pareto obtained with the actual Pareto. .......................................................................................................................... 68

Figure 3-8. Airfoil shape parameterization using B-spline curves. ................................ 69

Figure 3-9. Hyperbolic C-mesh: (a) farfield view, (b) view close to the surface. .......... 71

Figure 3-10. Viscous flow simulation results for RAE 2822 at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$: (a) the evolution of the lift and drag coefficients obtained by the low-fidelity model, (b) a comparison of the pressure distributions obtained by the high- and low-fidelity models. ....... 72

Figure 3-11. SOO showing (a) convergence of arguments, and (b) evolution of the objective. .......................................................................................................................... 74

Figure 3-12. SOO results showing baseline and optimized (a) airfoil shapes, and (b) pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$. ........ 74

Figure 3-13. SOO Mach contours at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$ of (a) the baseline airfoil, and (b) the SOO optimal design. .................................................. 75

Figure 3-14. Multi-objective optimization results at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$: (a) optimum solutions, (b) zoomed-in plot ....................... 76

Figure 3-15. MOO results showing (a) the airfoil shapes, and (b) the pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$ .............................. 77

Figure 3-16. Mach contours at $M_\infty = 0.734$, $C_l = 0.824$ of (a) Point 1, (b) Point 2. ........ 77
Figure 3-17. SOO results showing the variation of (a) the drag coefficient and (b) total optimization time as a function of the number of design variables. .................................................. 79

Figure 3-18. SOO results showing (a) the airfoil shapes, and (b) the pressure distributions at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$ .................................................. 80

Figure 3-19. SOO results showing the Mach contours at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$ of the optimal shapes for (a) 10 and (b) 12 design variables................................................................. 80

Figure 3-20. Comparison of the Pareto fronts obtained using 8, 10 and 12 design variables at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$. .................................................. 81

Figure 3-21. Robustness of the proposed algorithms is demonstrated by obtaining comparable (within one drag count) Pareto fronts when starting from two different initial points. ................................................................. 82

Figure 4-1. An illustration of the SDP algorithm showing (a) the design space, and (b) the Pareto front in the feature space................................................................. 102

Figure 4-2. An illustration of the point-by-point Pareto front exploration progression for (a) two design variables in the design space and (b) two objectives in the feature space. ................................................................. 106

Figure 4-3. Example high and low-fidelity evaluation values of five random designs. . 109

Figure 4-4. Pareto front obtained for Fonseca and Fleming function (a) design space, (b) feature space. ................................................................. 109

Figure 4-5. Airfoil models: (a) shape parameterization, (b) hyperbolic C-mesh........... 112

Figure 4-6. The final Pareto fronts from all three algorithms at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$ (a) final refined Pareto (b) zoomed in view of the same Pareto front. ................................................................. 114

Figure 4-7. MOO results showing (a) the airfoil shapes, and (b) the pressure distributions at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$ for different points along the Pareto. ................................................................. 115

Figure 4-8. Mach contours at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$ of (a) Point 1, (b) Point 2. ................................................................. 115

Figure 4-9. Baseline semi-span wing geometry with a constant section profile. ............ 118
Figure 4-10. Wing shape parameterization using B-spline curves for chord length, wing quarter-chord height and twist .......................................................... 119

Figure 4-11. Sample mesh used for the computational fluid dynamics simulations. ..... 120

Figure 4-12. Grid convergence results for the baseline design at $M_\infty = 0.5$ and $C_L = 0.2625$: (a) drag coefficient versus number of mesh cells, (b) simulation run time .................................................................................................. 121

Figure 4-13. Final Pareto front obtained from each algorithm: (a) optimum solutions and (b) zoomed-in plot. ........................................................................................................ 124

Figure 4-14. Wing shape parameter comparison for Baseline, Point 1 and Point 2 designs for (a) chord length, (b) wing quarter-chord height and (c) twist. ................................................................. 125

Figure 4-15. Pressure contours at $M_\infty = 0.5$ and $C_L = 0.2625$ (a) Point 1 (top view), (b) Point 2 (top view), (c) Point 1 (side view) and (d) Point 2 (side view). .......................................................................................................................... 126

Figure 4-16. MOO results showing the pressure distributions of two points on the Pareto at $M_\infty = 0.5$ and $C_L = 0.2625$ and at (a) $\eta = 0.02$, (b) $\eta = 0.5$, (c) $\eta = 0.75$ and (d) $\eta = 1$ .......................................................................................................................... 127

Figure 4-17. A comparison of the wing lift distributions for the Point 1 and Point 2 designs. .................................................................................................................. 128
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table 2-1. Grid convergence study for the baseline shape.</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2-2. Single objective optimization results.</td>
<td>40</td>
</tr>
<tr>
<td>Table 3-1. Grid convergence study for the baseline shape of Benchmark Case 2.</td>
<td>71</td>
</tr>
<tr>
<td>Table 3-2. Single objective optimization (SOO) results.</td>
<td>73</td>
</tr>
<tr>
<td>Table 3-3. Aerodynamic coefficients of the baseline design as a function of the number of design variables.</td>
<td>79</td>
</tr>
<tr>
<td>Table 3-4. SOO results as a function of the number of design variables.</td>
<td>79</td>
</tr>
<tr>
<td>Table 3-5. Comparison of the computational cost of the MOO as a function of the number of design variables.</td>
<td>81</td>
</tr>
<tr>
<td>Table 4-1. Grid convergence study for the baseline shape.</td>
<td>112</td>
</tr>
<tr>
<td>Table 4-2. Single objective optimization results for transonic airfoil design case.</td>
<td>114</td>
</tr>
<tr>
<td>Table 4-3. Cost of each multi-fidelity MOO algorithm for the transonic airfoil design case.</td>
<td>116</td>
</tr>
<tr>
<td>Table 4-4. Single-objective optimization results for the subsonic wing design.</td>
<td>121</td>
</tr>
<tr>
<td>Table 4-5. Wing shape parameter comparison for Baseline, Point 1 and Point 2 designs at multiple wing span stations.</td>
<td>125</td>
</tr>
<tr>
<td>Table 4-6. Cost of each MOO algorithm for subsonic wing design.</td>
<td>128</td>
</tr>
</tbody>
</table>
NOMENCLATURE

\[ A = \text{airfoil cross-sectional area, m}^2 \]
\[ a_\infty = \text{speed of sound, m/s} \]
\[ b = \text{wing span, m} \]
\[ C_d = \text{airfoil section drag coefficient, } d/(q_\infty c) \]
\[ C_D = \text{wing drag coefficient, } D/(q_\infty S) \]
\[ C_m = \text{airfoil section pitching moment coefficient, } M/(q_\infty c^2) \]
\[ C_{Mx} = \text{wing pitching moment coefficient, } m/(q_\infty cS) \]
\[ C_l = \text{airfoil section lift coefficient, } l/(q_\infty c) \]
\[ C_L = \text{wing lift coefficient, } L/(q_\infty S) \]
\[ c = \text{low-fidelity model output} \]
\[ c = \text{chord length, m} \]
\[ d = \text{airfoil section drag force, N} \]
\[ D = \text{wing drag force, N} \]
\[ d = \text{trust-region radius, m} \]
\[ f = \text{high-fidelity model output} \]
\[ H = \text{objective function value} \]
\[ l = \text{lower bound of } x \]
\[ l = \text{airfoil section lift force, N} \]
\[ L = \text{wing lift force, N} \]
\[ M_\infty = \text{Mach number, } V_\infty/a_\infty \]
\[ m = \text{pitching moment} \]
\[ n = \text{number of design variables} \]
\[ N_c = \text{number of coarse model evaluation} \]
\[ N_f = \text{number of fine model evaluation} \]
\( q_{\infty} \) = dynamic pressure, \( 1/2 \rho_{\infty} V_{\infty}^2 \)

\( s \) = surrogate model output

\( S \) = wing planform area, m²

\( u \) = upper bound of \( x \)

\( V_{\infty} \) = flow speed, m/s

\( V \) = wing volume, m³

\( x \) = design variables, m

\( y \) = coordinate along the wing span, m

\( z_{c/4} \) = vertical coordinates of the quarter-chord location in a wing

\( \gamma \) = wing twist, degrees

\( \alpha \) = angle of attack, degrees

\( \rho_{\infty} \) = density, kg/m³

\( \eta \) = wing span stations, \( y(b/2) \)
ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Leifur Leifsson, and my committee members, for their guidance and support throughout the course of this research.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience.

11/06/2018

Anand Amrit
ABSTRACT

The research objective of this work is to accelerate the process of multi-objective aerodynamic design exploration when using computationally intensive simulation models. The design of aerodynamic surfaces is important for modern engineered systems such as unmanned aerial systems and turbomachinery. Physics-based simulations are needed for capturing the nonlinear system behavior and nonlinear interactions between disciplines. The key challenges with using high-fidelity physics-based simulations as part of aerodynamic design include (1) the high computational cost of the simulations (ranging from few hours to days or weeks on high performance computing clusters), (2) large numbers of conflicting objectives and constraints, and design variables, and (3) the repetitive evaluations during the design exploration phase. The main contributions of this thesis are the adaptation and integration of multi-fidelity methods and Pareto set identification techniques to rapidly determine the best possible trade-offs of the aerodynamic characteristics. The proposed multi-fidelity aerodynamic Pareto set identification techniques use sequential domain patching and point-by-point exploration. The algorithms are validated using analytical problems and demonstrated on aerodynamic design problems involving transonic airfoils and subsonic wings. The proposed algorithms are benchmarked against a surrogate-assisted multi-objective evolutionary algorithm. It is found that approaches produce comparable Pareto fronts. Furthermore, the proposed multi-fidelity point-by-point aerodynamic MOO algorithm is over 50% more efficient than the benchmark method. The value of the proposed algorithms is more visible in cases where designers have a limited computational budget and only a few Pareto optimal points are required in the vicinity of a target design.
CHAPTER 1. INTRODUCTION

I. Motivation and Challenges

The aerodynamic design exploration process involves determining the best-possible trade-offs between multiple non-commensurable objectives [1, 2]. This process is typically handled using multi-objective optimization (MOO) techniques. The overall goal of this research work is to investigate numerical methods that enable rapid aerodynamic MOO when using computationally intensive predictive aerodynamic simulation models. Thereby, providing engineers with the ability to utilize information from high-fidelity models in the decision making process when designing aerodynamic systems. The key challenges of aerodynamic MOO with high-fidelity models are:

1. time-consuming model evaluations (ranging from several minutes to hours, days, or weeks on high performance computing clusters) [1],
2. a large number of design objectives (criteria) and parameters [1], and
3. multiple and repetitive model evaluations required by state-of-the-art design exploration techniques [1, 2].

The current state-of-the-art aerodynamic MOO relies on the weighted-sum optimization technique [3, 4]. This approach has been used widely for aerodynamic design exploration problems and involve the single-objective optimization (SOO) of an aggregate objective function. The aggregate objective function is typically constructed using a linear combination of all the objectives, with each of them multiplied by a pre-set weighting coefficient, see, for example, Refs. [5-9], casting it into a SOO problem. An improper setting of the weighting coefficients may, however, lead to the loss of important information with increasing complexity of the design problem. To avoid the issue of selecting appropriate weighting
coefficients, the epsilon-constrained method [10] can be used efficiently given that adjoint sensitivity information is available, e.g. [11, 12]. These approaches, however, only lead to one optimal solution per optimization iteration. To make a decision the designers may need multiple optimal solutions. Due to its ability to obtain several optimal solutions per algorithm run, genuine MOO, particularly non-dominated optimization [4], based on the so-called Pareto front [13] is becoming more popular in the design of complex aerodynamic problems.

Metaheuristic algorithms are one of the most widely used approaches by designers to obtain the Pareto front. Particle swarm optimization (PSO) [14], genetic algorithm (GA) [15, 16], multi-objective evolutionary algorithm (MOEA) [17, 18] are examples of metaheuristic algorithms that involve evaluations of population sizes of up to few hundreds of candidate designs and can generate the entire Pareto front in a single algorithm run. Other algorithms include the differential evolution [19], firefly algorithms [20], and cuckoo search [21]. A high computational cost is, unfortunately, a common feature of these techniques, which limits their use to aerodynamic design exploration problems of low complexity (i.e. problems with a low number of designable parameters and simulations with a low number of degrees of freedom).

Surrogate-based optimization (SBO) techniques [22-24] can alleviate the computational cost by replacing the costly simulations by their faster representations, referred to as surrogate models (or meta-models). In general, the SBO process is composed of four steps [22-24]:

1. sampling the design space,
2. acquisition of the simulation data,
3. construction of the surrogate models, and
4. identification of the candidate designs.
The process is continued until termination with new data samples allocated using prescribed infill criteria, followed by repetition of the steps (2) through (4). The most widely used surrogate models for aerodynamic design include response surface approximations [22], radial-based function models [23], and kriging interpolation [24]. Application of SBO in multi-objective case also known as surrogate-assisted MOO algorithms, has been of great interest in the past. For example, the Pareto-based Efficient Global Optimization [25] (ParEGO) uses the weighted-sum approach, the Pareto Set Pursuing (PSP) approach [26] utilizes global surrogate models for MOO, global surrogate-assisted MOO with constraints based on expected improvement of the objective functions is described in [23, 27], the Inexact Pre-Evaluation approach [28] is extended in [29] for MOEAs and locally constructed radial-bases function models, adaptive sampling and surrogate modeling are combined for MOGAs in [30], a global approximation-based MOO for robust design under interval uncertainty is described in [31], and, finally, surrogate-assisted MOO using global and local models is presented in [32]. Recent applications of surrogate-assisted MOO algorithms for the design of aerodynamic surfaces are following. Wang et al. [33] used response surface models and MOGA [15, 16] for aerodynamic MOO to maximize the pressure ratio and adiabatic efficiency of compressor rotors. MOO on the aerodynamic drag and lift forces of high speed train head shapes is performed by Zhang et al. [34] utilizing Kriging surrogate models [24]. Amrit et al. [15] and Leifsson et al. [35] performed MOO of transonic airfoil shapes with MOEAs [17, 18], Kriging surrogate models [27], and design space confinement strategies. All these techniques involve use of certain amount of high-fidelity model evaluations as training samples until the surrogate model accuracy is within an acceptable limit. This makes SBO cumbersome when the number of training samples increases.
Recently, there has been a growing interest in the use of multi-fidelity methods in design exploration [36, 37]. The multi-fidelity method is an approach that fuse information from models of varying degree of fidelity that leverage the computational speedup provided by the models of lower fidelities, and establish the required accuracy in the design optimization task using the model of higher fidelity. Low-fidelity models and model management strategies are the two main elements of multi-fidelity methods. Few examples of low-fidelity model include simplified models (such as coarsening the numerical discretization [38-40] and simplifying the governing equations [41, 42]), projection-based methods (such as proper orthogonal decomposition [43], and reduced basis method [44]), and data-fit methods (such as radial basis functions [45], kriging [46], and support vector regression [47]). Model management strategies include adaptation, fusion, and filtering. Adaptation approaches can be categorized into global methods (e.g., efficient global optimization (EGO) using global data-fit models and infill criteria based on expected improvement to balance exploitation and exploration [48]) or multi-fidelity trust-region methods using corrected low-fidelity models with the corrections classified as additive [49], multiplicative [50], comprehensive [51], or space mapping [52]) and local methods (such as SBO methods using local data-fit models [22]. Fusion approaches evaluate the low- and the high-fidelity models on a given set of samples and subsequently combine the outputs in one model. Examples of fusion methods include co-kriging [53], and Bayesian regression [54]. In filtering methods, the high-fidelity model is invoked following the evaluation of a low-fidelity filter. An example of a filtering method is the multi-fidelity stochastic collocation approach [55]. Although multi-fidelity methods are well established in the case of aerodynamic SOO, its use in multi-objective sense has not been studied extensively.
In summary, MOO of aerodynamic design problems involving computationally expensive simulations is an open area of research. Current state-of-the-art aerodynamic MOO algorithms are setup to estimate the entire Pareto front (which is cumbersome for expensive aerodynamic problems) and rely on either weighted-sum methods (which provide only one Pareto optimal solution per algorithm run) and/or evolutionary search methods (which need a large amount of model evaluations). Recent aerodynamic design exploration algorithms take advantage of global surrogate models to alleviate the computational burden [15, 17, 18, 35], but still need many model evaluations and can be impractical when dealing with the computationally intensive aerodynamic simulation models. Furthermore, in cases where there is a limited budget, a technique to obtain only a part of the Pareto is not available to obtain the best possible trade-offs at the vicinity of a target design. Multi-fidelity methods [37] are promising in reducing the computational burden, but these methods have not been utilized rigorously for multi-objective aerodynamic design exploration.

II. Research Goals and Questions

The overall research goal of this work is to fill this knowledge gap and investigate alternatives for efficient multi-fidelity multi-objective aerodynamic design exploration. To achieve this goal, the work seeks to answer the following research questions:

1. How can multi-fidelity methods and surrogate models be used to accelerate the PDE-constrained MOO to enable fast aerodynamic design exploration? What is the magnitude of the computational acceleration? Are the estimated best possible trade-offs using the multi-fidelity methods at the level of the high-fidelity model accuracy?

2. If the designer has a budget, is it possible to obtain only a part of the Pareto, or in other words, is it possible to obtain few optimal solutions in the vicinity of a target design which are non-commensurate using multi-fidelity methods?
(3) Is it possible to develop a strategy that can efficiently perform multi-fidelity MOO on computationally complex aerodynamic problems that have nonlinear design and feature spaces?

(4) Is it possible to develop a strategy that is robust such that the Pareto front estimated with multi-fidelity methods is always within a reasonable error margin irrespective of the number of MOO trials?

(5) Is it possible to evaluate the scalability of the multi-fidelity strategy or in other words, is it possible to measure the computational cost associated with the increase in design space dimensionality and simulation degrees of freedom?

The answers to these research questions will enable the creation and development of new and unique multi-objective aerodynamic design exploration methods. Ultimately, those methods can provide engineers with new capabilities for the design of aerodynamic systems using accurate simulation models and develop new technology.

**III. Research Contributions**

The research contributions of this work can be summarized as follows:

1. Adaptation of the sequential domain patching algorithm [56] for multi-fidelity aerodynamic design exploration. Starting from two end points of the Pareto, the algorithm is utilized to obtain an initial inaccurate Pareto front cheaply using low-fidelity aerodynamic models. Further refinement of the initial Pareto yields the accurate Pareto front.

2. Adaptation of the point-by-point Pareto set identification method [57] for multi-fidelity aerodynamic design exploration. Starting from a single point on the Pareto, the algorithm is utilized to obtain a part or entire Pareto front point-by-point using local
response surface approximation (RSA) model constructed using cheap low-fidelity aerodynamic models and few high-fidelity models for refinement.

(3) Validation of the proposed approaches using several analytical test cases.

(4) Demonstrations of the proposed approaches on two types of aerodynamic problems:
   a. A transonic airfoil design problem to obtain trade-offs of the characteristics.
   b. A subsonic wing design to obtain trade-offs of the characteristics.

(5) Evaluation and comparison of the proposed approaches using a fast surrogate-assisted multi-objective evolutionary algorithm (SA-MOEA) as a benchmark [15].

**A. List of Journal Articles**

The following journal articles form the contribution of this doctoral thesis:


The following is the author’s master’s thesis, and a journal article published based on it (these works are used in this work for benchmarking the proposed approaches):


**B. List of Conference Proceedings**

The following is a list of the conference proceedings that the author has contributed to during his doctoral work:


AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference,


C. List of Conference Talks

The following is a list of the conference talks given by the author during his doctoral work:


IV. Thesis Outline

The outline of this thesis is as follows. The next three chapters contain detailed descriptions of the proposed multi-fidelity aerodynamic MOO methodologies in the form of journal articles which are published or submitted. Chapter 2 is an article submitted to ‘Aerospace Science & Technology’ that describes the adaptation of sequential domain patching algorithm to solve few analytical problems and a two-dimensional transonic airfoil design. Chapter 3 is an article published in ‘Aerospace Science & Technology’ that describes the adaptation of point-by-point algorithm to solve few analytical problems and a two-dimensional transonic airfoil design. Chapter 4 is an article submitted to ‘Engineering Computations’ that compares the efficiency of sequential domain patching and the point-by-point algorithm with a multi-objective evolutionary algorithm using an analytical function, a two-dimensional transonic airfoil design and a three-dimensional subsonic wing design. The discussion in Chapter 5 summarizes the main contributions of this thesis and provides suggestions for future work.

References


CHAPTER 2. FAST MULTI-OBJECTIVE AERODYNAMIC OPTIMIZATION USING SEQUENTIAL DOMAIN PATCHING AND MULTI-FIDELITY MODELS


Abstract

Exploration of design trade-offs for aerodynamic surfaces requires solving of multi-objective optimization (MOO) problems. The major bottleneck here is the time-consuming evaluations of the computational fluid dynamics (CFD) model utilized to capture the nonlinear physics involved in designing aerodynamic surfaces. This, in conjunction with a large number of simulations necessary to yield a set of designs representing the best possible trade-offs between conflicting objectives (referred to as a Pareto front), makes CFD-driven MOO very challenging. This paper presents a computationally efficient methodology aimed at expediting the MOO process for aerodynamic design problems. The extreme points of the Pareto front are obtained quickly using single-objective optimizations. Starting from these extreme points, identification of an initial set of Pareto-optimal designs is carried out using a sequential domain patching algorithm. Refinement of the Pareto front, originally obtained at the level of the low-fidelity CFD model, is carried out utilizing local response surface approximations (RSAs) and adaptive corrections. The proposed algorithm is validated using a few multi-objective analytical problems as well as an aerodynamic problem involving MOO of two dimensional transonic airfoil shapes where the figures of interest are the drag and pitching moment coefficients. A multi-fidelity model is constructed using computational fluid dynamics model and control points parameterizing the shape of the airfoil. The results demonstrate that an entire or a part of the Pareto front can be obtained at a low cost when considering up to eight design variables.

Nomenclature

\[
\begin{align*}
A & = \text{cross-sectional area, } m^2 \\
a_\infty & = \text{speed of sound, } m/s \\
C_d & = \text{drag coefficient, } dl(q_{\infty}c)
\end{align*}
\]
\( C_m \) = pitching moment coefficient, \( M/(q_\infty c^2) \)

\( C_l \) = lift coefficient, \( l/(q_\infty c) \)

\( c \) = low-fidelity model output

\( c \) = chord length, m

\( d \) = drag force, N

\( d \) = trust-region radius, m

\( f \) = high-fidelity model output

\( H \) = objective function value

\( l \) = lower bound of \( x \)

\( l \) = lift force, N

\( M_\infty \) = Mach number, \( V_\infty/\alpha_\infty \)

\( M \) = pitching moment

\( n \) = number of design variables

\( N \) = total Pareto optimal solutions

\( N_c \) = number of low-fidelity model evaluations

\( N_f \) = number of high-fidelity model evaluations

\( q_\infty \) = dynamic pressure, \( 1/2\rho_\infty V_\infty^2 \)

\( s \) = surrogate model output

\( u \) = upper bound of \( x \)

\( V_\infty \) = flow speed, m/s

\( x \) = design variables, m

\( \rho_\infty \) = density, kg/m\(^3\)

\( \delta \) = trust region radius

\( \epsilon_H \) = norm of \( H \) from last two iterations

\( \epsilon_\delta \) = norm of \( \delta \) from last two iterations

\( \epsilon_x \) = norm of \( x \) from last two iterations
I. Introduction

Design of modern engineering systems often involves the use of accurate physics-based computational models. The fidelity of the simulations, e.g., in terms of discretization density of the structure at hand, increases when there is a need for higher accuracy and capturing nonlinear physics and nonlinear interactions between system disciplines, which is often required in the case of new or unconventional systems [1]. Utilization of accurate physics-based computer simulations for design purposes can pose significant challenges due to (1) their high computational cost (often ranging from a few hours to days or even weeks on high performance computing clusters), (2) a large number of design variables, constraints, and objectives (which may be conflicting), and (3) a large number of model evaluations needed by state-of-the-art optimization techniques. In this paper, an efficient approach is proposed to determine the best possible trade-offs between conflicting objectives for design exploration with accurate simulations of the flow past aerodynamic surfaces, such as the wings of an aircraft or the rotor blades of helicopters and wind turbines.

Design exploration of aerodynamic surfaces with multiple conflicting objectives can be performed by (1) parametric search guided by expert knowledge, (2) minimization of the aggregate objective function using single-objective optimization (SOO) algorithms, or (3) simultaneous minimization of multiple objectives using multi-objective optimization (MOO) routines. The first usually fails to yield optimal designs, however, it is frequently used to bypass difficulties pertinent to rigorous optimization. In the SOO approach, the aggregated objective function is typically formed by a linear combination of all objectives with the weighting coefficients set to express the designer preferences, see, e.g., [2-6]. If the objective
and constraint gradients can be computed using adjoint sensitivity information [8], the SOO problem can be solved efficiently using the epsilon-constrained method [7]. The disadvantage is that only one Pareto-optimal design is found per SOO algorithm run and the location of this design with respect to the overall Pareto front is unknown. Furthermore, the method is unable to identify non-convex portions of the front.

In the MOO approach, the goal is to obtain designs representing the best possible trade-offs between conflicting objectives. It is typically found in the form of a Pareto set (a discrete representation of the Pareto front) [4]. Population-based metaheuristics are the most popular solution approaches [9-15]. The prominent examples include evolutionary algorithms (EAs) [9], such as multi-objective genetic algorithms (MOGAs) [10] and MOEAs [11]. Other multi-objective metaheuristic approaches include particle swarm optimization [12], differential evolution [13], firefly [14] and cuckoo search [15]. Unlike the SOO approach, these population based techniques are capable of generating the entire Pareto set in a single algorithm run. Unfortunately, metaheuristics require numerous model evaluations, which limits their use to aerodynamic design exploration problems of low complexity (i.e., problems with a low number of designable parameters and simulations with a low number of degrees of freedom).

Surrogate-based optimization (SBO) [16, 17] techniques have recently become popular as a means of addressing the high computational cost of the optimization process. The main steps of the SBO procedure are: (1) sampling the design space using a design of experiments technique of choice, e.g., [18, 19], (2) acquiring the training data through high-fidelity model simulations, (3) constructing the surrogate model using the observations, and (4) updating the surrogate through allocation of additional samples. Popular surrogate modeling approaches include response surface approximations [16], radial-based function models [17], and kriging
interpolation [17]. New samples (so-called infill points) are assigned using appropriate infill criteria [17] which may involve identification of approximated optimal designs (such as Pareto-optimal solutions in case of MOO).

The SBO process plays a central role in surrogate-assisted MOO algorithms. Examples of such algorithms include Pareto-based Efficient Global Optimization [20] (ParEGO) which uses the weighted-sum approach, the Pareto Set Pursuing (PSP) approach [21] which utilizes global surrogate models for MOO, a global surrogate-assisted MOO with constraints based on expected improvement of the objective functions is described in [22], the Inexact Pre-Evaluation approach [23] is extended in [24] for MOEAs and locally constructed radial-basis function models, adaptive sampling and surrogate modeling are combined for MOGAs in [25], a global approximation-based MOO for robust design under interval uncertainty is described in [26], and, finally, surrogate-assisted MOO using global and local models is introduced in [27].

Recent applications of surrogate-assisted MOO algorithms for aerodynamic shape optimization involve various combinations of metaheuristics and surrogate modeling methods. Zhang et al. [28] utilized Kriging surrogate models [17], sampled on high-fidelity CFD models with Latin hypercube sampling [18], and the NSGA-II algorithm [10] to perform MOO on the aerodynamic forces of the high-speed train head shapes. Wang et al. [29] performed an aerodynamic MOO to maximize the pressure ratio and adiabatic efficiency of compressor rotors using response surface models and MOGA [10]. Debbie et al. [30] performed aerodynamic MOO design of helicopter blades with MOGA using Gaussian process regression techniques [17] to construct the surrogate models. Amrit et al. [11] and Leifsson et al. [31, 32] utilized MOEAs, Kriging surrogate models, and design space confinement strategies to
perform MOO of transonic airfoil shapes. Fincham et al. [33] used MOGA and radial-basis functions models [16, 17] to represent aerodynamic surfaces and performed a multi-objective aerodynamic shape optimization of camber morphing airfoil shapes.

Multi-fidelity methods [17, 34] utilize information from models of varying degree of fidelity to leverage the computational speedup of the low-fidelity models and the accuracy of the high-fidelity ones. A typical approach is to utilize the fast low-fidelity models to accelerate the design optimization process and yield initial approximations of the optimum designs, which is followed by (usually) iterative references to the high-fidelity models (through various model management strategies) aimed at refinement of the solution accuracy.

Low-fidelity modeling approaches include simplified modeling methods (e.g., simplified governing equations [35], and coarse discretization [36]), projection-based methods (e.g., proper orthogonal decomposition [37], and reduced basis method [38]), and data-fit methods (e.g., radial basis functions [16], kriging [17], and support vector regression [39]).

Model management strategies include adaptation, fusion, and filtering. Adaptation approaches can be divided into global methods (e.g., efficient global optimization (EGO) using global data-fit models and infill criteria based on expected improvement to balance exploitation and exploration [17]) or local methods (e.g., SBO methods using local data-fit models [17], and multi-fidelity trust-region methods using corrected low-fidelity models with the corrections classified as additive [40], multiplicative [41], comprehensive [42], or space mapping [43]). Fusion approaches evaluate the low- and the high-fidelity models on a given set of samples and subsequently combine the outputs in one model, (e.g., co-kriging [44], and Bayesian regression [45]). In filtering methods, the high-fidelity model is invoked following
the evaluation of a low-fidelity filter, (e.g., the multi-fidelity stochastic collocation approach [46]).

So far, multi-fidelity methods have been applied successfully to single-objective aerodynamic design problems (see, e.g., [35, 36, 41]). In this paper, an efficient multi-fidelity framework is proposed for aerodynamic design exploration in a multi-objective context. Our algorithm determines the entire Pareto, or a part of it depending upon the choice and computational budget of the designer. Two single-objective optimal designs are determined quickly using a multi-fidelity trust-region optimization algorithm with a low-fidelity model derived by a simplified modeling method [36] and adaptation using output space mapping [43]. The Pareto optimal solutions spanning between the two SOO points are obtained by relocating sequentially from one end of the front to another followed by a refinement process. For this task, the sequential domain patching technique is adopted here from the work by Koziel and Bekasiewicz [47]. In [47], the SBO is performed using a local data-fit model [16], which is constructed based on low-fidelity model evaluations that are sparsely-sampled in the vicinity of the current point on the Pareto front, and subsequently corrected using a single high-fidelity model evaluation and an additive correction [40, 43].

The distinct features of the novel aerodynamic design exploration approach presented in this paper include the following: (1) two Pareto-optimal points spanning the portion of the Pareto front to be explored are identified at low cost using multi-fidelity methods and SOO, (2) depending on the computational budget, either the entire Pareto or a part of it can be explored, (3) the proposed approach does not use metaheuristic algorithms (such as MOGAs [10] and MOEAs [11]), and (4) gradient information is not used to determine the Pareto front.
The paper is organized as follows. The next section describes the background and the proposed aerodynamic design exploration algorithm. In the following section, the proposed approach is characterized using analytical problems and the aerodynamic design exploration of transonic airfoil shapes in viscous flow. The paper ends with conclusion and remarks on future work.

II. Methodology

This section describes design exploration using sequential domain patching for Pareto set identification, and gives the details of the multi-objective optimization framework as well as multi-fidelity modeling.

A. Definition of the Pareto Front

Here, the concept of Pareto front is explained using a specific example of an aerodynamic design problem. The goal is to find a trade-off between various aerodynamic forces such as lift, drag, and pitching moment coefficients, denoted as $C_{l,f}$, $C_{d,f}$, and $C_{m,f}$, respectively. Let an accurate high-fidelity aerodynamics simulation model be denoted as $f(x) = [C_{l,f}(x) C_{d,f}(x) C_{m,f}(x)]^T$, where $x$ is the $n \times 1$ vector of design variables.

Let $F_k(x)$, $k = 1, \ldots, N_{obj}$, be a $k$th design objective of interest. If $N_{obj} > 1$ then any two designs $x^{(1)}$ and $x^{(2)}$ for which $F_k(x^{(1)}) < F_k(x^{(2)})$ and $F_l(x^{(2)}) < F_l(x^{(1)})$ for at least one pair $k \neq l$, are not commensurable, i.e., none is better than the other in the multi-objective sense. We define a Pareto dominance relation $\prec$, saying that for the two designs $x$ and $y$, we have $x \prec y$ ($x$ dominates $y$) if $F_k(x) \leq F_k(y)$ for all $k = 1, \ldots, N_{obj}$, and $F_k(x) < F_k(y)$ for at least one $k$ [48].

The goal of the multi-objective optimization is to find a representation of a so-called Pareto front (of Pareto-optimal set) $X_P$ of the design space $X$, such that for any $x \in X_P$, there is no $y \in X$ for which $y \prec x$. 

B. Pareto Front Exploration Using Sequential Domain Patching

The proposed Pareto front exploration approach is based on the sequential domain patching (SDP) algorithm proposed by Koziel and Bekasiewicz [47], and is adopted and applied in this work for multi-fidelity aerodynamic design. The approach proposed in this paper is formulated in terms of two scalar design objectives, $F_1$ and $F_2$ and produces a sequence of designs $x^{(k)*}$, $k = 1, 2, \ldots, N$, where $x^{(1)*}$ and $x^{(N)*}$ are the two ends of the Pareto front to be explored and $N$ is the total number of Pareto optimal points. In order to obtain the entire Pareto set, initially, two points representing the extreme Pareto-optimal solutions are obtained by minimizing individual objectives, one at a time, as shown in Fig. 2-1(a). If, instead of the entire front, only its part needs to be explored, then two SOOs are carried out for one of the objective functions while subjecting the second objective function into a nonlinear constraint to obtain two target points on the front, as indicated in Fig. 2-1 (b). An alternative method is a weighted-sum approach [6] that can be used to obtain two points on the Pareto front per the designer preferences encoded in the weighting factors.

Once the extreme ends of the Pareto front to be explored are obtained, the SDP-based MOO algorithm is executed, as explained in Section II.C, to obtain an initial Pareto set (see Fig. 2-2).

![Figure 2-1](image-url). Exploration of (a) the entire Pareto front and (b) only part of the Pareto front.
Figure 2-2. Pictorial representation of the sequential domain patching method: (a) patches in the design space (b) refinement of the Pareto front in the objective space.

The optimal solutions in the initial Pareto are explored within the patches constructed in the vicinity of the starting points as shown in Fig. 2-2(a). The procedure continues until the entire distance between $\mathbf{x}^{(1)*}$ and $\mathbf{x}^{(N)*}$ has been traversed. Due to the high cost of high-fidelity the model involved in the multi-objective aerodynamic problem, the algorithm is designed so as to obtain the initial Pareto set at the level of an auxiliary low-fidelity model. Subsequently, refinement of the initial Pareto set is performed using a limited number of high-fidelity model evaluations and response surface approximation (RSA) models to leverage the accuracy of the final Pareto set (Fig. 2-2 (a)). This process is explained in detail in Sections II.C, D and E.

C. Multi-Fidelity, Multi-Objective Aerodynamic Sequential Domain Patching Algorithm

This section describes the proposed SDP-based aerodynamic MOO algorithm in detail. The first extreme end of the Pareto front is a solution to the SOO problem of the form

$$\mathbf{x}^{(1)*} = \arg \min_{\mathbf{x}} F_{1}(\mathbf{x}),$$

subject to
\[ g(x) \leq 0, \]

where \( g(x) \) stands for the inequality constraints for the problem at hand. \( x^{(N)*} \) is obtained in a similar manner by minimizing \( F_2 \) subjected to appropriate inequality constraint.

The cost of solving (1) can be high depending on the dimension of the problem and the cost of the model evaluations. To expedite the process of solving (1), a trust-region-based multi-fidelity optimization algorithm [35] is executed. The multi-fidelity model is constructed using output space mapping [49] in this work. A combination of the accurate high-fidelity model \( f \) and a model \( c \), which is of lower fidelity than \( f \) and computationally faster to evaluate, is exploited by the output space mapping. Here, the low-fidelity model \( c \) is based on coarse-discretization CFD simulations (see, e.g., [35] for a discussion on approaches for low-fidelity modeling). The output space mapping algorithm produces a sequence \( x^{(1,j)} \), \( j = 0, 1, \ldots \), of approximate solutions to (1) as [49]

\[
x^{(1,j+1)} = \arg \min_{x, \|s^{(1,j)} - x^{(1,j)}\| \leq \delta^{(j)}} F_i(s^{(1,j)}(x)), \tag{2}
\]

where \( s^{(1,j)}(x) = [c_{x}^{(1,j)}(x) \ C_{d}^{(1,j)}(x) \ C_{m}^{(1,j)}(x)]^T \) is the surrogate model at iteration \( j \). The output space mapping surrogate model is

\[
s^{(1,j)}(x) = A^{(1,j)} \circ c(x) + D^{(1,j)}, \tag{3}
\]

where \( \circ \) denotes component-wise multiplication, and the multiplicative and additive terms, \( A^{(1,j)} \) and \( D^{(1,j)} \), respectively, are calculated analytically. For the drag coefficient, \( C_d \), the terms are calculated as

\[
\begin{bmatrix}
a_{d}^{(1,j)} \\
da_{d}^{(1,j)}
\end{bmatrix}
= (C_d^T C_d)^{-1} C_d^T F_d, \tag{4}
\]

\[
C_d = \begin{bmatrix}
C_{d.c}(x^{(1,0)}) & C_{d.c}(x^{(1,1)}) & \cdots & C_{d.c}(x^{(1,j)})
\end{bmatrix}^T,
\tag{5}
\]
\[
F_d = [C_{d,f}(x^{(1,0)}) \ C_{d,f}(x^{(1,1)}) \ldots \ C_{d,f}(x^{(1,j)})]^T,
\]

where \(C_{d,c}\) and \(C_{d,f}\) represent the drag coefficient values obtained by evaluations of the low- and high-fidelity models, respectively. Similar models are constructed for \(C_m\) and \(C_l\).

Using the SOO points, the MOO algorithm for the initial Pareto front representation is executed and can be formally summarized as follows [47]:

1. Patch size \(d = [d_1 \ldots d_n]^T\) is set using the procedure of Section II.D;
2. Current points are set as \(x_{c1} = x^{(1)*}\) and \(x_{cN} = x^{(N)*}\);
3. \(n\) perturbations of the size \(d\) are evaluated around \(x_{c1}\) (towards \(x_{cN}\) only) and the one that brings the largest improvement with respect to the second objective \(F_2\) is selected.
4. The patch is relocated so that it is centered at the best perturbation selected in Step 3; \(x_{c1}\) is updated;
5. \(n\) perturbations of the size \(d\) are evaluated around \(x_{cN}\) (towards \(x_{c1}\) only) and the one that brings the largest improvement with respect to the second objective \(F_1\) is selected.
6. The patch is relocated so that it is centered at the best perturbation selected in Step 5; \(x_{cN}\) is updated;
7. If the path between \(x^{(1)*}\) and \(x^{(N)*}\) is not complete, go to Step 3;

The flowchart shown in Fig. 2-3 outlines the Pareto front exploration procedure using the SDP algorithm and multi-fidelity aerodynamic models. The major differences of the proposed algorithm and the one presented in [47] lie in (i) the use of the multi-fidelity modeling (3)-(6), due the highly nonlinear aerodynamic models, used for solving the problem (2) and obtaining the SOO points, and (ii) the exploration of designs outside the design space enclosed within the patches of the SOO points (cf. Fig. 2-2).
Figure 2-3. Sequential domain patching based aerodynamic design exploration.

The algorithm yields a set of patches, covering a part of design space that contains the initial approximation of the set of Pareto-optimal solutions. The total computational cost of the algorithm depends on \( n \) and on the total number of patches. The net cost can be computed as \((M - 1)(n - 1)\) which excludes the cost of solving equation (1), where \( M = \sum_{k=1,...,n} m_k \), and is the number of intervals in the direction \( j \). However, in practice, the cost can be lower as some perturbations may not be evaluated due to the imposed constraints. Here, we describe in detail the step-by-step procedure of obtaining the Pareto front:

**Step 1:** Before the algorithm is initialized, two SOOs are performed from some random design within the design bounds. The SOO problems are solved using the trust-region, multi-
fidelity algorithm and the space mapping model [49]. The solutions from the SOO problems are used as input to the automated domain patching algorithm explained in Section II.D to obtain the patch size $d$.

**Step 2**: Solutions of the SOO problems are used as the starting points for the algorithm as marked in Fig. 2 (a) as $x^{(1)*}$ and $x^{(N)*}$. The next points are searched for in the vicinity of these starting points while moving in either direction, i.e., from $x^{(1)*}$ to $x^{(N)*}$ or vice versa.

**Step 3**: A patch is constructed with $n$ perturbations of size $d$ around $x^{(1)*}$. Each perturbation is evaluated on the low-fidelity model to obtain objective functions and constraints values. The design that brings the largest improvement with respect to the second objective $F_N$. The search for largest improvement in $F_N$ is performed with a given condition that the designs are well within the global bounds and also they satisfy the constraints. The algorithm is designed specifically for a design space which has all feasible designs. However, to satisfy the constraints, a surrogate model similar to equation (2) can be used.

**Step 4**: The best perturbation result obtained from Step 3 is used to update Step 2, i.e., $x_{c1}$ is updated. The patch is relocated, so that the center of the patch is the updated $x_{c1}$.

**Step 5**: A patch is constructed with $n$ perturbations of size $d$ around $x^{(N)*}$. Each perturbation is evaluated on the low fidelity model to obtain objective function and constraint values. The design that brings the largest improvement with respect to the second objective $F_i$. The search for largest improvement in $F_i$ is performed with a given condition that the designs are well within the global bounds and also they satisfy the linear and non-linear constraints.

**Step 6**: The best perturbation result obtained from Step 5 is used to update Step 2, i.e., $x_{cN}$ is updated. The patch is relocated, so that the center of the patch is the updated $x_{cN}$.

**Step 7**: The steps from 2-6 are continued until the path between $x^{(1)*}$ and $x^{(N)*}$ is complete.
D. Automated Determination of Patch Sizes

Using an automated technique to determine patch size, similar to the one presented in [47], the distance between $\mathbf{x}^{(1)*}$ and $\mathbf{x}^{(N)*}$ is split into integer-valued number of intervals. The number of intervals in each direction of the design variable given by $m_k$ is assigned by the following procedure, where we use the notation $\mathbf{x}^{(1)*} = [x_1^{(1)*} \ldots x_n^{(1)*}]^T$ (similarly for $\mathbf{x}^{(N)*}$):

1. $F(c)$ is evaluated at $n$ points $\mathbf{x}_k^{(1-N)*} = [x_1^{(1)*} \ldots x_{k-1}^{(1)*} x_k^{(1)*} x_{k+1} \ldots x_n^{(1)*}]^T$, $k = 1, \ldots, n$;

   where $n$ is the design space dimensionality.

2. Calculate $E_{1,k} = ||c(\mathbf{x}_k^{(1-N)*}) - c(\mathbf{x}^{(1)*})||/||c(\mathbf{x}^{(1)*})||, k = 1, \ldots, n$;

3. $F(c)$ is evaluated at $n$ points $\mathbf{x}_k^{(N-1)*} = [x_1^{(N)*} \ldots x_{k-1}^{(N)*} x_k^{(1)*} x_{k+1}^{(N)*} \ldots x_n^{(N)*}]^T$, $k = 1, \ldots, n$;

4. Calculate $E_{N,k} = ||c(\mathbf{x}_k^{(N-1)*}) - c(\mathbf{x}^{(N)*})||/||c(\mathbf{x}^{(N)*})||, k = 1, \ldots, n$;

5. Set $E_k = (E_{1,k} + E_{N,k})/2$;

6. Normalize $E_k = E_k/\max\{E_j : j = 1, \ldots, n\}$;

7. Set $m_k = \max\{2, m_{\max}, E_k\}, k = 1, \ldots, n$;

   Varying the $k^{th}$ component of the $\mathbf{x}^{(1)*}$ towards $\mathbf{x}^{(N)*}$ gives relative response changes, $E_{1,k}$ (similarly for $E_{N,k}$). The value of $m_k$ is rounded to a nearest integer with the minimum value 2 as the default. The maximum number of intervals per geometric direction, $m_{\max}$, is a user defined parameter and can be set based on a maximum allowed relative response change $E_{\max}$ as follows: $m_{\max} = \lceil \max\{E_k : k = 1, \ldots, n\}/E_{\max} \rceil$ (calculated for unnormalized $E_k$ factors). In case we have a specific computational budget, the value of $m_{\max}$ can be adjusted as per requirement.
E. Pareto Set Refinement

The algorithm discussed in Section II.C is used to determine the initial Pareto at the level of the low-fidelity model $c$. To obtain the high-fidelity Pareto-optimal designs $x_f^{(k)}$, $k = 1, ..., N$, the following procedure is executed:

$$x_f^{(k)} \leftarrow \arg \min_{x, \quad f_2(x) \leq f_2(x_f^{(k)})} F_1(s_q(x) + [f(x_f^{(k)}) - s_q(x_f^{(k)})]).$$  \hspace{1cm} (7)

In this refinement process, the first objective is improved without degrading the second objective. The above process begins with $x_f^{(k)} = x_c^{(k)}$ as the starting point and the process is iterated until convergence. The correction term in (7) makes sure that $s_q(x_f^{(k)}) = f(x_f^{(k)})$ at the initiation of each iteration. The surrogate model, $s_q$, used in this process is a second-order polynomial approximation without the mixed terms. The approximation model is based on low-fidelity model, $c$ evaluated at $x_c^{(k)}$ and the perturbed designs within the patch surrounding $x_c^{(k)}$.

III. Numerical Examples

In this section, the proposed algorithm is demonstrated using two analytical problems, and a two-dimensional multi-objective aerodynamic design problem.

A. Analytical Problems

The analytical problems, the Fonseca and Fleming function [50] and the Zitzler–Deb–Thiele's function N. 1 (ZDT 1) [51] are used to demonstrate the application of the proposed algorithm.

1. Zitzler–Deb–Thiele's function N. 1 (ZDT 1)

The formulation of the test problem is given by

$$\min f_1 = x_1,$$ \hspace{1cm} (8)

$$\min f_2 = u(1 - \sqrt{x_1/u}).$$ \hspace{1cm} (9)
where

\[ x_i \in [0,1], i=1, ..., 30, \]

and

\[ u = 1 + \frac{9}{7} \sum_{i=2}^{30} x_i. \]

The analytical functions \( f_1 \) and \( f_2 \) are considered as the high-fidelity model \( f \). A low-fidelity model \( c \) is formulated by adding noise (\( \Delta f \)) to the analytical functions as

\[
\begin{align*}
    f_{1,c} &= f_1 + \Delta f, \\
    f_{2,c} &= f_2 + \Delta f,
\end{align*}
\]

where \( \Delta f = 0.1x_1 + 0.5. \)

Figure 2-4 shows the characteristic features of low- and high-fidelity models. The approach explained in Section II.C is executed to obtain extreme points of the Pareto using SOO and then the entire Pareto front. Figure 2-5 (a) shows the initial Pareto obtained using the low-fidelity model \( c \). Using the refinement procedure explained in Section II.C, a final Pareto front is obtained as shown in red in Figure 2-5 (b).

![Figure 2-4](image_url)

**Figure 2-4.** Representation of high and low-fidelity evaluation values of 5 random designs.
2. Fonseca and Fleming Function

The formulation of the test problem is given by

\[
\begin{align*}
\min f_1 &= 1 - \exp\left[-\sum_{i=1}^{n} \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right] \\
\min f_2 &= 1 - \exp\left[-\sum_{i=1}^{n} \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right]
\end{align*}
\]

where

\[x_i \in [-4,4], i = 1, \ldots, 8.\]

The analytical functions \(f_1\) and \(f_2\) are considered as the high fidelity accurate model \(f\). The low-fidelity model \(c\) is formulated by adding noise \((\Delta f)\) to the analytical functions in the same way as in (11). Figures 2-6 (a) and (b) show the design space and the corresponding Pareto Front, respectively, obtained by executing the MOO algorithm, indicating the efficiency of the algorithm at higher dimensions.
B. Transonic Airfoil Design

This section demonstrates the proposed algorithm for the multi-objective design optimization of an airfoil in transonic flow.

1. Problem Description

The main goal of the aerodynamic problem is to obtain the trade-offs between conflicting objectives, the drag coefficient ($C_d$) and the pitching moment coefficient ($C_m$), of the RAE 2822 at a free-stream Mach number of $M_a = 0.734$, lift coefficient ($C_l$) of 0.824, and Reynolds number ($Re$) of $6.5 \times 10^6$, subject to an area cross-sectional (A) constraint. Specifically, the conflicting objectives considered here are: drag minimization and pitching moment maximization, i.e., we have $F_1(x) = C_{df}$ and $F_2(x) = C_{mf}$, and the multi-objective constrained optimization problem can be expressed as:

$$\min_{1 \leq x \leq u} C_d, \quad \max_{1 \leq x \leq u} C_m,$$

subject to

$$C_l(x) = 0.824,$$
and \[ A(\mathbf{x}) \geq A_{\text{baseline}}, \]

where \( A_{\text{baseline}} \) is the cross-sectional area of the baseline RAE2822 airfoil.

### 2. Design Variables

The airfoil shape is controlled using the B-spline parameterization approach described in Jie et al (2016) [49]. Figure 2-7 (a) shows eight control points four on each of the top and bottom surfaces, that can move in the vertical direction. The leading and trailing edge end points of the airfoil are fixed in all directions. The \( x \)-locations of the eight control points (eight design variables) are based on a fit to the RAE 2822 as: \( \mathbf{X} = [\mathbf{X}_u; \mathbf{X}_l]^T = [0.0 \ 0.15 \ 0.45 \ 0.80; 0.0 \ 0.35 \ 0.60 \ 0.90]^T \) and the initial design variable vector is \( \mathbf{x} = [\mathbf{x}_u; \mathbf{x}_l]^T = [0.0175 \ 0.0498 \ 0.0688 \ 0.0406; -0.0291 \ -0.0679 \ -0.0384 \ 0.0054]^T \). The lower bound of \( \mathbf{x} \) is set as \( \mathbf{l} = [0.015 \ 0.015 \ 0.015 \ 0.015; -0.08 \ -0.08 \ -0.08 \ -0.01]^T \), and the upper bound is set as \( \mathbf{u} = [0.08 \ 0.08 \ 0.08 \ 0.08; -0.01 \ -0.015 \ -0.015 \ 0.01]^T \).

![Airfoil computational models](image)

Figure 2-7. Airfoil computational models: (a) airfoil shape parameterization, (b) hyperbolic C-mesh.
3. High-Fidelity Viscous Aerodynamics Model

The physics problem involves solving a viscous case using Stanford University Unstructured (SU²) [52] implicit density-based flow solver. The high-fidelity aerodynamic model (f) solves the steady compressible Reynolds-Averaged Navier-Stokes (RANS) equations with the Spalart-Allmaras turbulent model [53] using the SU² implicit density-based solver. The second-order JST scheme [54] is used to calculate the convective flux along with one multi-grid level to accelerate the solution. The turbulent variables are convected using a first-order scalar upwind method. The flow solver convergence criterion is the one that occurs first of the two: (i) the change in the drag coefficient value over the last 100 iterations is less than $10^{-5}$, or (ii) a maximum number of iterations of 20,000 is met.

The computational grid is generated using the hyperbolic C-mesh of Kinsey and Barth [55] (see Fig. 2-7(b)) with the farfield set 100 chord lengths from the airfoil surface. To have the wall $y^+$ values within reasonable values (i.e. $y^+ < 5$) around the airfoil surface, the distance from the airfoil surface to the first node is $10^{-5}c$, where $c$ is the airfoil chord length. The grid points are clustered at the trailing edge and the leading edge of the airfoil with the density controlled by the number of points in the streamwise direction, and in the direction, normal to airfoil surface. Table 2-1 gives the results of a grid convergence study using the RAE 2822 airfoil at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$. The constant lift condition is achieved by internally changing the angle of attack within the flow solver. The simulation time presented in Table 2-1 gives the overall time to compute the constant lift condition using 32 processors. Table 2-1 indicates that Mesh 4 is the finest and the most accurate. However, the difference in coefficient of drag between meshes 3 and 4 is negligible compared to the difference in overall simulation time. Hence, Mesh 3 was chosen as the high-fidelity model (f) for this work.
Table 2-1. Grid convergence study for the baseline shape.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Number of Elements</th>
<th>Lift Counts (l.c.)</th>
<th>Drag Counts (d.c.)</th>
<th>Simulation Time* (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9,836</td>
<td>0.824</td>
<td>324.6</td>
<td>3.1</td>
</tr>
<tr>
<td>2</td>
<td>38,876</td>
<td>0.824</td>
<td>221.5</td>
<td>8.8</td>
</tr>
<tr>
<td>3</td>
<td>154,556</td>
<td>0.824</td>
<td>204.8</td>
<td>34.0</td>
</tr>
<tr>
<td>4</td>
<td>616,316</td>
<td>0.824</td>
<td>203.0</td>
<td>152.6</td>
</tr>
</tbody>
</table>

* Computed on a high-performance cluster with 32 processors. Flow solution only.

4. Low-Fidelity Viscous Aerodynamics Model

The model set up for low-fidelity model is same as that of high-fidelity model (f), with the grid density being far less than that of the high-fidelity one. As shown in Table 2-1, we use Mesh 1 for the low-fidelity model c. The low-fidelity model convergence criteria are set with the following values occurring first: (i) change in the drag coefficient value over the last 100 iterations is less than $10^{-4}$, or (ii) the maximum number of iterations is set to 5,000.

5. Single-Objective Optimization Results

Two SOO problems are solved using the SM algorithm [49] is described in Section II.B and the results are shown in Table 2-2. Algorithm (1) is executed iteratively using trust-region, gradient-based optimization to obtain the ‘Initial point’ (cf. Fig. 2-1). The gradient-based search uses the sequential quadratic programming algorithm, where the original problem is solved iteratively by replacing the original objective function (and nonlinear constraints) by their respective local quadratic models (linear for constraint functions). To obtain the gradient information, finite differences are utilized on the surrogate model ($s(\mathbf{x})$) with the finite difference step size set at $10^{-5}$. Once the local optimum is reached, the design is evaluated on the high-fidelity model f and fed into the gradient-based algorithm to search for further minima and the process is continued until convergence. Optimization of the multi-fidelity model is
constrained to the vicinity of the current design defined as \( \|x - x^{(i)}\| \leq \delta^{(i)} \), with the trust region radius \( \delta^{(i)} \) adjusted adaptively using the standard trust region rules [56]. The convergence tolerances for the termination conditions are set as \( \varepsilon_x = 10^{-4} \), \( \varepsilon_H = 10^{-4} \), and \( \varepsilon_\delta = 10^{-4} \).

Due to the computational expense of the CFD models, instead of exploring the entire Pareto, i.e. in between the optimal points of the two objective functions (the extreme points), only a part of the Pareto front is explored. The algorithm is run to obtain the Pareto in between \( C_m = -0.074 \) and \( C_m = -0.11 \) (chosen here for illustration purposes). Subsequently, two SOO problems are solved to obtain the end points of the Pareto front:

**SOO problem 1:**

\[
\mathbf{x}^{(1)*} = \arg\min_{x, \|x-x^{(i)}\| \leq \delta^{(i)}} C_d (s(x))
\]  

subjected to

\[
C_l(x) = 0.824, \\
C_m(x) \geq -0.11, \\
A(x) \geq A_{baseline}.
\]

**SOO problem 2:**

\[
\mathbf{x}^{(N)*} = \arg\min_{x, \|x-x^{(i)}\| \leq \delta^{(i)}} C_d (s(x))
\]  

subjected to

\[
C_l(x) = 0.824, \\
C_m(x) \geq -0.074, \\
A(x) \geq A_{baseline}.
\]

where \( s(x) \) is a fast surrogate model as described in Section II, and \( \mathbf{x}^{(1)*} \) and \( \mathbf{x}^{(N)*} \) are the two extreme points of the Pareto front.

As can be seen in Table 2-2, for problems 1 and 2, the SM algorithm reduces the drag coefficient value from 203.80 cts to 116.9 cts and 121.8 cts (note that we define one d.c as \( \Delta C_d \).
\[ = 10^{-4}, \text{ and one lift count (l.c.) as } \Delta C_l = 10^{-2}, \text{ respectively, while satisfying the constraints.} \]

Figures 2-8(a) and (b) show comparisons of the airfoil shapes and the pressure coefficient distributions of the baseline and SOO optimal designs. Figures 2-9(a), (b) and (c) show the pressure coefficient contours of the baseline and optimum shape designs, respectively, indicating a considerable reduction of the shock strength. In terms of the number of model evaluations, SM based optimization utilized approximately 500 low-fidelity models and only 4 high-fidelity models for both the problems. The cost in terms of CPU time for the entire optimization process is approximately 14 hours on a HPC with 32 processors for each of the SOO problems.

Table 2-2. **Single objective optimization results**

<table>
<thead>
<tr>
<th>Parameter/method</th>
<th>Baseline</th>
<th>SOO Problem 1</th>
<th>SOO Problem 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_l ) (l.c.)</td>
<td>82.35</td>
<td>82.39</td>
<td>82.39</td>
</tr>
<tr>
<td>( C_d ) (d.c.)</td>
<td>203.80</td>
<td>116.9</td>
<td>121.8</td>
</tr>
<tr>
<td>( C_{m,c/4} )</td>
<td>-0.0905</td>
<td>-0.1023</td>
<td>-0.0736</td>
</tr>
<tr>
<td>( A )</td>
<td>0.0779</td>
<td>0.0779</td>
<td>0.0779</td>
</tr>
<tr>
<td>( N_c )</td>
<td>-</td>
<td>550</td>
<td>499</td>
</tr>
<tr>
<td>( N_f )</td>
<td>-</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>CPU Time (hours)</td>
<td>-</td>
<td>14</td>
<td>13.55</td>
</tr>
</tbody>
</table>
Figure 2-8. SOO results showing baseline and optimized (a) airfoil shapes, and (b) pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$.

Figure 2-9. SOO Mach contours at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$ of (a) the baseline airfoil, (b) the SOO optimal design, $x_1^*$, and (c) the SOO optimal design, $x_2^*$. 
6. The Pareto Front

The design space exploration is performed using the algorithm described in Section II. Initial designs corresponding to the best possible value of the first objective (minimum drag coefficient) subjected to two different non-linear pitching moments (target pitching moment values) are obtained in the first step of the process using the output space mapping (SM) based SOO algorithm [49] as shown in (15) and (16). Due to the nature of constraint enforced (cf. (14)), the MOO algorithm is modified to explore designs outside the design space enclosed within the two SOO points (i.e. points obtained from (15) and (16)).

Subsequent designs along the Pareto are obtained using the algorithm on the low-fidelity model as described in Section II.C and the process is terminated when the entire Pareto front is traversed in between the two initial designs. Further, refinement of the initial Pareto front is performed by evaluating the optimal solutions on high-fidelity model and minimizing the surrogate model constructed at the vicinity of each optimal solutions as discussed in section II.E. The total cost in terms of CPU time to obtain the initial end points of the Pareto, the initial Pareto and the refined Pareto is 27 hrs, 13 hrs and 10 hrs respectively, on a HPC with 32 processors. Subsequently, the net cost of obtaining the final Pareto front which contains 17 Pareto optimal solutions (as shown in Fig. 2-10 (b)) is approximately 50 hrs.

Figure 2-10 (a) shows the refined optimal solution set (Pareto front) obtained in between the SOO points. A zoomed in plot of the final Pareto front is represented in Fig. 2-10 (b). Few points (point 1 and point 2) on the Pareto optimal set were selected to be compared with the baseline design. Figures 2-11 (a) and (b) show comparisons of all the airfoil shape designs and the pressure coefficient distributions for the selected points and the baseline design. There is a significant difference in the pressure coefficient distribution of the selected points compared to the baseline with the former having a considerable reduction in shock strength. Further,
Mach contour plots in Fig. 2-12 shows point 1 with higher shock strength leading to more drag compared to point 2. This aligns with the fact that to obtain a lower drag, there will be a decrease in pitching moment as shown in Fig. 2-10. Each CFD simulation is converged to within 1 drag count and hence irregularity in the Pareto front is observed if zoomed in as in Fig. 2-10 (b).

Figure 2-10. The final refined Pareto front at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$.

Figure 2-11. MOO results showing (a) the airfoil shapes, and (b) the pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$. 
Figure 2-12. Mach contours at $M_\infty = 0.734$, $C_l = 0.824$, $Re_\infty = 6.5 \times 10^6$ of (a) point 1, (b) point 2.

**IV. Conclusion**

This paper presents a unique methodology for the design exploration of aerodynamic problems in a multi-objective form. The extreme points of the Pareto front to be explored are obtained cheaply using multi-fidelity models and single-objective optimization. Further, starting from the extreme points, an approximate Pareto front is identified by constructing patches and using fast low-fidelity aerodynamic models to search for optimum points in those patches. A refinement of the approximate Pareto front is performed using high-fidelity models to obtain the final Pareto front. The key features of the proposed algorithm that distinguishes it from other surrogate-assisted multi-objective aerodynamic design exploration methods are: (1) use of fast low-fidelity models to identify an initial Pareto front quickly, (2) the use of few high-fidelity model evaluations to refine and obtain the final accurate Pareto front, (3) objective function aggregation is not required, (4) gradient information of the objective function is not used to obtain the Pareto optimal solutions. Future work will investigate the robustness and the scalability of the proposed algorithm.
References


Aerodynamic Design Problems, 54th AIAA Aerospace Sciences Meeting, San Diego, California, USA.


CHAPTER 3. MULTI-FIDELITY AERODYNAMIC DESIGN TRADE-OFF EXPLORATION USING POINT-BY-POINT PARETO SET IDENTIFICATION


Abstract

Aerodynamic design is inherently a multi-objective optimization (MOO) problem. Determining the best possible trade-offs between conflicting aerodynamic objectives can be computationally challenging when carried out directly at the level of high-fidelity computational fluid dynamics simulations. This paper presents a computationally cheap methodology for exploration of aerodynamic design trade-offs. In particular, point-by-point identification of a set of Pareto-optimal designs is executed starting in the neighbourhood of a single-objective optimal design, and using a trust-region-based, multi-fidelity optimization algorithm as well as locally constructed response surface approximations (RSAs). In this work, the RSAs are constructed using second-order polynomials without mixed terms, multi-fidelity models, and adaptive corrections. The application of the point-by-point MOO algorithm is demonstrated through MOO of transonic airfoil shapes using the Reynolds-Averaged Navier Stokes equations and the Spalart-Allmaras turbulence model. The results demonstrate that the Pareto front in the neighbourhood of an initial design can be obtained at a low cost when considering up to 12 design variables. The results also indicate that the computational cost of the optimization process grows slowly with the number of the design variables, and the repeatability of the algorithm is very good when starting the search from different initial points.

Nomenclature

\[ A \quad = \quad \text{cross-sectional area, m}^2 \]
\[ a_\infty \quad = \quad \text{speed of sound, m/s} \]
\[ C_d \quad = \quad \text{drag coefficient, } d/(q_\infty c) \]
\[ C_m \quad = \quad \text{pitching moment coefficient, } M/(q_\infty c^2) \]
Design exploration is important for the synthesis of aerospace systems. During the design phase, evaluations of the system performance are commonly obtained by means of computational models or simulations, as well as experiments. The requirements in the design of complex aerospace systems is towards using increasingly accurate simulation models to...
capture nonlinear physics and nonlinear interactions between disciplines [1]. Aerodynamic design exploration with accurate simulation models poses computational challenges due to (1) time-consuming model evaluations (ranging from several minutes to hours, days, or weeks on high performance computing clusters), (2) a large number of design objectives (criteria) and parameters, and (3) the multiple and repetitive model evaluations required by state-of-the-art design exploration techniques. This paper presents an efficient approach to determine the best possible trade-offs between conflicting criteria for design exploration with computationally costly predictive simulations of aerodynamic surfaces, such as aircraft wings, and helicopter rotor blades.

Hands-on design exploration of aerodynamic surfaces, such as parameter sweeps guided by engineering experience, is still widely practiced. On the other hand, there has been considerable research on automated design exploration through numerical optimization techniques. Automated aerodynamic design exploration is typically handled through constrained single-objective optimization (SOO). This approach has been intensely studied over the last decades, see, e.g., Refs. [2-6]. Nowadays, the most commonly used approaches include gradient-based search techniques with the gradients determined efficiently by adjoint methods (e.g., [7-17]). Unfortunately, constrained SOO only yields information with regards to one optimal design in terms of a single objective function with a set of constraints imposed on other objectives.

The purpose of multi-objective optimization (MOO) is to determine the best possible trade-offs between multiple non-commensurable objectives [18]. In the area of aerodynamic design exploration, a widely used MOO approach is the weighted-sum method [19] where an aggregate objective function is formed by a linear combination of all the objectives, with each
of them multiplied by a pre-set weighting coefficient [20-24]. Thus, the MOO problem is cast as a SOO problem that can be solved using the optimization techniques suitable for such problems. If adjoint sensitivity information is available, the epsilon-constrained method [25] can be used efficiently for the task, see, e.g., Refs. [26,27]. These approaches still only yield one optimal design with respect to the aggregate objective function, which depends heavily on the choice of the weighting coefficients.

The set of all designs representing the best possible trade-offs between the conflicting objectives, a so-called Pareto front [17], can be determined using a class of methods called evolutionary algorithms (EAs) [28,29]. Examples of such approaches include multi-objective genetic algorithms (MOGAs) [30,31] and MOEAs [32,33]. These population-based metaheuristics determine the Pareto front through procedures that mimic biological phenomena or social behavior, and generally requires hundreds or thousands of objective function evaluations to converge. Nowadays, virtually all metaheuristic algorithms have their multi-objective versions, e.g., particle swarm optimizers [34], differential evolution [35], as well as more recent methods, such as the firefly algorithms [36] or cuckoo search [37]. High computational cost is unfortunately a common feature of all of these techniques, which limits their use to aerodynamic design exploration problems of low complexity (i.e., problems with a low number of designable parameters and simulations with a low number of degrees of freedom).

Surrogate-based optimization (SBO) techniques [38-40] are widely used in engineering design to replace computationally costly simulations by their faster representations, referred to as surrogate models (or metamodels). In general, the SBO process is composed of four major steps: (1) sampling the design space, (2) acquisition of the simulation data, (3) construction of
the surrogate models, and (4) identification of the candidate designs. The process is continued until termination with new data samples allocated using prescribed infill criteria, followed by repetition of the steps (2) through (4). Various design of experiment techniques [41] are typically used in Step (1). The most widely used surrogate models for aerodynamic design include response surface approximations [38], radial-based function models [39], and kriging interpolation [39]. Commonly used infill criteria in aerodynamic design balance exploration and exploitation [39].

SBO is fairly well established in the single-objective case and has proven to successfully deal with the computational burden to a certain degree in aerodynamic design, see, for example, Refs. [42-46]. In the multi-objective case, there has been a growing interest in using surrogate-assisted (also called approximation-assisted or metamodel-assisted) MOO algorithms. For example, the Pareto-based Efficient Global Optimization [47] (ParEGO) uses the weighted-sum approach, the Pareto Set Pursuing (PSP) approach [48] utilizes global surrogate models for MOO, global surrogate-assisted MOO with constraints based on expected improvement of the objective functions is described in [39,49], the Inexact Pre-Evaluation approach [50] is extended in [51] for MOEAs and locally constructed radial-bases function models, adaptive sampling and surrogate modeling are combined for MOGAs in [52], a global approximation-based MOO for robust design under interval uncertainty is described in [53], and, finally, surrogate-assisted MOO using global and local models is presented in [54].

A few examples of recent applications of surrogate-assisted MOO algorithms for the design of aerodynamic surfaces are the following. Zhang et al. [55] utilize Kriging surrogate models [39], sampled on high-fidelity CFD models with Latin Hypercube Sampling [41], and the NSGA-II algorithm [30,31] to perform MOO on the aerodynamic drag and lift forces of high
speed train head shapes. Wang et al. [56] used response surface models and MOGA [30,31] for aerodynamic MOO to maximize the pressure ratio and adiabatic efficiency of compressor rotors. Debbie et al. [57] used Gaussian process regression techniques [39] to construct surrogate models and perform aerodynamic MOO design of helicopter blades with MOGA. Amrit et al. [32] and Leifsson et al. [58] performed MOO of transonic airfoil shapes with MOEAs, Kriging surrogate models, and design space confinement strategies. Fincham et al. [59] showed a multi-objective aerodynamic shape optimization of camber morphing airfoils using radial basis functions models [38,39] to represent aerodynamic surfaces and MOGA.

Multi-fidelity methods [40,60,61] are approaches to fuse information from models of varying degree of fidelity. Multi-fidelity methods leverage the computational speedup provided by the models of lower fidelities, and establish the required accuracy in the design optimization task using the model of high fidelity. The main elements of multi-fidelity methods are low-fidelity models and model management strategies. The former provides an approximation of the high-fidelity model at a low computational cost, whereas the latter selects the model to evaluate and guarantees that the accuracy of the multi-fidelity models matches the level of the high-fidelity one. Low-fidelity modeling approaches include simplified models (such as simplifying the governing equations [62-65], and coarsening the numerical discretization [64-66]), projection-based methods (such as proper orthogonal decomposition [67,68], and reduced basis method [69,70]), and data-fit methods (such as radial basis functions [38,71], kriging [39,72,73], and support vector regression [74]). Model management strategies include adaptation, fusion, and filtering. Adaptation approaches can be categorized into global methods (e.g., efficient global optimization (EGO) using global data-fit models and infill criteria based on expected improvement to balance exploitation and exploration [39,49,75]) or
local methods (such as SBO methods using local data-fit models [38,39,40,76,77], and multi-fidelity trust-region methods using corrected low-fidelity models with the corrections classified as additive [78-80], multiplicative [81-83], comprehensive [84,85], or space mapping [86-88]). Fusion approaches evaluate the low- and the high-fidelity models on a given set of samples and subsequently combine the outputs in one model. Examples of fusion methods include co-kriging [89,90], and Bayesian regression [91]. In filtering methods, the high-fidelity model is invoked following the evaluation of a low-fidelity filter. An example of a filtering method is the multi-fidelity stochastic collocation approach [92].

In the case of aerodynamic SOO, multi-fidelity methods are well-established for deterministic design applications (see, for example, [63-65,79,85,89]), and have been identified as being critical in alleviating the computational burden for stochastic applications [61]. However, in the case of aerodynamic MOO, multi-fidelity methods have not been studied extensively.

In this paper, a novel aerodynamic design exploration algorithm is proposed that determines the Pareto front in the vicinity of a single-objective optimal design. In our approach, the single-objective optimal design is determined quickly and accurately using a multi-fidelity trust-region optimization algorithm with a low-fidelity model derived by a simplified modeling method [65] and an adaptation using output space mapping [86,87]. Next, the trade-off designs in the vicinity of the single-objective optimal design are obtained by moving along the Pareto front point-by-point, and identifying the subsequent Pareto-optimal solutions using SBO. The point-by-point Pareto set exploration is adopted here from the work by Koziel and Bekasiewicz [93] in the area of microwave antenna multi-objective design. In this work, the SBO is performed using a local data-fit model [38], which is constructed based on low-fidelity model
evaluations that are sparsely-sampled in the vicinity of the current point on the Pareto front, and subsequently corrected using a single high-fidelity model evaluation and an additive correction [78,86].

The distinct characteristics of the proposed approach compared to the related surrogate-assisted MOO methods utilized for aerodynamic design exploration, such as those used in Refs. [20-24, 26, 27, 32, 55-59], are listed as follows. (1) The proposed technique utilizes SOO with multi-fidelity methods to quickly locate one point on the Pareto front. (2) Based on the available computational budget, the Pareto front can be determined in the vicinity of the first point using the point-by-point Pareto set exploration [93]. (3) The technique does not rely on metaheuristic algorithms (such as MOGAs [30,31] and MOEAs [32,33]). (4) Furthermore, design space confinement in any form is not required. (5) Objective function aggregation is not required. (6) Objective function gradient information is not required to determine the Pareto front.

The paper is organized as follows. The next section describes the background and the proposed aerodynamic design exploration algorithm. In the following section, the proposed approach is characterized using analytical problems and the aerodynamic design exploration of transonic airfoil shapes in viscous flow. The paper ends with conclusion and remarks on future work.

II. Methodology

This section describes the formulation of the Pareto front, local exploration of the front using point-by-point Pareto set identification, and gives the details of the point-by-point multi-objective optimization algorithm and the multi-fidelity modeling.
A. Definition of the Pareto Front

Let \( \mathbf{x} \) be the \( n \times 1 \) vector of \( n \) design variables. Let \( \mathbf{f}(\mathbf{x}) = [C_{l.f}(\mathbf{x}) \ C_{d.f}(\mathbf{x}) \ C_{m.f}(\mathbf{x})]^T \) be the attributes of the airfoil obtained by an accurate high-fidelity aerodynamics simulation model. Here, the scalars \( C_{l.f}, C_{d.f}, \) and \( C_{m.f} \) are the lift, drag, and pitching moment coefficients, respectively. Let \( F_k(\mathbf{x}), k = 1, \ldots, N_{\text{obj}}, \) be a \( k \)th design objective of interest. A typical performance objective would be to minimize the drag coefficient, in which case \( F_k(\mathbf{x}) = C_{d.f} \). Another objective would be to minimize the pitching moment coefficient, i.e., \( F_k(\mathbf{x}) = C_{m.f} \).

If \( N_{\text{obj}} > 1 \) then any two designs \( \mathbf{x}^{(1)} \) and \( \mathbf{x}^{(2)} \) for which \( F_k(\mathbf{x}^{(1)}) < F_k(\mathbf{x}^{(2)}) \) and \( F_l(\mathbf{x}^{(2)}) < F_l(\mathbf{x}^{(1)}) \) for at least one pair \( k \neq l \), are not commensurable, i.e., none is better than the other in the multi-objective sense. We define Pareto dominance relation \( \prec \), saying that for the two designs \( \mathbf{x} \) and \( \mathbf{y} \), we have \( \mathbf{x} \prec \mathbf{y} \) (\( \mathbf{x} \) dominates \( \mathbf{y} \)) if \( F_k(\mathbf{x}) \leq F_k(\mathbf{y}) \) for all \( k = 1, \ldots, N_{\text{obj}}, \) and \( F_k(\mathbf{x}) < F_k(\mathbf{y}) \) for at least one \( k \) [94]. The goal of the multi-objective optimization is to find a representation of a so-called Pareto front (of Pareto-optimal set) \( \mathbf{X}_P \) of the design space \( \mathbf{X} \), such that for any \( \mathbf{x} \in \mathbf{X}_P \), there is no \( \mathbf{y} \in \mathbf{X} \) for which \( \mathbf{y} \prec \mathbf{x} \).

B. Point-by-Point Exploration of the Pareto Front

The point-by-point Pareto front exploration approach is illustrated in terms of two scalar design objectives, \( F_1 \) and \( F_2 \), in Figs. 3-1 and 3-2. Initially, a point on the Pareto front is obtained to be used as the starting point for Pareto front exploration (Fig. 3-1). In this work, a single-objective optimization (SOO) is carried out for one of the objective function while either subjecting the second objective function into a nonlinear constraint to get a point, somewhere on the Pareto or at the extreme ends of the Pareto, if the SOO is performed, without subjecting the second objective function into any constraint. Alternatively, for example, the weighted sum method [23,24] can be used to obtain the initial point on the Pareto front. As indicated on Fig.
3-2, an optimal point can be obtained anywhere on the Pareto front using SOO. Successive Pareto optimal points can be obtained by moving upwards or downwards the SOO point as shown in Fig. 3-2 in red arrows.

Starting from the initial point \( x^{(1)} \), the search for the Pareto front in its vicinity is conducted as follows (cf. Fig. 3-1). A local patch is formed around the initial point. In this patch, the target point on the Pareto front is searched using fast multi-fidelity models. The search progresses iteratively until the target point is found. This step may require the setup of a few patches, but the search relies heavily on the fast multi-fidelity models and can be executed at low computational cost. In this work, a local response surface approximation is constructed using fast low-fidelity models and corrected using one high-fidelity model evaluation. From the next point on the Pareto front, the procedure is continued until the number of requested points on the front are found, or the search is stopped once a given computational budget is exhausted. Figure 3-3 gives the flowchart of the proposed aerodynamic design exploration algorithm.

![Flowchart of the proposed aerodynamic design exploration algorithm](image)

**Figure 3-1.** Hypothetical representation of the point-by-point Pareto front exploration progression shown for (a) two design variables in the design and (b) two objectives in the feature space.
C. Multi-Fidelity, Point-by-Point Multi-Objective Optimization Algorithm

The MOO algorithm produces a sequence of designs $\mathbf{x}^{(k)}$, $k = 1, 2, \ldots$, on Pareto front. The first point, $\mathbf{x}^{(1)}$, can be determined by any means, such as SOO with the objective constructed using weighted sum method [19], or by SOO of one of the objective with constraints on the
other objectives. In this work, we use the latter approach. In particular, \( x^{(1)} \) is a solution to the SOO problem of the form

\[
x^{(1)} = \arg \min_x F_1( f(x)), \tag{1}
\]

subjected to

\[
F_2(f(x)) - b \leq 0,
\]

where \( b \) is the threshold value for the second objective, and \( f(x) \) is the output of the accurate high-fidelity simulation model. Thus, an optimum design value of the first objective is obtained which lies on the Pareto front.

To accelerate the process of solving (1), a trust-region-based multi-fidelity optimization algorithm [40, 60-63] is used. The multi-fidelity model is constructed using output space mapping [95] in this work. In particular, the output space mapping approach exploits a combination of the accurate high-fidelity model \( f \), and a model \( c \), which is of lower fidelity than \( f \), but is computationally faster to evaluate. Here, the low-fidelity model \( c \) is based on coarse-discretization CFD simulations (see, e.g., Refs. [40, 60-63] for a discussion on approaches for low-fidelity modeling). The output space mapping algorithm produces a sequence \( x^{(1,j)}, j = 0, 1, \ldots \), of approximate solutions to (1) as [95]

\[
x^{(1,j+1)} = \arg \min_{x, \|x^{(1,j)} - x^{(1,j+1)}\| \leq \delta^{(j)}} F_1(s^{(1,j)}(x)), \tag{2}
\]

where \( s^{(1,j)}(x) = [C_{l,s}^{(1,j)}(x) \ C_{d,s}^{(1,j)}(x) \ C_{m,s}^{(1,j)}(x)]^T \) is the surrogate model at iteration \( j \). The output space mapping surrogate model is

\[
s^{(1,j)}(x) = A^{(1,j)} \circ c(x) + D^{(1,j)}, \tag{3}
\]
where \( \circ \) denoted component-wise multiplication, and the multiplicative and additive terms, \( \mathbf{A}^{(1,j)} \) and \( \mathbf{D}^{(1,j)} \), respectively, are calculated analytically. For the drag coefficient \( C_d \), the terms are calculated as

\[
\begin{bmatrix}
0_d^{(1,j)} \\
\mathbf{d}_d^{(1,j)}
\end{bmatrix} = (\mathbf{C}_d^T \mathbf{C}_d)^{-1} \mathbf{C}_d^T \mathbf{F}_d,
\]

(4)

\[
\mathbf{C}_d = \begin{bmatrix}
\mathbf{C}_{d,c}(\mathbf{x}^{(1,0)}) & \mathbf{C}_{d,c}(\mathbf{x}^{(1,1)}) & \cdots & \mathbf{C}_{d,c}(\mathbf{x}^{(1,j)}) \\
1 & 1 & \cdots & 1
\end{bmatrix}^T,
\]

(5)

\[
\mathbf{F}_d = \begin{bmatrix}
\mathbf{C}_{d,f}(\mathbf{x}^{(1,0)}) & \mathbf{C}_{d,f}(\mathbf{x}^{(1,1)}) & \cdots & \mathbf{C}_{d,f}(\mathbf{x}^{(1,j)})
\end{bmatrix}^T,
\]

(6)

where \( \mathbf{C}_{d,c} \) and \( \mathbf{C}_{d,f} \) represent the drag coefficient values obtained by evaluations of the low- and high-fidelity models, respectively. Similar models are constructed for \( \mathbf{C}_m \) and \( \mathbf{C}_m \).

The next designs that are determined along the Pareto front, \( \mathbf{x}^{(k)} \), \( k = 2, 3, \ldots \), are obtained by exploring it point-by-point \cite{93}. The optimal design from solution (1) is utilized as a starting point and a constrained SOO is performed to get the next point in the Pareto. Let \( F_2^{(k)} \) be the threshold value for the second objective, then we have

\[
\mathbf{x}^{(k)} = \arg \min_{\mathbf{x}, F_2(f(\mathbf{x})) \leq F_2^{(k)}} F_1(f(\mathbf{x})).
\]

(7)

Here, \( \mathbf{x}^{(k)} \), \( k = 2, 3, \ldots \), is the \( k \)th element of the Pareto set and the process is continued until the design specifications are met. Pareto optimal points can be obtained in both directions along the Pareto, i.e., above and below the optimal design from solution (1) as shown in Fig. 3-3.

The search for the point \( \mathbf{x}^{(k)} \) close to the design \( \mathbf{x}^{(k-1)} \) is performed using surrogate-based optimization as follows. Specifically, \( \mathbf{x}^{(k)} \) is obtained iteratively as a sequence of solutions to (7) \( \mathbf{x}^{(k,j)} \), \( j = 0, 1, \ldots \), with \( \mathbf{x}^{(k,0)} = \mathbf{x}^{(k-1)} \), as follows

\[
\mathbf{x}^{(k,j+1)} = \arg \min_{\mathbf{x}, F_2(s(\mathbf{x})) \leq F_2^{(k)}} F_1(s^{(k,j)}(\mathbf{x})),
\]

(8)

where the surrogate \( s^{(k,j)} \) is defined as
where $s_q^{(k,j)}$ is a local response surface approximation (RSA) model of $c$, constructed in the vicinity of current design $x^{(k,j)}$. The vicinity is defined here by the patch (or interval range) $[x^{(k,j)} - \delta, x^{(k,j)} + \delta]$, where $\delta$ is the trust region radius [6]. The RSA model is described here below.

The multi-objective design optimization algorithm is summarized as follows:

1. Obtain $x^{(1)}$ by performing single-objective surrogate-based optimization;
2. Set $F_2^{(1)} = F_2(f(x^{(1)}))$ and set $k = 2$;
3. Set $x^{(k,0)} = x^{(k-1)}$;
4. Evaluate $f^{(k,j)}(x^{(k,j)})$;
5. Construct local RSA model $s_q^{(k,j)}(x)$ within the interval defined as $[x^{(k,j)} - \delta, x^{(k,j)} + \delta]$;
6. Use Eqn. (8) to obtain $x^{(k,j+1)}$ by optimizing the surrogate model $s_q^{(k,j)}(x)$;
7. If $j = 2$, set $x^{(k)} = x^{(k,j+1)}$ and go to Step 8; otherwise update the index $j = j + 1$ and go to Step 4;
8. If the termination condition is met, terminate the algorithm; otherwise update the index $k = k+1$, set the threshold $F_2^{(k)}$ and go to Step 3.

In Step 5, a local patch is formed encompassing the starting point obtained from previous step as shown in Fig. 3-1. To determine the patch dimension, given by $[x^{(k,j)} - \delta, x^{(k,j)} + \delta]$, a sensitivity analysis is performed once by perturbing the starting design randomly. Normally, $\delta$ is set at 0.5-1% of the difference of upper and lower bounds of the design space.

Furthermore, in Step 5, the RSA is constructed within the local patch using a quadratic model without mixed terms given by

$$s_q(x) = a_0 + \sum_{k=1}^n a_k x_k + \sum_{k=1}^n a_{n+k} x_k^2,$$  \(10\)
where \( a_k, k = 0, 1, 2, \ldots, 2n \), are the coefficients of the quadratic RSA, and \( x_k, k = 1, \ldots, n \) is the \( k \)th component of the variable vector \( x \). In this work, the surrogate \( s_q \) is setup using a star distribution with \( 2n + 1 \) samples (see, e.g., Koziel et al. [40]) evaluated by the low-fidelity model \( (e) \). Subsequently, the surrogate model, \( s \), is constructed using the RSA and one high-fidelity model evaluation (cf. (9)).

The minimization in Step 6 is performed using a gradient-based search algorithm with the gradients obtained by finite differentiation of the surrogate model (cf. (8)). The target values of the second objective function, \( F_2^{(k)} \), is set a-priori by the designer, and the inequality constraint is handled directly. The results from gradient-based algorithm is checked for following three criteria, (i) \( \| x^{(i)} - x^{(i-1)} \| < \varepsilon_x \), (ii) \( |F_1^{(i)} - F_1^{(k)}| < \varepsilon_F \), (iii) \( \delta^{(i)} < \varepsilon_\delta \), where \( \varepsilon_x, \varepsilon_F, \) and \( \varepsilon_\delta \) are user defined convergence tolerances. In our implementation, the three criteria are combined through the logical alternative (OR). If the criteria are met, it is evaluated on the high-fidelity model, to obtain the target point. If not, the patch is updated using a trust region radius framework [6], based on which the trust-region size is reduced by setting \( \delta = \delta/m \), where \( m = 3 \), or increased by setting \( \delta = \delta \cdot m \), where \( m = 2 \).

In Step 7, the target point obtained from Step 6 is set as the starting point for the next algorithm iteration. The process is continued until the designer is satisfied (or a computational budget is exhausted), or the extreme point of the Pareto front is reached (i.e., the first objective cannot be further improved).

**III. Numerical Examples**

In this section, the proposed algorithm is demonstrated using three analytical problems (convex, concave and discontinuous Pareto fronts), and a two-dimensional multi-objective
aerodynamic design problem. Furthermore, the scalability and robustness of the algorithm is characterized.

A. Analytical Problems

The analytical problems are the Zitzler–Deb–Thiele's function N. 1 (ZDT 1) [96], Zitzler–Deb–Thiele's function N. 2 (ZDT 2) [96], and the Kursawe [96] function, which are convex, concave and discontinuous Pareto fronts, respectively.

1. Zitzler-Deb-Thiele’s function N. 1 (ZDT 1)

The formulation of the test problem is given by

\[ \text{Min } f_1 = x_1, \]

\[ \min f_2 = u(1 - \sqrt{x_1/u}), \]  

subject to \( x_i \in [0,1], i = 1, \ldots, 8, \)

where \( u = 1 + \frac{9}{7} \sum_{i=2}^{8} x_i. \)

The analytical functions \( f_1 \) and \( f_2 \) are considered as the high fidelity accurate model \( f \). A low fidelity model \( c \) is formulated by adding noise (\( \Delta f \)) to the analytical functions as shown below:

\[ f_{1,c} = f_1 + \Delta f, \]  

\[ f_{2,c} = f_2 + \Delta f, \]

where \( \Delta f = 0.1x_1 + 0.5. \)

Figure 3-4 shows the difference in objective function value for high and low-fidelity models. The approach explained in Section II is executed to obtain the starting point using SOO and then the entire Pareto front. The Pareto obtained was validated using the data points from actual Pareto as shown in Fig. 3-5.
2. Zitzler-Deb-Thiele’s function $N. 2$ (ZDT 2)

The formulation of the test problem is given by

$$\min f_1 = x_1,$$

$$\min f_2 = u(1 - (x_1/u)^2),$$

subject to

$$x_i \in [0,1], \ i = 1,...,5,$$

where

$$u = 1 + 2.25 \sum_{i=2}^{5} x_i.$$
Figure 3-6. ZDT 2 results: Comparison of Pareto obtained with the actual Pareto.

The analytical functions $f_1$ and $f_2$ are considered as the high fidelity accurate model $f$. A low fidelity model $c$ is formulated by adding noise to the analytical functions as shown in Eqns. (13), and (14). The Pareto obtained was validated using the data points from actual Pareto (cf. Fig. 3-6).

3. Kursawe function

The proposed algorithm is demonstrated to obtain a discontinuous Pareto by considering a complex problem known as Kursawe [97]. The conflicting functions as given by Kursawe [97] are

$$\begin{align*}
\min f_1 &= \sum_{i=1}^{2} [-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})], \\
\min f_2 &= \sum_{i=1}^{3} [|x_i|^{0.8} + 5 \sin(x^3)],
\end{align*}$$

subject to $x_i \in [-5,5], i = 1, \ldots, 3$.

The analytical functions $f_1$ and $f_2$ are considered as the high fidelity accurate model $f$. A low fidelity model $c$ is formulated by adding noise to the analytical functions as shown in Eqns. (13) and (14). Figure 3-7 shows the Pareto optimal solutions obtained using the proposed algorithm. It clearly shows, the approach can capture discontinuous Pareto front effectively.
Figure 3-7. *Kursawe function results: comparison of Pareto obtained with the actual Pareto.*

**B. Two-Dimensional Aerodynamic Design Problem**

This section demonstrates a multi-objective optimization of a two-dimensional aerodynamic problem concerning the RAE2822 airfoil.

**1. Problem Description**

The objective is to find the trade-offs between the conflicting objectives, drag coefficient ($C_d$) and pitching moment coefficient ($C_m$) of the RAE 2822 at a free-stream Mach number of $M_\infty = 0.734$, lift coefficient of 0.824, and Reynolds number of $6.5 \times 10^6$ subject to an area and pitching moment constraint. We want to explore the designs in both feasible and infeasible regions while satisfying the area constraint at a constant lift coefficient. The conflicting objectives considered here are: drag minimization and pitching moment maximization, i.e., we have $F_1(x) = C_{d,f}$ and $F_2(x) = C_{m,f}$. The multi-objective constrained optimization problem can be expressed as

$$\begin{align*}
\min_{l \leq x \leq u} & \quad C_d, \\
\max_{l \leq x \leq u} & \quad C_m,
\end{align*}$$

subject to

$$C_l = 0.824,$$

$$C_m \geq -0.092,$$

$$A \geq A_{baseline}.$$
2. Design Variables

The B-spline parameterization approach, described in Jie et al (2016), [95], is used in this case to control the upper and lower surfaces of the airfoil. We use eight control points, as shown in Fig. 3-8, where two are fixed at the leading- and trailing-edges, and the other ones, four for each surface, can move in the vertical direction. This yields eight design variables.

Based on a fit to the RAE2822, we set the \( x \)-locations of the free control points as: \( \mathbf{X} = [\mathbf{X}_u; \mathbf{X}_l]^T = [0.0 \ 0.15 \ 0.45 \ 0.80; \ 0.0 \ 0.35 \ 0.60 \ 0.90]^T \). The initial design variable vector is \( \mathbf{x} = [\mathbf{x}_u; \mathbf{x}_l]^T = [0.0175 \ 0.0498 \ 0.0688 \ 0.0406; -0.0291 \ -0.0679 \ -0.0384 \ 0.0054]^T \). The lower bound of \( \mathbf{x} \) is set as \( \mathbf{l} = [0.015 \ 0.015 \ 0.015 \ 0.015; -0.08 \ -0.08 \ -0.08 \ -0.01]^T \), and the upper bound is set as \( \mathbf{u} = [0.08 \ 0.08 \ 0.08 \ 0.08; -0.01 \ -0.015 \ -0.015 \ 0.01]^T \).

![Figure 3-8. Airfoil shape parameterization using B-spline curves.](image)
3. High-Fidelity Viscous Aerodynamics Model

The SU\(^2\) implicit density-based flow solver is used for the viscous case, solving the steady compressible Reynolds-averaged Navier-Stokes (RANS) equations with the Spalart-Allmaras turbulent model [98]. The convective flux will be calculated using the second order JST scheme [99]. One multi-grid level is used for solution acceleration. The turbulent variables are convected using a first-order scalar upwind method. The flow solver convergence criterion is the one that occurs first of the two: (i) the change in the drag coefficient value over the last 100 iterations is less than \(10^{-5}\), or (ii) a maximum number of iterations of 20,000 is met.

The grids are generated using the hyperbolic C-mesh of Kinsey and Barth [100] (see Fig. 3-9). The farfield is set 100 chords away from the airfoil surface. The grid points are clustered at the trailing edge and the leading edge of the airfoil to give a minimum streamwise spacing of 0.001\(c\), and the distance from the airfoil surface to the first node is \(10^{-5}c\). The grid density is controlled by the number of points in the streamwise direction, and the number of points in the direction normal to airfoil surface. We set the number of points in the wake region equal to the number in the normal direction. Table 3-1 gives the results of a grid convergence study using the RAE 2822 airfoil at \(M = 0.734\) and \(C_l = 0.824\). The constant lift condition is determined by internally changing the angle of attack within the flow solver. The simulation time presented in Table 3-1 gives the overall time to compute the constant lift condition.

For the optimization studies, we use Mesh 3 for the high-fidelity model f. Mesh 4 is the finest and the most accurate. The difference between meshes 3 and 4 is around 1.75 drag counts for the baseline shape. However, Mesh 4 is almost five times more expensive than Mesh 3, hence the latter was chosen as the high-fidelity model for this work.
Figure 3-9. Hyperbolic C-mesh: (a) farfield view, (b) view close to the surface.

Table 3-1. Grid convergence study for the baseline shape of Benchmark Case 2.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Number of Elements</th>
<th>Lift Counts (l.c.)</th>
<th>Drag Counts (d.c.)</th>
<th>Simulation Time* (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9,836</td>
<td>0.824</td>
<td>324.6</td>
<td>3.1</td>
</tr>
<tr>
<td>2</td>
<td>38,876</td>
<td>0.824</td>
<td>221.5</td>
<td>8.8</td>
</tr>
<tr>
<td>3</td>
<td>154,556</td>
<td>0.824</td>
<td>204.8</td>
<td>34.0</td>
</tr>
<tr>
<td>4</td>
<td>616,316</td>
<td>0.824</td>
<td>203.0</td>
<td>152.6</td>
</tr>
</tbody>
</table>

* Computed on a high-performance cluster with 32 processors. Flow solution only.

4. Low-Fidelity Viscous Aerodynamics Model

The model set up for low-fidelity model is same as that of high-fidelity model (f). As shown in Table 3-1, we use Mesh 1 for the low-fidelity model c. The low-fidelity model convergence criteria are set with the following values: (i) change in the drag coefficient value over the last 100 iterations is less than $10^{-4}$, or (ii) the maximum number of iterations is set to 5,000. Figure 10 (a) shows that the low-fidelity solver is converged well within 5,000-iteration limit, and Figure 10 (b) shows that the low-fidelity model is a good representation of high-fidelity one in terms of the pressure coefficient distributions.
Figure 3-10. Viscous flow simulation results for RAE 2822 at \( M_\infty = 0.734, \ Cl = 0.824 \) and \( Re_\infty = 6.5 \times 10^6 \): (a) the evolution of the lift and drag coefficients obtained by the low-fidelity model, (b) a comparison of the pressure distributions obtained by the high- and low-fidelity models.

5. Single-Objective Optimization Results

The SOO problem is solved using space mapping (SM) algorithm [95] as discussed in Section II.B and the results are shown in Table 3-2. Algorithm (1) is executed iteratively using trust-region, gradient-based optimization to obtain the ‘Initial point’ (cf. Fig. 3-1). The gradient-based search uses the sequential quadratic programming algorithm, where the original problem is solved iteratively by replacing the original objective function (and nonlinear constraints) by their respective local quadratic models (linear for constraint functions). To obtain the gradient information, finite differences are utilized on the surrogate model with the finite difference step size set at \( 10^{-5} \). Once the optimum is reached, the design is evaluated on the high-fidelity model \( \mathbf{f} \) and fed into the gradient-based algorithm to search for further minima and the process is continued until convergence. Optimization of the surrogate model is constrained to the vicinity of the current design defined as \( \| \mathbf{x} - \mathbf{x}^{(i)} \| \leq \delta^{(i)} \), with the trust region radius \( \delta^{(i)} \) adjusted adaptively using the standard trust region rules [40]. The convergence tolerances for the termination conditions (defined in Section II.C) are set as \( \varepsilon_x = 10^{-4}, \ \varepsilon_H = 10^{-4}, \) and \( \varepsilon_\delta = 10^{-4} \).
Figure 3-11 shows the convergence history of the algorithm indicating a considerable reduction in the objective function compared to that of the baseline airfoil. As it can be seen in Table 3-2, the SM algorithm reduces the drag coefficient value from 203.80 drag counts (d.c) to 118.4 d.c (note that we define one d.c as $\Delta C_d = 10^{-4}$, and one lift count (l.c.) as $\Delta C_l = 10^{-2}$). Figure 3-12 (a) and (b) show comparisons of the shapes and pressure coefficient distributions of the baseline and SOO optimal designs. Figure 13 shows the pressure coefficient contours of the baseline and optimum shape design respectively, indicating a considerable reduction of shock strength. In terms of number of function evaluations, SM based optimization utilized 499 low fidelity models and only 4 high fidelity models. The cost in terms of CPU time for the entire optimization process is approximately 13.55 hours on a HPC with 32 processors.

<table>
<thead>
<tr>
<th>Parameter/method</th>
<th>Baseline</th>
<th>SOO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_l$ (l.c.)</td>
<td>82.4</td>
<td>82.4</td>
</tr>
<tr>
<td>$C_d$ (d.c.)</td>
<td>203.8</td>
<td>118.4</td>
</tr>
<tr>
<td>$C_{m,c/4}$</td>
<td>-0.0905</td>
<td>-0.0904</td>
</tr>
<tr>
<td>$A$</td>
<td>0.0779</td>
<td>0.0779</td>
</tr>
<tr>
<td>$N_c$</td>
<td>-</td>
<td>499</td>
</tr>
<tr>
<td>$N_f$</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>CPU Time (hours)</td>
<td>-</td>
<td>13.55</td>
</tr>
</tbody>
</table>
Figure 3-11. SOO showing (a) convergence of arguments, and (b) evolution of the objective.

Figure 3-12. SOO results showing baseline and optimized (a) airfoil shapes, and (b) pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$. 
Figure 3-13. SOO Mach contours at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$ of (a) the baseline airfoil, and (b) the SOO optimal design.

6. Point-by-Point Pareto Front Exploration

The design space exploration in feasible and infeasible regions is performed using the algorithm described in Section II. The initial design corresponding to the best possible value of the first objective (minimum drag coefficient) is obtained in the first step of the process using the output space mapping (SM) algorithm [95]. A target pitching moment value is then identified for the algorithm to attain while minimizing the drag coefficient value. The runs are performed at $M_\infty = 0.734$ and $A_{baseline} = 0.0779$. Subsequent designs along the Pareto were obtained using (2) and the process is terminated when the target pitching moment value reaches approximately +/- 25% of the pitching moment for initial design. The proposed algorithm needed one iteration to reach each target point. The cost of stepping from one point to another is approximately one hour. The total cost in terms of CPU time to obtain the entire Pareto is around 30 hrs on a HPC with 32 processors which includes cost of obtaining the initial design.

Figures 2-14 (a) and (b) show the optimal solution set (Pareto front) obtained and a zoomed-in plot near SOO optimum, respectively. The plots clearly reflect that; we cannot obtain any better optimum drag coefficient value other than SOO optimum. Few points (Point 1 and point
2) on the Pareto optimal set were selected to be compared with SOO optimum and the baseline design. Figures 2-15 (a) and (b) show comparisons of all the airfoil shape designs and the pressure coefficient distributions for the selected points. There is a significant difference in the pressure coefficient distribution of the selected points compared to that of SOO optimum with the later having a considerable reduction in shock strength. A similar pattern can be observed from the Mach contour plots in Fig. 2-16 which shows point 1 with higher shock strength leading to more drag compared to point 2. Both the results of Figs. 2-15 and 2-16 align with the fact that to obtain a lower drag than the SOO point which is point 2, there will be a decrease in pitching moment as shown in Fig. 2-14 (a).

Figure 3-14. Multi-objective optimization results at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$: (a) optimum solutions, (b) zoomed-in plot.
Figure 3-15. MOO results showing (a) the airfoil shapes, and (b) the pressure distributions at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$.

Figure 3-16. Mach contours at $M_{\infty} = 0.734$, $C_l = 0.824$ of (a) Point 1, (b) Point 2.

C. Algorithm Scalability

The scalability of the point-by-point algorithm is investigated with respect to the dimensionality of the design space. More specifically, we are interested in characterizing how the computational cost of the algorithm varies with the design space dimensionality. Using the design problem described in Section II.B, the dimensionality is varied by changing the number
of control points in the airfoil shape parameterization (cf. Section III.B.2) from 4 to 12, in steps of 2, and the Pareto front for each case is determined using the proposed algorithm. The simulation models are the same ones as described in Section II.B, and remain fixed during the study. The results of the SOO and MOO runs are presented separately.

Tables 3-3 and 3-4 give the baseline design values and the results of the SOO for each design parameterization, respectively. Figure 3-17 shows how the drag coefficients of the optimal designs and the corresponding computational cost vary with the number of design variables. Figures 3-18 and 3-19 give the shapes and characteristics of the airfoils with 8, 10, and 12 design variables.

The SOO results show that the optimal designs are improved with the higher dimensionality. The case with 4 design variables does yield a better design than the baseline. The case with 6 design variables improves the design from 277.7 d.c. to 151.0 d.c., or by 126.7 d.c. (−45.6%). The drag coefficients of the optimal design with 8, 10, 12 design variables are comparable to each other, however, ranging from 118.4 d.c. to 117.2 d.c (giving a reduction of 73.4 d.c. (−38.5%) with respect to the baseline in the case with 12 design variables). This indicates that adding more designable parameters in this case may not yield significant improvements in the objective function. In terms of the computational cost, it is interesting to notice that the number of high-fidelity model evaluations is the same for the cases with 8, 10, and 12 design variables. However, the number of low-fidelity model evaluations grows quickly with increasing dimensionality.
Table 3-3. Aerodynamic coefficients of the baseline design as a function of the number of design variables.

<table>
<thead>
<tr>
<th>Coefficients/Number of Design Variables</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_l$ (l.c.)</td>
<td>82.4</td>
<td>82.4</td>
<td>82.4</td>
<td>82.4</td>
<td>82.4</td>
</tr>
<tr>
<td>$C_d$ (d.c.)</td>
<td>197.0</td>
<td>277.7</td>
<td>203.8</td>
<td>193.6</td>
<td>190.6</td>
</tr>
<tr>
<td>$C_{m,c/4}$</td>
<td>-0.0829</td>
<td>-0.0878</td>
<td>-0.0905</td>
<td>-0.0975</td>
<td>-0.0996</td>
</tr>
<tr>
<td>$A$</td>
<td>0.0779</td>
<td>0.0778</td>
<td>0.0779</td>
<td>0.0778</td>
<td>0.0778</td>
</tr>
</tbody>
</table>

Table 3-4. SOO results as a function of the number of design variables.

<table>
<thead>
<tr>
<th>Parameters/Number of Design Variables</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_l$ (l.c.)</td>
<td>82.4</td>
<td>82.4</td>
<td>82.4</td>
<td>82.4</td>
<td>82.4</td>
</tr>
<tr>
<td>$C_d$ (d.c.)</td>
<td>197.0</td>
<td>151.0</td>
<td>118.4</td>
<td>117.8</td>
<td>117.2</td>
</tr>
<tr>
<td>$C_{m,c/4}$</td>
<td>-0.0829</td>
<td>-0.0922</td>
<td>-0.0904</td>
<td>-0.0905</td>
<td>-0.0900</td>
</tr>
<tr>
<td>$A$</td>
<td>0.0779</td>
<td>0.0779</td>
<td>0.0779</td>
<td>0.0779</td>
<td>0.0779</td>
</tr>
<tr>
<td>$N_c$</td>
<td>37</td>
<td>354</td>
<td>499</td>
<td>573</td>
<td>700</td>
</tr>
<tr>
<td>$N_f$</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>CPU Time (hours)</td>
<td>0.4</td>
<td>11.8</td>
<td>13.6</td>
<td>16.6</td>
<td>26.3</td>
</tr>
</tbody>
</table>

Figure 3-17. SOO results showing the variation of (a) the drag coefficient and (b) total optimization time as a function of the number of design variables.
The results of Table 3-4 for 8, 10, and 12 design variables are utilized as the starting points for the algorithm of Section II. The algorithm is executed to yield the Pareto fronts as shown in Fig. 3-20. The MOO results clearly indicate that increasing the parameterization complexity (here, the number of control points), leads to overall better solutions. In particular, the fronts obtained for a smaller number of design variables are generally dominated by those obtained
at a higher-dimensional parameter spaces. Clearly, this is achieved at the expense of the increased computational time. Table 3-5 provides shows a comparison of the number of low- and high-fidelity model evaluations as well as the overall CPU time for the three considered test cases.

![Comparison of the Pareto fronts obtained using 8, 10 and 12 design variables at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$.](image)

**Table 3-5.** Comparison of the computational cost of the MOO as a function of the number of design variables.

<table>
<thead>
<tr>
<th>Parameter/Number of Design Variables</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_c$</td>
<td>462</td>
<td>651</td>
<td>714</td>
</tr>
<tr>
<td>$N_f$</td>
<td>22</td>
<td>31</td>
<td>34</td>
</tr>
<tr>
<td>CPU Time (hours)</td>
<td>18.7</td>
<td>26.35</td>
<td>29.5</td>
</tr>
</tbody>
</table>
D. Algorithm Robustness

The robustness of the proposed algorithm of yielding the same (or comparable) Pareto fronts in terms of different initial points and direction of exploration along the front is investigated. In particular, the Pareto front obtained for the case with 8 design variables (shown in Sects. II.B and II.C) is compared with a Pareto front obtained when starting from another initial point and moving along the front in the opposite direction. Figure 3-21 shows the two Pareto fronts where the black points are the prior results and the red points are new results. Overall, the two Pareto fronts are comparable with the largest discrepancy being around one drag count, which is within the numerical accuracy of the high-fidelity CFD model used in this case.

![Figure 3-21. Robustness of the proposed algorithms is demonstrated by obtaining comparable (within one drag count) Pareto fronts when starting from two different initial points.](image)
IV. Conclusion

An efficient methodology for aerodynamic design exploration has been presented. The Pareto front, which represents the best possible trade-offs between conflicting objectives, is identified in a point-by-point manner using fast multi-fidelity aerodynamic models. The proposed approach has a number of features that distinguish it from other surrogate-assisted aerodynamic design exploration methods available in the literature: (1) multi-fidelity models are utilized to quickly locate one point on the Pareto front, (2) the Pareto front is determined point-by-point in the vicinity of the initial point, (3) the algorithm does not rely on population-based metaheuristics, (4) design space confinement is not required, (5) objective function aggregation is not required, and (6) objective function gradients are not needed. Comprehensive numerical studies involving both analytical test cases as well as aerodynamic shape optimization problems demonstrate the computational efficiency and robustness of the proposed approach. Furthermore, the algorithm has been shown to have good (here, close to linear) scalability with respect to the dimensionality of the design space. Future work will investigate extensions of the technique to handle more complex problems in terms of a larger number design variables as well as the degrees of freedom of the simulation model.

References


CHAPTER 4. APPLICATIONS OF SURROGATE-ASSISTED AND MULTI-FIDELITY MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS TO SIMULATION-BASED AERODYNAMIC DESIGN


Structured Abstract

Purpose

The purpose of this work is to apply and compare surrogate-assisted and multi-fidelity multi-objective optimization algorithms to simulation-based aerodynamic design exploration.

Design/methodology/approach

The three algorithms for multi-objective aerodynamic optimization compared in this work are (1) the combination of evolutionary algorithms, design space reduction, and surrogate models, (2) the multi-fidelity point-by-point Pareto set identification, and (3) the multi-fidelity sequential domain patching (SDP) Pareto set identification. The algorithms are applied to three cases: (a) an analytical test case, (b) design of transonic airfoil shapes, and (c) the design of subsonic wing shapes, and are evaluated based on the resulting best possible trade-offs and the computational overhead.

Findings

The results show that all three algorithms yield comparable best possible trade-offs for all the test cases. For the aerodynamic test cases, the multi-fidelity Pareto set identification algorithms outperform the surrogate-assisted evolutionary algorithm by up to 50% in terms of cost. Furthermore, the point-by-point algorithm is around 27% more efficient than the SDP algorithm.

Originality/value

The novelty of this work include (a) the first applications of the SDP algorithm to multi-fidelity aerodynamic design exploration, (b) the first comparison of these multi-fidelity multi-objective optimization algorithms, and (c) new results of a complex simulation-based multi-objective
aerodynamic design of subsonic wing shapes involving two conflicting criteria, several nonlinear constraints, and over ten design variables.

**Keywords:** simulation-based aerodynamic design, transonic airfoil design, subsonic wing design, multi-fidelity methods, multi-objective optimization, Pareto front exploration.

**Nomenclature**

\[ A = \text{airfoil cross-sectional area, m}^2 \]
\[ a_\infty = \text{speed of sound, m/s} \]
\[ b = \text{wing span, m} \]
\[ C_d = \text{airfoil section drag coefficient, } d/(q_\infty c) \]
\[ C_D = \text{wing drag coefficient, } D/(q_\infty S) \]
\[ C_m = \text{airfoil section pitching moment coefficient, } M/(q_\infty c^2) \]
\[ C_{Mx} = \text{wing pitching moment coefficient, } m/(q_\infty c S) \]
\[ C_l = \text{airfoil section lift coefficient, } l/(q_\infty c) \]
\[ C_L = \text{wing lift coefficient, } L/(q_\infty S) \]
\[ c = \text{low-fidelity model output} \]
\[ c = \text{chord length, m} \]
\[ d = \text{airfoil section drag force, N} \]
\[ D = \text{wing drag force, N} \]
\[ d = \text{trust-region radius, m} \]
\[ f = \text{high-fidelity model output} \]
\[ H = \text{objective function value} \]
\[ l = \text{lower bound of } x \]
\[ l = \text{airfoil section lift force, N} \]
\[ L = \text{wing lift force, N} \]
\[ M_\infty = \text{Mach number, } V_\infty/a_\infty \]
\[ m = \text{pitching moment} \]
\( n \) = number of design variables
\( N_c \) = number of coarse model evaluation
\( N_f \) = number of fine model evaluation
\( q_\infty \) = dynamic pressure, \( \frac{1}{2} \rho_\infty V_\infty^2 \)
\( s \) = surrogate model output
\( S \) = wing planform area, \( \text{m}^2 \)
\( u \) = upper bound of \( x \)
\( V_\infty \) = flow speed, \( \text{m/s} \)
\( V \) = wing volume, \( \text{m}^3 \)
\( x \) = design variables, \( \text{m} \)
\( y \) = coordinate along the wing span, \( \text{m} \)
\( z_{c/4} \) = vertical coordinates of the quarter-chord location in a wing
\( \gamma \) = wing twist, degrees
\( \alpha \) = angle of attack, degrees
\( \rho_\infty \) = density, \( \text{kg/m}^3 \)
\( \eta \) = wing span stations, \( y/(b/2) \)

I. Introduction

The design of aerodynamic surfaces is important for modern engineered systems such as unmanned aerial systems, turbomachinery, and aircraft. Physics-based simulations are essential in analyzing and optimizing aerodynamic surfaces, especially for capturing nonlinear system behavior and nonlinear interactions between multiple disciplines [1]. The main challenges with using accurate physics-based simulations as part of analysis and optimization include (1) high computational cost (ranging from few hours to days or weeks on high performance computing clusters), (2) repetitive evaluations during the design exploration phase, and (3) large numbers of conflicting objectives and constraints, and design variables. In
this paper, three recently developed approaches for obtaining the best possible trade-offs for aerodynamic surfaces are applied for the first time to simulation-based aerodynamic design of a subsonic wing and compared in terms of the resulting best trade-offs and the computational overhead.

Design exploration can be executed in various ways: (1) design by engineering experience, (2) exhaustive search in design space, (3) single-objective optimization (SOO), and (4) multi-objective optimization (MOO). Among all SOO techniques, gradient-based search techniques [2] and constrained single-objective optimization (SOO) techniques [3, 4] has been widely used for aerodynamic design. SOO methods, however, yield only one optimal solution in a single run.

Designers, typically, prefer to have numerous optimal solutions or the best possible trade-offs between multiple conflicting objectives represented by a Pareto front [5]. Approaches to estimate Pareto fronts include evolutionary algorithms (EAs) [6], multi-objective genetic algorithms (MOGAs) [7], and multi-objective EAs (MOEAs) [8]. Other multi-objective approaches include particle swarm optimization [9], differential evolution [10], firefly [11] and cuckoo search [12]. All these approaches are metaheuristic methods and require many model evaluations, thus making it impossible to use for aerodynamic design exploration problems.

High-fidelity computational fluid dynamics (CFD) simulations are important to the design of complex engineering systems. Unfortunately, high-fidelity CFD simulations can be computationally expensive and, therefore, it can be impractical to apply them directly in MOO. Surrogate-based optimization (SBO) [13-15] techniques are widely used to alleviate the computational burden. Various surrogate modeling approaches are utilized for this purpose, such as response surface approximations [15], radial-based function models [14], and kriging
interpolation [14]. The SBO process has been successfully implemented as part of several MOO algorithms (called surrogate-assisted MOO algorithms). The Pareto-based Efficient Global Optimization [16] (ParEGO) uses the weighted-sum approach and surrogate modeling for MOO. Similarly, the Pareto Set Pursuing (PSP) approach [17] utilizes global surrogate models for MOO. Adaptive sampling and surrogate modeling are combined for MOGAs in [18], surrogate-assisted MOO using global and local models is introduced in [19] and finally a global approximation-based MOO for robust design under interval uncertainty is described in [20].

Surrogate-based model construction can, however, involve considerable amount of high-fidelity model evaluations, which can become impractical in some cases. Another mechanism to alleviate the computational burden is to use multi-fidelity methods, which are methods that leverage the computational speed of models of low fidelity and the accuracy of the high-fidelity ones [21]. The main elements of multi-fidelity methods are low-fidelity modeling methods and model management strategies [21]. Low-fidelity modeling methods include reduced order models (e.g., simplified governing equations [22], and coarse discretization [23]), projection-based methods (e.g., proper orthogonal decomposition [24], and reduced basis method [25]), and data-fit methods (e.g., radial basis functions, kriging, and support vector regression [26]). Model management strategies include adaptation, fusion, and filtering [21]. Filtering methods include invoke the high-fidelity model following the evaluation of a low-fidelity filter. Examples of filtering methods include the multi-fidelity stochastic collocation approach [27]). Fusion methods construct a fast multi-fidelity model by combining the output data from the low- and high-fidelity models. Examples of such methods include cokriging [28], and Bayesian regression [29]. Adaptation approaches are either global methods (e.g., efficient global
optimization (EGO) using global data-fit models and infill criteria [15]) or local methods (e.g., space mapping [30], comprehensive [31], multiplicative [34], or SBO methods using local data-fit models [15]).

Approaches for multi-objective design exploration of aerodynamic surfaces typically rely on the weighted-sum method and gradient-based search techniques, see, e.g. [51-53]. There have been, however, several efforts using evolutionary methods and surrogate models. Multi-objective design optimization of regional jet aircraft wing shapes was presented by Chiba et al. [1] using high-fidelity CFD models and evolutionary algorithms. Mengsitu et al. [33] demonstrated multi-objective optimization of turbine blades using EAs and surrogate models based on artificial neural networks. Xu et al. [34] demonstrated multi-objective aerodynamic design optimization of high-speed trains using kriging surrogate models and GAs.

As far is known, not many works involving multi-objective aerodynamic design optimization using multi-fidelity methods have been reported in the literature. Leifsson et al. [54] used multi-fidelity models and local response surface surrogate models for multi-objective design exploration of transonic airfoil shapes. Amrit et al. [8] developed design strategies for multi-objective aerodynamic design exploration using design space reduction, evolutionary algorithms, cokriging [28, 31], and multi-fidelity models. In our recent work [35], we developed a multi-objective aerodynamic design optimization algorithm using multi-fidelity models and the point-by-point Pareto set identification technique, originally developed for microwave antenna design [37]. The outcome of these work has shown that multi-fidelity methods can significantly reduce the computational overhead of multi-objective aerodynamic design exploration.
This paper presents new applications and comparisons of three recently developed surrogate-assisted and multi-fidelity MOO algorithms. In particular, the algorithms are (1) the surrogate-assisted MOEA (SA-MOEA) developed by Amrit et al. [8], (2) the multi-fidelity multi-objective aerodynamic design exploration using Point-by-point Pareto set identification [35], and (3) the multi-fidelity sequential domain patching (SDP) algorithm, originally developed by Koziel et al. [36] for the design of microwave antennas. The novelty of this work is (a) the first applications of the SDP algorithm to multi-fidelity aerodynamic MOO, and (b) new results of simulation-based multi-objective aerodynamic design of subsonic wing shapes involving two conflicting criteria, several nonlinear constraints, and over ten design variables. Furthermore, the algorithms are applied to an analytical test case and the design of transonic airfoil shapes. The paper presents the results of these application studies and compares the algorithms in terms of the resulting best possible aerodynamic trade-offs, as well as the computational overhead.

The next section gives the details of the multi-objective aerodynamic design exploration algorithms. The following section presents results of applications of the algorithms to an analytical problem, transonic airfoil design in viscous flow, and subsonic rectangular wing design in inviscid flow. The paper ends with conclusion and remarks on future work.

II. Methods

This section describes about Pareto front and its global and local exploration methods, and gives the details of the multi-objective optimization algorithm and the multi-fidelity modeling.

A. Definition of the Pareto Front

Here, the concept of Pareto front is explained using a specific example of an aerodynamic design problem. The goal is to find a trade-off between various aerodynamic forces such as lift, drag, and pitching moment coefficients, denoted as $C_{l,f}$, $C_{d,f}$, and $C_{m,f}$, respectively. Let an
accurate high-fidelity aerodynamics simulation model be denotes as \( f(x) = [C_l(x) \ C_d(x) \ C_m(x)]^T \), where \( x \) is the \( n \times 1 \) vector of design variables.

Let \( F_k(x), k = 1, \ldots, \text{Nobj} \), be a \( k \)th design objective of interest. If \( \text{Nobj} > 1 \) then any two designs \( x^{(1)} \) and \( x^{(2)} \) for which \( F_k(x^{(1)}) < F_k(x^{(2)}) \) and \( F_l(x^{(2)}) < F_l(x^{(1)}) \) for at least one pair \( k \neq l \), are not commensurable, i.e., none is better than the other in the multi-objective sense. We define a Pareto dominance relation \( \prec \), saying that for the two designs \( x \) and \( y \), we have \( x \prec y \) (\( x \) dominates \( y \)) if \( F_k(x) \leq F_k(y) \) for all \( k = 1, \ldots, \text{Nobj} \), and \( F_k(x) < F_k(y) \) for at least one \( k \) [49].

The goal of the multi-objective optimization is to find a representation of a so-called Pareto front (of Pareto-optimal set) \( X_P \) of the design space \( X \), such that for any \( x \in X_P \), there is no \( y \in X \) for which \( y \prec x \).

**B. Algorithm 1: Pareto Front Exploration Using Surrogate-Assisted Multi-Fidelity Based Multi-Objective Evolutionary Algorithm**

The SA-MOEA-based algorithm [8] utilizes the two ends of the Pareto front as its starting point and is obtained using space-mapping based single objective optimizations. To expedite the optimization procedure, the algorithm involves utilization of surrogate models constructed from fast low-fidelity model, \( c \), based on coarse-discretization CFD simulations. A design speedup is achieved by performing most of the operations using the data driven models while using few high-fidelity models to refine the model and yield a Pareto set that is sufficiently accurate.

The steps of the SA-MOEA-based algorithm [8] follow:

1. Setup a physics-based surrogate \( s_0 \);
2. Perform design space reduction using \( s_0 \);
3. Sample the design space and acquire the surrogate model data with $s_0$;

4. Construct a kriging surrogate $s_{\text{KR}}$ based on the data from Step 3;

5. Obtain an approximate Pareto set representation by optimizing $s_{\text{KR}}$ using MOEA [38];

6. Evaluate the high-fidelity model $f$ along the Pareto front;

7. Construct/update the co-kriging surrogate $s_{\text{CO}}$;

8. Update Pareto set by optimizing $s_{\text{CO}}$ using MOEA [38]; and

9. If termination condition is not satisfied go to Step 6; else END

In Step 1, instead of using expensive high-fidelity PDE simulations to search a Pareto front, a fast surrogate model is utilized to speed up the process. The surrogate model includes a combination of physics-based and data-driven surrogate models which is clearly described in [30]. Setting up an accurate surrogate model can be expensive on a large design space. Hence, in Step 2, the design space is reduced (in terms of design variable ranges, as well as dimensionality) within which, kriging (Step 4) and co-kriging (Step 7) models is set up using a limited number of model evaluations. The design space reduction methodology is described [8]. In Step 3, Latin Hypercube Sampling [39] (LHS) is used to select the shapes within the reduced design space and the corresponding low-fidelity model values for each sample is collected. A kriging interpolation model is constructed in Step 4, using the sampled data from previous step. Amrit et.al [8] describes the surrogate model construction. In Step 5, a MOEA is run using the kriging model constructed in Step 4. However, the Pareto optimal set obtained is not accurate, as the kriging model was based on low-fidelity model. In Step 6 and 7, a few designs are selected uniformly along the Pareto front predicted by the kriging model optimization. A high-fidelity model evaluation is performed on the selected designs and are
used to construct a cokriging model as explained in [8, 28]. Step 8 involves using the MOEA and the co-kriging model to refine the Pareto front. If the alignment between the Pareto front and the samples evaluated on high-fidelity is sufficient, the algorithm is terminated. The convergence condition is based on the distance between the predicted front and the high-fidelity verification samples (distance measured in the feature space). In Step 9, if the convergence criteria is not met, the cycle is repeated from Step 6 until convergence.

C. Algorithm 2: Pareto Front Exploration Using Sequential Domain Patching Algorithm

The SDP algorithm involves obtaining optimal solutions in between two points on the Pareto-front. Once the extreme ends of the Pareto front to be explored are obtained (two constrained SOOs are executed [36]), the SDP-based MOO algorithm is executed, as explained in Koziel et al. [36], to obtain an initial Pareto set (see Fig. 4-1).

The multi-fidelity SDP algorithm, as shown in Fig. 4-1, produces a sequence of designs $\mathbf{x}^{(k)*}$, $k = 1, 2, \ldots, N$, where $\mathbf{x}^{(1)*}$ and $\mathbf{x}^{(N)*}$ are the extreme ends of the Pareto front and solutions to the single-objective optimization problem. It can be formally summarized as shown below considering only two design objectives:

1. Patch size $\mathbf{d} = [d_1 \ldots d_n]^T$ is set using the procedure described in Koziel et al. [36];
2. Current points are set as $\mathbf{x}_{c1} = \mathbf{x}^{(1)*}$ and $\mathbf{x}_{cN} = \mathbf{x}^{(N)*}$;
3. $n$ perturbations of the size $\mathbf{d}$ are evaluated around $\mathbf{x}_{c1}$ (towards $\mathbf{x}_{cN}$ only) and the one that brings the largest improvement with respect to the second objective $F_2$ is selected.
4. The patch is relocated so that it is centered at the best perturbation selected in Step 3; $\mathbf{x}_{c1}$ is updated;
5. $n$ perturbations of the size $\mathbf{d}$ are evaluated around $\mathbf{x}_{cN}$ (towards $\mathbf{x}_{c1}$ only) and the one that brings the largest improvement with respect to the second objective $F_1$ is selected.
Figure 4-1. An illustration of the SDP algorithm showing (a) the design space, and (b) the Pareto front in the feature space.

6. The patch is relocated so that it is centered at the best perturbation selected in Step 5; $x_{c\mathcal{N}}$ is updated;

7. If the path between $x^{(1)*}$ and $x^{(N)*}$ is not complete, go to Step 3.

The multi-fidelity SDP algorithm discussed above is used to determine the initial Pareto at the level of low-fidelity model $c$. To obtain the high-fidelity Pareto-optimal designs $x^{(k)}_c$, $k = 1, \ldots, N$, the following Pareto set refinement procedure is executed:

$$x^{(k)}_f \leftarrow \arg \min_{x \in \mathcal{X}} \left\{ F_1(s_q(x) + [f(x^{(k)}_f) - s_q(x^{(k)}_f)]) \right\} \quad (1)$$

In this refinement process, the first objective is improved without degrading the second objective. The above process begins with $x^{(k)}_f = x^{(k)}_c$ as the starting point and the process is iterated until convergence is met. The correction term in (1) makes sure that $s_q(x^{(k)}_f) = f(x^{(k)}_f)$ at the initiation of each iteration. The surrogate model, $s_q$, used in this process is a second-order polynomial approximation without the mixed terms given by
\[ s_q(x) = a_0 + \sum_{k=1}^{n} a_k x_k + \sum_{k=1}^{n} a_{n+k} x_k^2, \]  

(2)

where \( a_k, k = 0, 1, 2, \ldots, 2n \), are the coefficients of the quadratic RSA, and \( x_k, k = 1, \ldots, n \) is the \( k \)th component of the variable vector \( x \). This approximation model is based on low-fidelity model, \( c \) evaluated at \( x_c^{(k)} \) and the perturbed designs within the patch surrounding \( x_c^{(k)} \).

The algorithm covers a part of the design space in the form of patches that contains the initial approximation of the set of Pareto-optimal solutions. The total computational cost of the algorithm can be computed as \((M - 1)(n - 1)\) which excludes the cost of solving (1), where \( M = \sum_{k = 1, \ldots, n} m_k \), and \( m_k \) is the number of intervals in the direction \( j \). However, in practice the total cost can be lower as some perturbations may not be evaluated due to the imposed constraints.

In Step 1, two SOO problems are solved on a surrogate model, constructed using space mapping algorithm [40], to obtain the two end points of the Pareto-front to be explored. The solutions from the SOO problems are used as an input to the automated domain patching algorithm (explained in Koziel et al. [36]) to obtain the patch size \( d \). Solutions from the initial two SOO problems are marked in Fig.1 (b) as \( x^{(1)*} \) and \( x^{(N)*} \). In Step 2, the next optimum solution is searched in vicinity of these starting points while moving in either direction, i.e. from \( x^{(1)*} \) to \( x^{(N)*} \) or vice versa. In Step 3, a patch is constructed with \( n \) perturbations of size \( d \) around \( x^{(1)*} \) and/or \( x^{(N)*} \). Each perturbation is evaluated on the low fidelity model to obtain objective functions and constraints values. The design that brings the largest improvement with respect to the second objective \( F_2 \). The search for largest improvement in \( F_2 \) is performed with a given condition that the designs are well within the global bounds and also they satisfy the linear and non-linear constraints. To satisfy the constraints, a surrogate model like equation can be used as discussed in Koziel et.al. [36]. In Step 4, the best perturbation result obtained from Step 3 is used to update Step 2, i.e. \( x_{i+1} \) is updated. The patch is relocated, so that the center
of the patch is the updated $x_{c1}$. A patch is constructed in Step 5 with $n$ perturbations of size $d$ around $x^{(N)*}$. Each perturbation is evaluated on the low fidelity model to obtain objective functions and constraints values. The design that brings the largest improvement with respect to the second objective $F_1$. The search for largest improvement in $F_1$ is performed such that the designs are well within the global bounds and also they satisfy the linear and non-linear constraints. In Step 6, The best perturbation result obtained from Step 5 is used to update Step 2, i.e. $x_{cN}$ is updated. The patch is relocated, so that the center of the patch is the updated $x_{cN}$. Finally, in Step 7, the steps from 2-6 is continued until the path between $x^{(1)*}$ and $x^{(N)*}$ is complete.

D. Algorithm 3: Pareto Front Exploration Using the Point-by-Point Algorithm

The multi-fidelity point-by-point algorithm [35] utilizes any single point on the Pareto front as its starting point and is obtained by a space-mapping based single objective optimization. From the starting point, the multi-objective optimization algorithm produces a sequence of optimal designs $x^{(k)}$, $k = 1, 2, \ldots$, point by point. The starting point, $x^{(1)}$, is a solution to the SOO problem of the form

$$x^{(1)} = \arg \min_x F_1(f(x)).$$

subjected to

$$F_2(f(x)) - b \leq 0.$$ 

where $b$ is the threshold value for the second objective, and $f(x)$ is the output of the accurate high-fidelity simulation model. Thus, an optimum design value of the first objective is obtained which lies on the Pareto front.

The SOO process in (3) is expedited by a trust-region-based multi-fidelity optimization algorithm [41-44] using output space mapping [30] to construct the multi-fidelity model. The
output space mapping approach exploits a combination of the accurate high-fidelity model $f$, and a model $c$, which is of lower fidelity than $f$, but is computationally faster to evaluate. Here, the low-fidelity model $c$ is based on coarse-discretization CFD simulations (see, e.g., Refs. [41-44] for a discussion on approaches for low-fidelity modeling).

Starting from the optimal design obtained from (3), further designs that are determined along the Pareto front, $x^{(k)}$, $k = 2, 3, \ldots$, are obtained by exploring it point-by-point [37], using a constrained SOO process as shown below. Let $F_2^{(k)}$ be the threshold value for the second objective, then we have

$$x^{(k)} = \arg \min_{x, F_2(f(x)) \leq F_2^{(k)}} F_1(f(x)).$$

(4)

Here, $x^{(k)}$, $k = 2, 3, \ldots$, is the $k^{th}$ element of the Pareto set and the process is continued until the design specifications are met. Pareto optimal points can be obtained in both directions along the Pareto, i.e., above and below the optimal design from solution of (3). The search for the point $x^{(k)}$ close to the design $x^{(k-1)}$ is performed using surrogate-based optimization as follows. Specifically, $x^{(k)}$ is obtained iteratively as a sequence of solutions to (4) $x^{(k,j)}$, $j = 0, 1, \ldots$, with $x^{(k,0)} = x^{(k-1)}$, as follows

$$x^{(k,j+1)} = \arg \min_{x, F_2(s(x)) \leq F_2^{(k)}} F_1(s^{(k,j)}(x)),$$

(5)

where the surrogate $s^{(k,j)}$ is defined as

$$s^{(k,j)}(x) = s_q^{(k,j)}(x) + [f(x^{(k,j)}) - s_q^{(k,j)}(x^{(k,j)})]$$

(6)

where $s_q^{(k,j)}$ is a local response surface approximation (RSA) [14] model of $c$, constructed in the vicinity of current design $x^{(k,j)}$. The vicinity is defined here by the patch (or interval range) $[x^{(k,j)} - \delta, x^{(k,j)} + \delta]$, where $\delta$ is the trust region radius [4]. The RSA model is described here below.
The multi-fidelity point-by-point algorithm is summarized as follows:

1. Obtain $x^{(1)}$ by performing single-objective multi-fidelity optimization;
2. Set $F_2^{(1)} = F_2(f(x^{(1)}))$ and set $k = 2$;
3. Set $x^{(k,0)} = x^{(k-1)}$;
4. Evaluate $f^{(k,j)}(x^{(k,j)})$;
5. Construct a local RSA model $s^{(k,j)}(x)$ within the interval defined as $[x^{(k,j)} - \delta, x^{(k,j)} + \delta]$;
6. Use (5) to obtain $x^{(k,j+1)}$ by optimizing the surrogate model $s^{(k,j)}(x)$;
7. If $j = 2$, set $x^{(k)} = x^{(k,j+1)}$ and go to Step 8; otherwise update the index $j = j + 1$ and go to Step 4;
8. If the termination condition is met, terminate the algorithm; otherwise update the index $k = k + 1$, set the threshold $F_2^{(k)}$ and go to Step 3.

A local patch is formed encompassing the starting point obtained from Step 4, as shown in Fig. 4-2. The patch dimension is determined by performing a sensitivity analysis once and is given by $[x^{(k,j)} - \delta, x^{(k,j)} + \delta]$. Normally, $\delta$ is set at 0.5 - 1% of the difference of upper and lower bounds of the design space.

Figure 4-2. An illustration of the point-by-point Pareto front exploration progression for (a) two design variables in the design space and (b) two objectives in the feature space.
Furthermore, in Step 5, the RSA is constructed within the local patch using a quadratic model $s_q$ without mixed terms (cf. (2)). In this work, the surrogate $s_q$ is setup using a star distribution with $2n + 1$ samples (see, e.g., Koziel et al. [14]) evaluated by the low-fidelity model $(c)$. Subsequently, the surrogate model, $s$, is constructed using the RSA and one high-fidelity model evaluation (cf. (6)).

A gradient-based search engine is used to perform the minimization as discussed in Step 6 with the gradients obtained by finite differentiation of the surrogate model (cf. (5)). The target values of the second objective function, $F_2^{(k)}$, is set a-priori by the designer, and the inequality constraint is handled directly. The results from gradient-based algorithm is checked for the following three criteria, 

(i) $\|x^{(i)} - x^{(i-1)}\| < \varepsilon_x$, 
(ii) $|F_1^{(i)} - F_1^{(k)}| < \varepsilon_F$, 
(iii) $\delta^{(i)} < \varepsilon_\delta$, where $\varepsilon_x$, $\varepsilon_F$, and $\varepsilon_\delta$ are user defined convergence tolerances. In our implementation, the three criteria are combined through the logical alternative (OR). If the criteria are met, it is evaluated on the high-fidelity model, to obtain the target point. If not, the patch is updated using a trust region radius framework [6], based on which the trust-region size is reduced by setting $\delta = \delta/m$, where $m = 3$, or increased by setting $\delta = \delta \cdot m$, where $m = 2$.

In Step 7, the target point obtained from Step 6 is set as the starting point for the next algorithm iteration. The process is continued until the designer is satisfied (or a computational budget is exhausted), or the extreme point of the Pareto front is reached (i.e., the first objective cannot be further improved).

### III. Numerical Examples

This section presents results of the surrogate-assisted and multi-fidelity MOO algorithms on an analytical problem, transonic airfoil design, and a subsonic wind design.
A. Analytical Problem

The problem is identified to demonstrate the performance of the algorithms [45]. The objective functions of the test problem with five design variables are given by

\[
\begin{align*}
\min f_1 &= 1 - \exp[-\sum_{i=1}^{5}(x_i - \frac{1}{\sqrt{5}})^2], \\
\min f_2 &= 1 - \exp[-\sum_{i=1}^{5}(x_i + \frac{1}{\sqrt{5}})^2],
\end{align*}
\]

where

\[x_i \in [-4, 4].\]

The analytical functions \(f_1\) and \(f_2\) are considered as the high fidelity accurate model \(f\). A low fidelity model \(c\) is formulated by adding noise \((\Delta f)\) to the analytical functions as

\[
\begin{align*}
f_{1,c} &= f_1 + \Delta f, \\
f_{2,c} &= f_2 + \Delta f,
\end{align*}
\]

where

\[\Delta f = 0.1x_1 + 0.5.\]

Figure 4-3 shows the characteristic features of low- and high-fidelity models. Figure 4-4 (a) and (b) shows the Pareto optimal designs and the final Pareto front, respectively, obtained by the multi-fidelity multi-objective optimization algorithms. For the purpose of visualization only a part of the Pareto obtained from SDP and Point-by-point algorithm is plotted followed by the entire Pareto from MOEA algorithm. It can be seen that the Pareto front obtained by the algorithms are comparable.
Figure 4-3. *Example high and low-fidelity evaluation values of five random designs.*

Figure 4-4. *Pareto front obtained for Fonseca and Fleming function (a) design space, (b) feature space.*

**B. Transonic Airfoil Design**

In this section, results of the surrogate-assisted and multi-fidelity multi-objective optimization of transonic airfoil shapes using a viscous fluid flow model is presented. We give the problem description, design variable formulation, and results for the SOO and the Pareto front.
1. Problem Description

The main goal of this aerodynamic problem is to obtain the trade-offs between two conflicting objectives, the drag coefficient ($C_d$) and the pitching moment coefficient ($C_m$), of the RAE 2822 at a free-stream Mach number of $M_o = 0.734$, lift coefficient ($C_l$) of 0.824, and Reynolds number ($Re$) of $6.5 \times 10^6$, subject to an area cross-sectional ($A$) constraint. Specifically, the conflicting objectives considered here are: drag minimization and pitching moment maximization, i.e., we have $F_1(x) = C_{d, f}$ and $F_2(x) = C_{m, f}$, and the multi-objective constrained optimization problem can be expressed as

$$\min_{l \leq x \leq u} C_d, \quad \max_{l \leq x \leq u} C_m,$$

subject to

$$C_l(x) = 0.824,$$

$$A(x) \geq A_{\text{baseline}},$$

where $A_{\text{baseline}}$ is the cross-sectional area of the baseline RAE282 airfoil.

2. Design Variables

The airfoil shape is controlled using the B-spline parameterization approach described in Amrit et al. [33]. Figure 4-5 (a) shows eight control points, four each on the top and bottom surfaces that can move in vertical direction. The leading and trailing edge end points of the airfoil are fixed. The design variable vector is $x = [Z_u; Z_l]^T = [z_{u1} \ldots z_{u4}; z_{l1} \ldots z_{l4}]^T$, where $z_{ui}$ and $z_{li}$ are the vertical coordinates of the control points of the upper and lower surfaces, respectively, and $i = 1, \ldots, 4$, denotes the order of the control points from the leading edge towards the trailing edge. The $(x/c)$ locations of the eight control points (eight design variables) are fixed and are based on a fit to the RAE 2822 airfoil shape. Specifically, $X = [X_u; X_l]^T = [(x/c)_{u1} \ldots (x/c)_{u4}; (x/c)_{l1} \ldots (x/c)_{l4}]^T = [0.0 0.15 0.45 0.80 0.0 0.35 0.60 0.90]^T$. The initial design variable vector
is $x = [0.0175 \ 0.0498 \ 0.0688 \ 0.0406 \ -0.0291 \ -0.0679 \ -0.0384 \ 0.0054]^T$. The lower bound of $x$ is set as $l = [0.015 \ 0.015 \ 0.015 \ 0.015 \ -0.08 \ -0.08 \ -0.08 \ -0.01]^T$, and the upper bound is set as $u = [0.08 \ 0.08 \ 0.08 \ 0.08 \ -0.01 \ -0.015 \ -0.015 \ 0.01]^T$.

3. High-Fidelity Viscous Aerodynamics Model

The high-fidelity aerodynamic model ($f$) solves the steady compressible Reynolds-Averaged Navier-Stokes (RANS) equations with the Spalart-Allmaras turbulent model [47] using the SU$^2$ implicit density-based solver [46]. The second-order JST scheme [48] is used to calculate the convective flux along with one multi-grid level to accelerate the solution. The turbulent variables are convected using a first-order scalar upwind method. The flow solver convergence criterion is the one that occurs first of the two: (i) the change in the drag coefficient value over the last 100 iterations is less than $10^{-5}$, or (ii) a maximum number of iterations of 20,000 is met.

The computational grid is generated using the hyperbolic C-mesh of Kinsey and Barth [49] (see Fig. 5 (b)) with the farfield set 100 chord lengths from the airfoil surface. To have the wall $y^+$ values within reasonable values (i.e. $y^+ < 5$) around the airfoil surface, the distance from the airfoil surface to the first node is $10^{-5}c$, where $c$ is the airfoil chord length. The grid points are clustered at the trailing edge and the leading edge of the airfoil with the density controlled by the number of points in the streamwise direction, and in the direction, normal to airfoil surface.

Table 4-1 gives the results of a grid convergence study using the RAE 2822 airfoil at $M_\infty = 0.734$, $C_l = 0.824$, and $Re_\infty = 6.5 \times 10^6$. The constant lift condition is achieved by internally changing the angle of attack within the flow solver. The simulation time presented in Table 4-1 is the overall time to compute the constant lift condition using 32 processors. Table 4-1 indicates that Mesh 4 is the finest and the most accurate. However, the difference in coefficient
Figure 4-5. Airfoil models: (a) shape parameterization, (b) hyperbolic C-mesh.

of drag between meshes 3 and 4 is negligible compared to the difference in overall simulation time. Hence, Mesh 3 was chosen as the high-fidelity model (f) for this work.

4. Low-Fidelity Viscous Aerodynamics Model

The model set up for low-fidelity model is same as that of high-fidelity model (f), with the grid size being far less than that of high-fidelity model. As shown in Table 4-1, we use Mesh 1 for the low-fidelity model c. The low-fidelity model convergence criteria are set with the following values only.

(i) change in the drag coefficient value over the last 100 iterations is less than $10^{-4}$, or (ii) the maximum number of iterations is set to 5,000.

Table 4-1. Grid convergence study for the baseline shape.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Number of Elements</th>
<th>Lift Counts (l.c.)</th>
<th>Drag Counts (d.c.)</th>
<th>Simulation Time* (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9,836</td>
<td>0.824</td>
<td>324.6</td>
<td>3.1</td>
</tr>
<tr>
<td>2</td>
<td>38,876</td>
<td>0.824</td>
<td>221.5</td>
<td>8.8</td>
</tr>
<tr>
<td>3</td>
<td>154,556</td>
<td>0.824</td>
<td>204.8</td>
<td>34.0</td>
</tr>
<tr>
<td>4</td>
<td>616,316</td>
<td>0.824</td>
<td>203.0</td>
<td>152.6</td>
</tr>
</tbody>
</table>

* Computed on a high-performance cluster with 32 processors. Flow solution
5. Single-Objective Optimization (SOO) Results

Initial designs corresponding to the best possible value of the first objective (minimum drag coefficient) subjected to two different nonlinear pitching moments (target pitching moment values) are obtained in the first step of each algorithm using the output SM based SOO algorithm [40] shown in Table 4-2 as SOO 1 and SOO 2. While MOEA and SDP use both the SOO points as their starting points, Point-by-point algorithm uses only SOO 1 point. The runs are performed at $M_{\infty} = 0.734$, $A_{\text{baseline}} = 0.0779$ and at $C_l = 0.824$.

6. The Pareto Front

The design space exploration is performed using all the algorithms described in Section II. Due to the computational expense of the CFD models, instead of exploring the entire Pareto, i.e. in between the optimal points of the two objective functions (the extreme points), only a part of the Pareto front is explored. Subsequent designs along the Pareto were obtained using the algorithms described in Section II and the process is terminated when the Pareto front is traversed in between the two initial designs. The algorithm is run to obtain a Pareto in between $C_m = -0.074$ and $C_m = -0.11$ (selected here for illustration purposes).

Figure 4-6 (a) shows the final optimal solution set (Pareto front) obtained in between the SOO points. A zoomed-in plot of the final Pareto front is represented in Fig. 4-6 (b). Two points designated as Point 1 and Point 2 on the Pareto optimal set were selected to be compared with the baseline design. Figures 4-7 (a) and (b) show comparisons of all the airfoil shape designs and the pressure coefficient distributions for the selected points and the baseline design. There is a significant difference in the pressure coefficient distribution of the selected points compared to the baseline with the former having a considerable reduction in shock strength. Further, the Mach contour plots in Fig. 4-8 show Point 1 with higher shock strength leading to more drag compared to Point 2. This aligns with the fact that to obtain a lower drag,
there will be a decrease in pitching moment as shown in Fig. 4-6. Note that the main reason behind irregularity in the Pareto front obtained from SDP and Point-by-point algorithms (as shown in Fig. 4-6) is that each CFD simulation is converged to within 1 drag count, and the terminating conditions for reaching a target point in both the algorithms is set to 1 drag count. All the three Pareto fronts are within 1 drag count error.

Table 4-2. *Single objective optimization results for transonic airfoil design case.*

<table>
<thead>
<tr>
<th>Parameter/method</th>
<th>Baseline</th>
<th>SOO 1</th>
<th>SOO 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_l$ (l.c.)</td>
<td>82.35</td>
<td>82.39</td>
<td>82.39</td>
</tr>
<tr>
<td>$C_d$ (d.c.)</td>
<td>203.80</td>
<td>116.9</td>
<td>121.8</td>
</tr>
<tr>
<td>$C_{m,c/4}$</td>
<td>-0.0905</td>
<td>-0.1023</td>
<td>-0.0736</td>
</tr>
<tr>
<td>$A$</td>
<td>0.0779</td>
<td>0.0779</td>
<td>0.0779</td>
</tr>
<tr>
<td>$N_c$</td>
<td>-</td>
<td>550</td>
<td>499</td>
</tr>
<tr>
<td>$N_f$</td>
<td>-</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>CPU Time (hours)</td>
<td>-</td>
<td>14</td>
<td>13.55</td>
</tr>
</tbody>
</table>

Figure 4-6. *The final Pareto fronts from all three algorithms at $M_{\infty} = 0.734$, $C_l = 0.824$ and $Re_{\infty} = 6.5 \times 10^6$ (a) final refined Pareto (b) zoomed in view of the same Pareto front.*
Figure 4-7. **MOO results showing (a) the airfoil shapes, and (b) the pressure distributions at** $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$ for different points along the Pareto.

![Airfoil Shapes](image1)

![Pressure Distributions](image2)

Figure 4-8. **Mach contours at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$ of (a) Point 1, (b) Point 2.**

The total cost in terms of CPU time to obtain the final Pareto on a HPC with 32 processors is shown in Table 4-3 which includes the cost of obtaining the starting points. The MOEA algorithm utilizes a fixed number of high-fidelity samples every iteration until convergence for Pareto-front refinement. For transonic airfoil design exploration the number of MOO iterations...
need to be increased as compared to simple analytical problems (shown in Section III.A) to explore the nonlinear design space accurately. Due to this, the MOEA-based MOO approach requires many high-fidelity model evaluations when compared to other two algorithms. Apart from that, the MOEA-based algorithm requires the two end points of the Pareto front to be explored and, hence, needs to solve two SOO problems. Thus, the high cost of obtaining the starting points and more dependency on the high-fidelity model evaluations for the refinement step render the MOEA-based algorithm to be the more expensive than the other two algorithms. The full Pareto front, however, is produced by the MOEA-based algorithm.

The SDP algorithm requires the two end points of the Pareto front to be explored as its starting points as the MOEA-based algorithm. Unlike MOEA, however, the SDP algorithm mostly relies on the low-fidelity model to obtain an initial Pareto front. Furthermore, two refinement steps require 35 high-fidelity model evaluations to obtain 20 optimal points along the Pareto front. The total computational time to obtain the entire Pareto was approximately 50 hrs.

The point-by-point algorithm is found to be the cheapest among the three algorithms requiring only 35 hrs of computational time to obtain 20 Pareto optimal solutions. The high exploitation of the low-fidelity model and utilization of only one starting point are the reasons for it to be the most efficient algorithm.

Table 4-3. Cost of each multi-fidelity MOO algorithm for the transonic airfoil design case.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Name</th>
<th>$N_c$</th>
<th>$N_f$</th>
<th>Total time (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SA-MOEA</td>
<td>50</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>SDP</td>
<td>350</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>Point-by-Point</td>
<td>462</td>
<td>22</td>
<td>35</td>
</tr>
</tbody>
</table>
C. Subsonic Rectangular Wing Design

This section presents results of applications of the surrogate-assisted and multi-fidelity algorithms to the MOO of a rectangular subsonic wing shapes. We give the problem formulation, design variable formulation, and results for the SOO and the Pareto front.

1. Problem Description

The objective is to obtain the best possible trade-offs between the drag coefficient ($C_D$) and the pitching moment coefficient ($C_{Mx}$) of a rectangular wing (see Fig. 4-9) in an inviscid subsonic flow at a Mach number of $M_\infty = 0.5$ and a fixed lift coefficient of 0.2625 ($C_L$). The multi-objective constrained optimization problem is written as

$$\min_{l \leq x \leq u} C_D, \min_{l \leq x \leq u} C_{Mx}$$

subject to

$$C_L = 0.2625,$$

$$100 \left( \frac{|S_0 - S_l|}{S_0} \right) - 1\% \leq 0,$$

$$100 \left( \frac{V_0 - V}{V_0} \right) \leq 0,$$

and

$$-3^\circ - \alpha \leq 0,$$

$$\alpha - 6^\circ \leq 0.$$

The designable parameters are the wing semi-span ($b/2$), as well as the span-wise distributions of twist ($\gamma$), chord ($c$), and dihedral ($zc/a$). Although the distributions of sweep and airfoil shape can change in the problem, they are fixed to zero center-chord sweep and the NACA 0012 airfoil shape for this case. Six nonlinear constraints are enforced (one equality and five inequality). The equality constraint ensures a lift coefficient of 0.2625, and is enforced implicitly within the flow solver. The first inequality constraint ensures that planform area remains within 1% of the baseline value of 3.06, while the second ensures that the internal wing volume is greater than or equal to that of the
baseline. Lastly, the angle of attack is limited to a range of \(-3^\circ\) to \(6^\circ\). Here, we represent the volume, area, and moment constraints as percentages of the reference values, which, respectively, are the minimum volume \((V_0 = 0.24818)\), target planform area \((S_0 = 3.04110)\), and the maximum moment coefficient value \((C_{M_{\text{max}}} = 0.1069)\). The angle of attack is denoted by \(\alpha\).

2. Design Variables

B-spline parameterization [35] is used to control the surface of the rectangular wing. There are 11 designable parameters: the wing semi span \((b/2)\), and distributions of the wing chord length \((c)\), \(z_{c/4}\) (vertical) coordinates at the quarter-chord, and the twist \((\gamma)\) as shown in Fig. 4-10. Specifically, the designable vector is written as \(\mathbf{x} = [b/2 \ c \ z_{c/4} \ \gamma]^T = [b/2 \ c_r \ c_{c,y} \ c_{c,z} \ c_t \ (z_{c/4})_{c,y} \ (z_{c/4})_{c,z} \ (z_{c/4})_{t,y} \ (z_{c/4})_{t,z} \ \gamma]^T\), where the subscripts \(r\), \(c\), \(y\), \(z\), and \(t\) denote the root control point, center control point, horizontal direction, vertical direction, and the tip control point. The horizontal coordinates of each center control point can vary between 0.2 and 0.8. For chord, the vertical coordinates of the root and tip control points can vary between 0.45 and 1.55, while the center point can vary between 0.1 and 1.9. For \(z\)-coordinates, the tip control point can vary between...
Wing shape parameterization using B-spline curves for chord length, wing quarter-chord height and twist.

-0.45 and 0.45 vertically, while this range for the center point is -0.8 to 0.8. Finally, for twist, the tip control point can vary between -3.12 and 3.12 vertically, while the center point can vary from -5 to 5. Nonlinear constraints are enforced to ensure that each of these distributions fall within the ranges [0.45 1.55], [-0.45 0.45], and [-3.12 3.12] for chord, z-coordinates, and twist angle, respectively. The initial design variable vector is set $x = [3.06 0.5 1 1 1 0.5 0 0 0.5 0 0]^T$. The lower bound of $x$ is set as $l = [2.46 0.20 0.45 0.10 0.45 0.20 -0.80 -0.45 0.20 -5.00 -3.12]^T$, and the upper bound is set as $u = [3.67 0.80 1.55 1.90 1.55 0.80 0.80 0.45 0.80 5.00 3.12]^T$.

3. Computational Model

In order to evaluate the loads on a particular wing design, the aerodynamics model executes several sequential steps. First, the design variables are converted to the span-wise geometric distributions (e.g. twist, chord, sectional shape) using B-spline parameterization. Then, using the resulting geometry, an Engineering Sketch Pad (ESP) [50] script is executed to produce the solid model. This CAD model is then used by a Pointwise script to generate a structured mesh with an O-type topology as shown in Fig. 11. Then, SU2 version 5.0.0 [46] is used to solve the Euler equations on the grid at the specified Mach number. Each design evaluation
adjusts the angle of attack to produce the prescribed lift coefficient implicitly within the flow solver.

Before initiating the optimization process, a grid convergence study was carried out for the baseline design, as shown in Fig. 4-12 (a). A set of refinement factors, ranging from 0.25 to 1.75, in increments of 0.25, was chosen to increase the mesh size in the grid convergence study. Each refinement factor value is utilized to divide the off-wall spacing, while multiplying the numbers of cells (along wing, around airfoil, between wing and far-field). The resulting meshes had numbers of cells ranging from approximately 25,000 to 9M cells as shown in Fig. 4-12 (a). A low-fidelity model, i.e. a coarse discretized mesh, is selected having an off-wall spacing of 0.004, 100 cells along the span, approximately 170 cells around the airfoil sections, and 100 cells between the wing and far-field surface, which is located at a radius of 50. The low-fidelity model with approximately 1.7M cells and the high-fidelity model with approximately 9 million cells consume around 0.5 hrs and 2.5 hrs respectively per simulation with 32 processors on a high-performance computing cluster as shown in Fig. 4-12 (b).

Figure 4-11. *Sample mesh used for the computational fluid dynamics simulations.*
4. Single-Objective Optimization (SOO) Results

The two SOO problems, i.e. to minimize the drag coefficient ($C_D$) and to minimize the pitching moment coefficient ($C_M$), are solved using the SM algorithm [35] and results are shown in Table 4-4. The cost in terms of CPU time for each of the SOO process is approximately 9 hours on a HPC with 32 processors. It should be noted here that while the MOEA-based and SDP algorithms use both the SOO points as their starting points, the point-by-point algorithm uses only SOO 1 point as its starting point.

![Figure 4-12](image)

Figure 4-12. Grid convergence results for the baseline design at $M_x = 0.5$ and $C_L = 0.2625$: (a) drag coefficient versus number of mesh cells, (b) simulation run time.

Table 4-4. Single-objective optimization results for the subsonic wing design.

<table>
<thead>
<tr>
<th>Parameter/Method</th>
<th>Baseline</th>
<th>SOO 1</th>
<th>SOO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_D$ (d.c.)</td>
<td>40.78</td>
<td>32.14</td>
<td>42.41</td>
</tr>
<tr>
<td>$C_{Mx}$</td>
<td>0.11369</td>
<td>0.1194</td>
<td>0.0990</td>
</tr>
<tr>
<td>$N_f$</td>
<td>-</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>$N_c$</td>
<td>-</td>
<td>18,403</td>
<td>18,000</td>
</tr>
<tr>
<td>$t_c$ (hr)</td>
<td>-</td>
<td>2.95</td>
<td>2.8</td>
</tr>
<tr>
<td>$t_f$ (hr)</td>
<td>-</td>
<td>6.46</td>
<td>6.4</td>
</tr>
<tr>
<td>$t_{tot}$ (hr)</td>
<td>-</td>
<td>9.41</td>
<td>9.2</td>
</tr>
<tr>
<td>Termination criterion</td>
<td>-</td>
<td>Step size</td>
<td>Step size</td>
</tr>
</tbody>
</table>
5. Pareto-Front

The Pareto front for the multi-objective aerodynamic design problem is explored by executing all three multi-fidelity algorithms. A target Pareto front is identified to be explored and is determined to be +/- 10% of the pitching moment value of SOO point (shown in Table 4-4). Subsequent Pareto optimal designs are obtained using each algorithm discussed in Section II and are iterated until their respective terminating conditions are met.

Figure 4-13 (a) shows a comparison of Pareto fronts obtained by executing the algorithms. It can be seen that the Pareto fronts are comparable. Figure 4-13 (b) is a zoomed in view of the same Pareto with all the three Pareto fronts within less than one drag count (which is within the numerical accuracy level of the high-fidelity model). Two designs, designated as Point 1 and Point 2 on Fig. 4-13 (b), are selected from the Pareto for comparison purposes.

Figure 4-14 shows the design variable distributions for the baseline and the designated points from the Pareto fronts. It indicates a relatively longer chord length along the wing semi-span and higher washout for Point 1 compared to Point 2. Table 4-5 gives the numerical values of the wing design parameters for the baseline and both the selected points from the Pareto front at non-dimensionalized wing semi-span locations $\eta = 0.02, 0.5, 0.75$ and 1 (here, $\eta \equiv y/(b/2)$). The results indicate that both Points 1 and 2 have a larger wing semi-span compared to that of the baseline with Point 2 having the largest wing semi-span of 3.40 m. Point 1 has a larger washout than the baseline and Point 2. The maximum washout of Point 1 is -3.12 deg., whereas Point 1 has a maximum of -1.01 deg.

A comparison of the wing shapes and surface pressure contours of the designs of Point 1 and Point 2 are shown in Fig. 4-15. Figures 4-15 (a) and (b) show the planform view, and Figs. 4-15 (c) and (d) show the side view. It can be seen that the tip of the wing for the design
designed by Point 2 curves upward, which is a shape similar to a winglet and has the function of reducing lift-induced drag. The pressure coefficient distribution of the designated points along the Pareto is plotted along four wing stations as shown in Fig. 4-16. It can be observed in Figs. 4-16 (a) and (b) that although designs Point 1 and Point 2 have almost the same root chord length, Point 1 has lower pressure coefficient value at the leading edge due to a positive dihedral angle whereas Point 2 has an anhedral angle (see Table 4-5). At $\eta = 0.75$, although Point 1 has a higher positive dihedral angle, the pressure coefficient distributions of both points are comparable as shown in Fig. 4-16 (c). This is because of the relatively lower twist value for Point 1 which compensates the pressure effects. Further, Fig. 4-16 (d) shows that at $\eta = 1$, Point 2 has lower pressure coefficient value at the leading edge than Point 1. This is due to a relatively higher dihedral angle for Point 2 and much higher negative twist for Point 1 (cf. Table 5). Figure 4-17 shows a comparison of the wing section lift coefficient distribution of Points 1 and 2. Point 1 has a higher inboard section lift coefficient than Point 2, whereas the outboard section lift coefficients are comparable. Point 2 has a larger semi-span than Point 1 (a difference of 0.26 m or a relative difference of 8.3%). Since the chord lengths of the wings are comparable, the higher pitching moment coefficient of Point 2 is therefore mainly due to the larger wing span.

The best possible trade-offs between the drag coefficient and the pitching moment coefficient for this design problem can be characterized by the wing span and the wing shape and twist near the wing tip. In particular, adding length to the wing span, increasing the curvature of the wing tip with a moderate amount of twist will yield design of low drag coefficient and high pitching moment coefficient, whereas reducing the wing span and wing
tip curvature and increasing the tip twist will yield designs of high drag coefficient and low pitching moment coefficient. For these designs, the wing chord distribution is trapezoidal.

A comparison of number of low-fidelity model evaluations ($N_c$), number of high-fidelity model evaluations ($N_f$), and the total computational cost of each algorithm to obtain 20 optimal solutions is reflected in Table 4-6. For this problem, the point-by-point algorithm is the fastest with approximately 40 hrs of computational time (500 low-fidelity and 30 high-fidelity model evaluations) and the MOEA-based algorithm is the most expensive with approximately 80 hrs of computational time (50 low-fidelity and 60 high-fidelity model evaluations) on a high-performance computing cluster with 32 processors. In other words, the point-by-point requires around 50% less time than the MOEA-based algorithm. Furthermore, the point-by-point algorithm is around 27% more efficient than the SDP algorithm for this case. The reasons for the point-by-point algorithm being more efficient are, as in transonic airfoil design case, that it requires only one SOO point and exploits the low-fidelity model to obtain the Pareto front at the high-fidelity model level, whereas the MOEA-based and SDP algorithm both require two SOO points.

Figure 4-13. Final Pareto front obtained from each algorithm: (a) optimum solutions and (b) zoomed-in plot.
Figure 4-14. Wing shape parameter comparison for Baseline, Point 1 and Point 2 designs for (a) chord length, (b) wing quarter-chord height and (c) twist.

Table 4-5. Wing shape parameter comparison for Baseline, Point 1 and Point 2 designs at multiple wing span stations.

<table>
<thead>
<tr>
<th>Designs</th>
<th>$\eta$ [m]</th>
<th>$b/2$ [m]</th>
<th>$c$</th>
<th>$zc/4$ [m]</th>
<th>$\gamma$ [deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.02</td>
<td>3.06</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Point 1</td>
<td>0.02</td>
<td>3.14</td>
<td>1.41</td>
<td>0.01</td>
<td>-0.02</td>
</tr>
<tr>
<td>Point 2</td>
<td>0.02</td>
<td>3.40</td>
<td>1.40</td>
<td>-0.01</td>
<td>0</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.5</td>
<td>3.06</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Point 1</td>
<td>0.5</td>
<td>3.14</td>
<td>0.97</td>
<td>0.12</td>
<td>-1.33</td>
</tr>
<tr>
<td>Point 2</td>
<td>0.5</td>
<td>3.40</td>
<td>0.89</td>
<td>-0.10</td>
<td>-0.40</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.75</td>
<td>3.06</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Point 1</td>
<td>0.75</td>
<td>3.14</td>
<td>0.73</td>
<td>0.12</td>
<td>-2.18</td>
</tr>
<tr>
<td>Point 2</td>
<td>0.75</td>
<td>3.40</td>
<td>0.67</td>
<td>-0.01</td>
<td>-0.69</td>
</tr>
<tr>
<td>Baseline</td>
<td>1</td>
<td>3.06</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Point 1</td>
<td>1</td>
<td>3.14</td>
<td>0.46</td>
<td>0.02</td>
<td>-3.12</td>
</tr>
<tr>
<td>Point 2</td>
<td>1</td>
<td>3.40</td>
<td>0.45</td>
<td>0.35</td>
<td>-1.01</td>
</tr>
</tbody>
</table>
Figure 4-15. Pressure contours at $M_\infty = 0.5$ and $C_l = 0.2625$ (a) Point 1 (top view), (b) Point 2 (top view), (c) Point 1 (side view) and (d) Point 2 (side view).
Figure 4-16. MOO results showing the pressure distributions of two points on the Pareto at $M_\infty = 0.5$ and $C_L = 0.2625$ and at (a) $\eta = 0.02$, (b) $\eta = 0.5$, (c) $\eta = 0.75$ and (d) $\eta = 1$. 
Figure 4-17. A comparison of the wing lift distributions for the Point 1 and Point 2 designs.

Table 4-6. Cost of each MOO algorithm for subsonic wing design.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Name</th>
<th>Ne</th>
<th>Nf</th>
<th>Total time (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SA-MOEA</td>
<td>50</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>SDP</td>
<td>375</td>
<td>40</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Point-by-Point</td>
<td>500</td>
<td>30</td>
<td>40</td>
</tr>
</tbody>
</table>

IV. Conclusion

This paper presents results of applications of three recently developed multi-objective optimization algorithms to simulation-based aerodynamic design of transonic airfoils and subsonic wings. The surrogate-assisted multi-objective evolutionary algorithm (SA-MOEA) utilizes a combination of design space reduction, surrogate modeling methods, and an evolutionary search algorithm. The other two algorithms use multi-fidelity methods, and Pareto set identification techniques, namely, the sequential domain patching (SDP) and the point-by-
point method, as well as local search with gradient-based techniques. The results of applications show that the algorithms obtain comparable Pareto front solutions. In terms of computational cost, the multi-fidelity point-by-point algorithm outperforms both the SA-MOEA and the multi-fidelity SDP algorithm for the cases considered.

A distinct advantage of the SA-MOEA algorithm over the other two is that it can yield the entire Pareto front in one algorithm run. This desirable characteristic, however, comes at an additional computational cost. The advantages of the multi-fidelity SDP and point-by-point algorithms over SA-MOEA is that a part of the Pareto front can be estimated based on the available computational budget. The higher computational efficiency of the multi-fidelity point-by-point algorithm is due to the fact that it requires only one starting point on the Pareto front, whereas the SA-MOEA and multi-fidelity SDP algorithm both require two starting points. Furthermore, the multi-fidelity point-by-point and SDP algorithms are more efficient than the SA-MOEA algorithm due to their capability of exploiting fast low-fidelity models. It should be noted that the overall computational cost of the multi-fidelity point-by-point and SDP algorithms is, however, limited by the efficiency of the search algorithm used to determine the initial points on the Pareto front.

Future work will investigate how the computational cost grows with the complexity of the aerodynamic design problem with respect to the dimensionality of the problem and the degrees of freedom of the simulations. Furthermore, future work will investigate the robustness and reliability of the multi-fidelity point-by-point and SDP algorithms with respect to the setup of the low-fidelity models. Lastly, extensions of the multi-fidelity point-by-point and SDP algorithms for aerodynamic design exploration with more than two objectives.
References


CHAPTER 5. CONCLUSION

In this chapter, we summarize the main contributions of this doctoral thesis and attempt to answer the research questions posed in Chapter 1. In particular, we discuss role of multi-fidelity aerodynamic modeling, the effectiveness of each of the proposed multi-fidelity aerodynamic multi-objective optimization (MOO) algorithms, comparison of the algorithms, and future research directions.

I. Main Contributions

A. Multi-Fidelity Aerodynamic Modeling

The multi-fidelity models utilized in the MOO process involve a combination of a computationally faster low-fidelity model \( c \), and an accurate high-fidelity model \( f \). In particular, for the aerodynamic problems discussed in this work, \( c \) and \( f \) are based on coarse and fine discretization computational fluid dynamics (CFD) simulations, respectively. In this work, a grid convergence study is used to estimate the accuracy and computational cost for each level of model. The multi-fidelity aerodynamic models are then constructed using output-space mapping and utilized to perform constrained single-objective optimizations (SOO) within a trust-region framework. The proposed multi-fidelity MOO approaches use the trust-region-based multi-fidelity SOO to reach the Pareto cheaply.

B. Multi-Fidelity Aerodynamic Design Exploration Using Sequential Domain Patching

The multi-fidelity sequential domain patching (SDP) MOO algorithm involves solving SOO problems to obtain the extremes of the Pareto to be explored, and, subsequently, obtain an approximation of the Pareto front by exploring patches setup using the fast low-fidelity model. More specifically, the Pareto-optimal solutions are explored within a patch using the low-fidelity aerodynamics model \( c \) at the vicinity of the starting points, and the process is continued
sequentially until the entire Pareto is traversed and producing a Pareto front that is approximate. The SDP exploration is performed simultaneously from each extreme end of the Pareto front. Then, a refinement of the initial Pareto front to the level of the high-fidelity model is performed utilizing local response surface approximations (RSAs) of the low-fidelity model data and adaptive corrections based on high-fidelity aerodynamics model data.

C. Multi-Fidelity Aerodynamic Design Exploration Using the Point-by-Point Method

The multi-fidelity point-by-point MOO algorithm involves solving one SOO problem to obtain a single point on the Pareto front to be explored, and, subsequently, exploring the Pareto-optimal solutions point-by-point through a series of constrained optimization problems. The search for the optimal solutions are performed on a local RSA model constructed on low-fidelity model data and refined using one high-fidelity model evaluation per iteration through a linear output-space mapping correction.

D. A Comparison of the Proposed Approaches

In this work, the proposed multi-fidelity aerodynamic exploration approaches are compared using several analytical problems, a transonic airfoil design problem, and a subsonic wing design problem. Furthermore, the proposed approaches are benchmarked against a surrogate-assisted multi-objective evolutionary algorithm (SA-MOEA) which uses design space reduction, surrogate modeling, and evolutionary search algorithms.

The results indicate that the three approaches produced comparable Pareto fronts for all the design problems. In particular, for the aerodynamic design exploration problems the three approaches yield the Pareto fronts within the level of accuracy of the high-fidelity models. The cost of obtaining the starting points in the case of the SA-MOEA and multi-fidelity SDP algorithms is higher than that of the multi-fidelity point-by-point algorithm. Moreover, the dependency on the high-fidelity model data is strongest in the case of SA-MOEA, weaker for
the multi-fidelity SDP algorithm, and the weakest for the multi-fidelity point-by-point algorithm. Consequently, the multi-fidelity point-by-point algorithm is the most efficient algorithm for the aerodynamic design exploration cases considered in this work. The multi-fidelity SDP algorithm, however, seems to be slightly more accurate than the multi-fidelity point-by-point algorithm because it utilizes information on the extremes of the Pareto although this characteristic comes at an additional computational cost. Additionally, it is shown that the computational expense of the SA-MOEA algorithm faster with the complexity of the problem than the proposed multi-fidelity approaches.

E. Answers to the Research Questions

The overall research goal of this work is to investigate alternatives of multi-fidelity multi-objective algorithms to accelerate simulation-based aerodynamic design exploration. To achieve this goal, the work presented in Chapters 2, 3, and 4 (which are based on the journal articles produced in this work) seeks to answer the research questions posed in Chapter 1 (cf. Section 1.2). Based on the presented work, the following conclusions are drawn.

- **Research Question 1**: How can multi-fidelity methods and surrogate models be used to accelerate the PDE-constrained MOO to enable fast aerodynamic design exploration? What is the magnitude of the computational acceleration? Are the estimated best possible trade-offs using the multi-fidelity methods at the level of the high-fidelity model accuracy?

  In this work, two multi-fidelity MOO algorithms for aerodynamic design exploration, the multi-fidelity sequential domain patching (SDP) (Chapter 2) and the multi-fidelity point-by-point algorithm (Chapter 3), are proposed. Both algorithms utilize multi-fidelity methods to expedite aerodynamic design exploration. Compared to the
benchmark method, a surrogate-assisted multi-objective evolutionary algorithm (SA-MOEs), the proposed multi-fidelity MOO algorithms with the Pareto set identification can be up to 50% more efficient while still yielding estimates of the Pareto fronts at the high-fidelity model accuracy.

- **Research Question 2:** If the designer has a budget, is it possible to obtain only a part of the Pareto, or in other words, is it possible to obtain few optimal solutions in the vicinity of a target design which are non-commensurate using multi-fidelity methods? The proposed algorithms can obtain the entire or only a part of the Pareto front. Once a point (or points) on the Pareto front is obtained, the algorithms can obtain several optimal solutions in its vicinity that are non-commensurable.

- **Research Question 3:** Is it possible to develop a strategy that can efficiently perform multi-fidelity MOO on computationally complex aerodynamic problems that have nonlinear design and feature spaces? The proposed multi-fidelity MOO algorithms are evaluated by comparing the solutions and the computational cost with the benchmark SA-MOEA using problems of increasing complexity (i.e., problems from simple analytical functions to transonic airfoil design to subsonic wing design; where the number of design variables and the number of simulation degrees of freedom increase significantly). The results show that, although the nonlinearity of the design and feature spaces increases, the proposed approaches yield comparable results as the benchmark method at a lower computational cost.
• **Research Question 4:** *Is it possible to develop a strategy that is robust such that the Pareto front estimated with multi-fidelity methods is always within a reasonable error margin irrespective of the number of MOO trials?*

The robustness (in terms of yielding the same or comparable results using different initial designs) of the multi-fidelity point-by-point algorithm is established by obtaining the same Pareto front with different initial points within an error margin which depends on the accuracy level of the high-fidelity model computational mesh.

• **Research Question 5:** *Is it possible to evaluate the scalability of the multi-fidelity strategy or in other words, is it possible to measure the computational cost associated with the increase in design space dimensionality and simulation degrees of freedom?*

The scalability of the multi-fidelity point-by-point algorithm is evaluated by performing several MOO studies on the transonic airfoil design case while increasing the number of dimensions from eight up to twelve. The results indicate that with increasing problem dimensionality the computational cost of the multi-fidelity point-by-point algorithm increases slowly in terms of high-fidelity model evaluations but quickly in terms of the low-fidelity model evaluations.

**II. Suggestions for Future Work**

Future work should investigate further how the computational cost grows with the complexity of the aerodynamic design problem with respect to the dimensionality of the problem and the degrees of freedom of the simulations. In particular, future work should focus on approaches to alleviate the computational burden of searching for initial point(s) on the Pareto front, and, subsequently, traversing quickly along the Pareto front. The use of adjoint sensitivity information can be exploited. Future work should also investigate the robustness and reliability
of the multi-fidelity point-by-point and SDP algorithms with respect to the setup of the low-fidelity models. Lastly, extensions of the multi-fidelity point-by-point and SDP algorithms for aerodynamic design exploration with more than two objectives. To handle problems of high complexity, investigations on including principal component analysis technique in the multi-fidelity MOO procedures is recommended.