MULTI-OBJECTIVE EFFICIENT PARAMETRIC OPTIMIZATION

A Dissertation

by

JONATHAN M. WEAVER-ROSEN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Richard J. Malak Jr.
Committee Members,	Douglas Allaire
	Darren Hartl
	Daniel McAdams
Head of Department,	Andreas Polycarpou

August 2021

Major Subject: Mechanical Engineering

ABSTRACT

Parametric optimization is the process of solving an optimization problem as a function of currently unknown or changing variables known as parameters. Parametric optimization methods have been shown to benefit engineering design and optimal morphing applications through its specialized problem formulation and specialized algorithms. Despite its benefits to engineering design, existing parametric optimization algorithms (similar to many optimization algorithms) can be inefficient when design analyses are expensive. Since many engineering design problems involve some level of expensive analysis, a more efficient parametric optimization algorithm is needed.

Therefore, the multi-objective efficient parametric optimization (MO-EPO) algorithm is developed to allow for the efficient optimization of problems with multiple parameters and objectives. This technique relies on the new parametric hypervolume indicator (pHVI) which measures the space dominated by a given set of data involving both objectives and parameters. The pHVI benefits parametric optimization by enabling the comparison of optimization results, enabling the visualization and detection of optimization convergence, and providing information for an optimization algorithm. MO-EPO uses response surface models of expensive functions to find and evaluate a designs expected to improve the solution and/or models. With new information, response surface models are updated and the process is repeated. “Improvement” is measured by the pHVI metric allowing for the consideration of any number of objectives and parameters.

The novel MO-EPO algorithm is demonstrated on a number of analytical benchmarking problems and two distinct morphing applications with various numbers of objectives and parameters. In each case, MO-EPO is shown to find solutions that are as good as or better than those found from the existing P3GA (i.e., equal or greater pHVI value) when the number of design evaluations is limited. Both the pHVI metric and the MO-EPO algorithm are significant contributions to parametric optimization methodology and engineering design.

DEDICATION

For my wife, my son, my parents, and family.

ACKNOWLEDGMENTS

I would like to thank my advisor and mentor Dr. Richard Malak for his guidance and support throughout my graduate studies. I thank my family for their unwavering support and encouragement. I also wish to thank the countless people who supported me during my studies including colleagues, friends, teachers, and mentors.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Richard J. Malak Jr. [advisor], Dr. Douglas Allaire, and Dr. Daniel McAdams of the Department of Mechanical Engineering and Dr. Darren Hartl of the Department of Aerospace Engineering.

The analyses depicted in Chapter 3 were conducted in collaboration with Pedro Leal of the Department of Aerospace Engineering at Texas A&M University. The parametric problem formulated in Chapter 4 was heavily inspired by Dr. Darren Hartl. The optimization results presented in Chapter 8 were collected in part by Yutaka Terada. All other work conducted for the dissertation was completed by the student independently.

Funding Sources

These graduate studies have been supported by the NASA *University Leadership Initiative* (ULI) program under federal award number NNX17AJ96A, titled Adaptive Aerostructures for Revolutionary Civil Supersonic Transportation in addition to fellowships from the Dwight Look College of Engineering and the J. Mike Walker '66 Department of Mechanical Engineering at Texas A&M University.

NOMENCLATURE

$\boldsymbol{\theta}$	Parameter vector
Φ	Standard normal cumulative distribution function
ϕ	Standard normal probability distribution function
ϕ_{hc}	Hypercone angle (for hypercone heuristic)
$\boldsymbol{x}$	Optimization variable vector
$\mathbf{J}$	Objective function vector
CFD	Computational Fluid Dynamics
EGO	Efficient global optimization
EPO	Efficient parametric optimization
FEA	Finite element analysis
HCH	Hypercone heuristic
HVI	Hypervolume indicator
MO-EPO	Multi-objective efficient parametric optimization
OML	Outer mold line
P3GA	Predictive parameterized Pareto genetic algorithm
pHVI	Parametric hypervolume indicator
PPD	Parameterized Pareto dominance
SVDD	Support vector domain description

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

CONTRIBUTORS AND FUNDING SOURCES v

NOMENCLATURE vi

TABLE OF CONTENTS vii

LIST OF FIGURES x

LIST OF TABLES xv

1. INTRODUCTION: PARAMETRIC OPTIMIZATION FOR ENGINEERING DESIGN 1

1.1 Motivation for Parametric Optimization* 1
1.2 Terminology and Definitions 3
1.3 Contributions 5
1.3.1 Hypercone Heuristic for Parametric Pareto Dominance 5
1.3.2 Methods for Calculating the Parametric Hypervolume Indicator 6
1.3.3 An Improved Implementation of the Predictive Parameterized Pareto Genetic Algorithm (P3GA) 6
1.3.4 Multi-Objective Efficient Parametric Optimization (MO-EPO) Algorithm 7
1.3.5 Improved Scalable Test Problems for Performance Assessment of Parametric Optimization Algorithms 8

2. BACKGROUND* 9

2.1 Parametric Optimization 9
2.1.1 Single Objective Linear and Quadratic Methods 11
2.1.1.1 Numerical Continuation 13
2.1.1.2 Critical Region Method 15
2.1.2 Single Objective Nonlinear Methods 16
2.1.2.1 Numerical Continuation 16
2.1.2.2 Problem Approximation 18
2.1.3 Multi-Objective Nonlinear Methods 19
2.1.3.1 Predictive Parametric Pareto Genetic Algorithm (P3GA) 20

2.1.3.2 P3GA: Algorithmic Improvements 20
2.2 Efficient Optimization 23
2.2.1 Efficient Global Optimization (EGO) Algorithm 23
2.2.2 Multi-Objective Efficient Optimization 25

3. PARAMETRIC OPTIMIZATION FOR MORPHING STRUCTURES* 29

3.1 Motivation 29
3.2 Parametric Optimization 31
3.2.1 Theory 31
3.2.2 Methods 31
3.3 Camber Morphing Wing Description 32
3.4 Parametric Formulation of a Morphing Aerostructure Design Problem 33
3.5 Results 34
3.5.1 Two Parameter Problem 35
3.5.2 Three Parameter Problem 37
3.6 Benchmarking 39
3.7 Conclusion 42

4. EFFICIENT PARAMETRIC OPTIMIZATION* 45

4.1 Efficient Parametric Optimization (EPO) 45
4.2 Benchmarking 49
4.2.1 Parametric Hypervolume Indicator 49
4.2.2 Test Problem #1 51
4.2.2.1 One Parameter 51
4.2.2.2 Two Parameters 53
4.2.3 Test Problem #2 54
4.3 Finite Element Case Study 56
4.4 Discussion 59
4.5 Limitations and Conclusions 60

5. PARAMETRIC HYPERVOLUME INDICATOR 61

5.1 Multi-Objective Hypervolume Indicator 62
5.2 Parametric Incompatibility of the Multi-Objective Hypervolume Indicator 63
5.3 Parametric Hypervolume Indicator Methods 67
5.3.1 SVDD-Based Method 68
5.3.2 HCH-Based Method 69
5.3.3 Current Implementations 70
5.4 Results 71
5.4.1 Scalable Test Problems 71
5.4.2 Discussion of the Two pHVI Methods 76
5.4.3 Parametric Optimization with the HCH 77
5.5 Choosing the Reference Space and Hypercone Angle 80
5.6 Conclusions 84

6. MULTI-OBJECTIVE EFFICIENT PARAMETRIC OPTIMIZATION 86

6.1 Preliminaries 86
6.2 An Algorithm for Multi-Objective Efficient Parametric Optimization (MO-EPO) 88
6.2.1 Necessary Assumptions 89
6.2.1.1 Smooth and Continuous Functions 89
6.2.1.2 Predictive Parametric Pareto Dominance 89
6.2.2 Implementation-Specific Details 91
6.2.3 User Options 92
6.3 Benchmarking 94
6.3.1 Scalable Test Problem #1 95
6.3.2 Scalable Test Problem #2 101
6.3.3 Scalable Test Problem #3 104

7. PARAMETRIC OPTIMIZATION OF MORPHING AIRFOIL 107

7.1 Single Objective Formulation 108
7.2 Multi-Objective Formulation 112
7.3 Conclusions 114

8. PARAMETRIC OPTIMIZATION OF MORPHING SUPERSONIC CIVILIAN TRANSPORT* 116

8.1 Motivation 116
8.2 Equivalent Area Description 117
8.2.1 Equivalent Area Surrogate Model 119
8.2.2 Problem Formulation 122
8.3 Results and Discussion 124

9. CONCLUSIONS 129

9.1 Summary of Contributions 129
9.2 Limitations and Future Work 131

REFERENCES 133

APPENDIX A. TEST PROBLEM DEFINITIONS 147

A.1 Test Problem #1 147
A.2 Test Problem #2 148
A.3 Test Problem #3 149

APPENDIX B. CAMBER MORPHING WING DESCRIPTION* 150

LIST OF FIGURES

FIGURE Page

1.1 (a) Static robust design provides less sensitivity to changing conditions but accepts sub-optimality. (b) Morphing designs can adapt to changing conditions to maintain optimality. 2

2.1 Singularity types in parametric optimization: (a)-(c) folds, (d)-(f) branches, and (g)-(i) jumps. 14

2.2 Predictor-corrector via (a) orthogonal line search and (b) fixed-parameter, traditional optimization. Starting with point 1, calculate the linear approximation of $x^*(\theta)$ to predict point 2, then "correct" back onto the true path at point 3.......... 17

2.3 A list of optimization methods that rely on adaptive sampling of a response surface sorted by problem type.......... 24

3.1 Morphing airfoil example.......... 32

3.2 Results considering altitude held constant at 10,000 ft. (a) Population of structurally feasible morphed designs projected into this 3-D domain and lift-to-drag ratios of optimal performance across changing flight conditions. (b) Comparison between the performance of the morphing NACA 0012 configurations and that of a fixed NACA 0012, a fixed NACA 4415 and a fixed NACA 64(1)-212. 35

3.3 Four structurally consistent morphed configurations from the optimal solution map and their relation to the performance model are depicted along with airfoil shape coefficients, designed flight condition, and lift-to-drag ratio.......... 36

3.4 Approximate optimal solution map providing the value of each shape coefficient for the camber morphing NACA 0012 when altitude is fixed at 10,000 ft. Such a map could be used to schedule a controller such that morphing structures automatically adapt during operation.......... 38

3.5 Two-dimensional subsets of the four-dimensional performance model for the morphing airfoil: (a) sections of this model where altitude is fixed at 20,000 ft and airspeed is sampled at 20, 35, 50, and 65 m/s; (b) sections of the same model fixing airspeed at 35 m/s and aircraft altitude at 10, 20, 30, and 40 thousand feet. 39

3.6 Approximate optimal solution map providing the value of each shape coefficient for the camber morphing NACA 0012 when airspeed is fixed at 35 m/s. 40

3.7 Results of parametric optimization via P3GA contrasted with the results of the traditional genetic algorithm [1]: (a) the performance model (surface) as compared to the 8 x 8 grid of traditional optimization solutions (points), (b) percent error between the performance model and validation points as calculated via Eqn. (3.2)... 41

3.8 Difference between the results of the optimal solution map and the traditional genetic algorithm [1] from the four optimization variables (P_1^c, P_2^c, P_3^c, and P_4^c) using Eqn. (3.3) for the two-parameter subset where altitude is fixed at 10,000 ft........... 43

4.1 Parametrically nondominated solutions found from all 20 runs for each method against the true solution for test problem # 1 with one parameter...................... 52

4.2 Average pHVI versus the number of high-fidelity function evaluations for test problem #1 with (a) one parameter and (b) two parameters................................ 53

4.3 Average pHVI versus the number of high-fidelity function evaluations for test problem #2 with (a) one parameter and (b) two parameters................................ 55

4.4 The (a) full stiffened panel assembly and (b) a single section of the assembly for Abaqus analysis. 57

4.5 Performance of parametrically nondominated solutions for each method for the finite element case study. 59

4.6 pHVI versus the number of expensive design evaluations for the finite element analysis case study.. 59

5.1 Notional representation of dominated space in two objectives. Arrows represent objective preference direction. Green curve represents a notional Pareto frontier. 63

5.2 Parametric data in two dimensions (one objective and one parameter) demonstrating parametric Pareto dominance. Arrow represents objective preference direction. Green curve represents a notional parameterized Pareto frontier. 64

5.3 (a) True parametric solution in two dimensions (one objective and one parameter) for a notional parametric problem. (b) Application of parametric Pareto dominance to data in a grid. Arrow represents objective preference direction. Parameter has no preference direction.. 66

5.4 The application of Pareto dominance to parametric data is fundamentally incorrect as (a) the dominated space can be overestimated and (b) parametrically nondominated data are incorrectly classified as dominated. Black arrow denotes true objective preference direction while red arrow denotes the invalid preference given to the parameter.. 66

5.5 Dominated space of two-dimensional data (one objective and one parameter) illustrated by shaded regions for the (a) SVDD-based method, and (b) HCH-based method. Arrows represent objective preference direction. Green curves represent notional parameterized Pareto frontiers. 69

5.6 Test problem #1 (a) pHVI and (b) time results for one objective and one parameter. . 72

5.7 Test problem #1 (a) pHVI and (b) time results for one objective and two parameters. 72

5.8 Test problem #1 (a) pHVI and (b) time results for two objectives and one parameter. 73

5.9 Test problem #2 (a) pHVI and (b) time results for one objective and one parameter. . 73

5.10 Test problem #2 (a) pHVI and (b) time results for one objective and two parameters. 74

5.11 Test problem #2 (a) pHVI and (b) time results for two objectives and one parameter. 74

5.12 (a) Convergence of pHVI and (b) nondominated designs for three implementations of the hypercone heuristic as dominance criterion in modified P3GA applied to test problem #1 with one objective and one parameter. Note that the true solution is given by the green line in (b). 78

5.13 Nondominated set of solutions for (a) a single run of each hypercone angle and (b) a single run of $\phi_{hc} = 150^\circ$ only. Result using the 30° cone angle is farthest from the true solution, and the result using the 150° cone angle misses parts of the parameter space.. 79

5.14 HCH visualization for different combinations of hypercone angle and reference space description (cf. Table 5.2) on a sparse data set of 20 data points from test problem #1.. 82

5.15 HCH visualization for different combinations of hypercone angle and reference space description (cf. Table 5.2) on a more dense data set of 200 data points from test problem #1. 83

6.1 pHVI versus the number of design evaluations for test problem 1 (cf. Eqn (A.1)) considering all parameter and objective functions as expensive. 96

6.2 A close up view of the results presented in Fig. 6.1 to visualize the trends at low numbers of design evaluations.. 97

6.3 pHVI versus the wall-clock time assuming that each design evaluation requires 1 minute for test problem 1 (cf. Eqn (A.1)) considering all parameter and objective functions as expensive.. 98

6.4 pHVI versus the number of design evaluations for test problem 1 (cf. Eqn (A.1)) considering all objective functions as expensive and all parameters as cheap and independent.. 99

6.5 A close up view of the results presented in Fig. 6.4 to visualize the trends at low numbers of design evaluations. 100

6.6 pHVI versus the number of design evaluations for test problem 2 (cf. Eqn (A.2)) considering all parameter and objective functions as expensive. 102

6.7 pHVI versus the wall-clock time assuming that each design evaluation requires 1 minute for test problem 2 (cf. Eqn (A.2)) considering all parameter and objective functions as expensive. 103

6.8 pHVI versus the number of design evaluations for test problem 3 (cf. Eqn (A.3)) considering all parameter and objective functions as expensive. 105

6.9 pHVI versus the wall-clock time assuming that each design evaluation requires 5 minutes for test problem 3 (cf. Eqn (A.3)) considering all parameter and objective functions as expensive. 106

7.1 pHVI value as a function of time for P3GA and MO-EPO for the single objective problem (Eqn. (7.1)) on the range of (a) 0-10,000 seconds and (b) 0-500,000 seconds. 110

7.2 pHVI value as a function of the number of designs evaluated for P3GA and MO-EPO for the single objective problem (Eqn. (7.1)) on the range of (a) 0-750 design evaluations and (b) 0-50,000 design evaluations. 110

7.3 Parametric frontiers from P3GA requiring 50,000 design evaluations versus that of MO-EPO requiring 518 design evaluations for multiple 2D subspaces of the full 4D attribute space. 111

7.4 pHVI value as a function of (a) time and (b) the number of designs evaluated for P3GA and MO-EPO for the multi-objective problem (Eqn. (7.2)). 113

7.5 Parametrically nondominated designs from P3GA requiring 2,000 design evaluations versus all designs from MO-EPO requiring 1,000 design evaluations for multiple 2D subspaces of the full 5D attribute space. Lift-to-drag ratio should be maximized while moment coefficient should be minimized. 114

7.6 Optimal morphed airfoil shapes found from MO-EPO prioritizing (a) lift-to-drag ratio and (b) moment coefficient magnitude. 115

8.1 Baseline inviscid equivalent area distribution of the NASA Configuration 25D. 118

8.2 Spline-based equivalent area distribution modification schematic. 118

8.3 Variations of the equivalent area distribution for the NASA 25D aircraft for various local Mach number and angle of attack. 119

8.4 Mach and angle of attack combinations used for training and validation. 120

8.5 Equivalent area distribution predictions compared to the distribution calculated via CFD for (a) $\Delta M = -0.075$, $\Delta\alpha = 0.000$, (b) $\Delta M = 0.075$, $\Delta\alpha = -0.375$, and (c) $\Delta M = 0.000$, $\Delta\alpha = 0.375$. . . . 121

8.6 PLdB prediction error of validation data presented by (a) a heatmap and (b) histogram. . . . 122

8.7 pHVI value as a function of the number of designs evaluated for P3GA and MO-EPO for the problem defined in Eqn. (8.2) for (a) 0-6,000 design evaluations and (b) 0-120,000 design evaluations. . . . 125

8.8 MO-EPO results plotted in nine subspaces of the full, four-dimensional parameterized Pareto frontier. Dotted line illustrates the resulting PLdB of the undeformed aircraft at each variation of Mach and angle of attack. . . . 126

8.9 Deformations to the equivalent area distribution for PLdB reduction resulting in a perceived sonic boom of (a) 80.61 dB and (b) 80.52 dB. . . . 127

LIST OF TABLES

TABLE Page

3.1 Shape coefficients for parent (reference) NACA 0012 airfoil considered in this work from which all morphed configurations are generated. 33

4.1 Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #1 with one parameter (Eqn. (A.1)). Theoretically, negative δ values are not possible, but there is a small error inherent in the calculated pHVI. Each method evaluates the high-fidelity model only 150 times.. 51

4.2 Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #1 with two parameters (Eqn. (A.1)). Each method evaluates the high-fidelity model 600 times. 54

4.3 Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #2 with one parameter (Eqn. (A.2)). Each method evaluates the high-fidelity model only 150 times......................... 54

4.4 Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #2 with two parameters (Eqn. (A.2)). Each method evaluates the high-fidelity model only 600 times......................... 55

5.1 Example of Reference Space Description... 80

5.2 Reference Space Descriptions used in Figs. 5.14 and 5.15............................. 81

6.1 P3GA options for all three test problems.. 95

7.1 MO-EPO options to solve Eqn. (7.1). Refer to Algorithm 5. 109

7.2 Bounds of the reference space used to calculate the pHVI for solutions to Eqn. (7.1). Note that the reference space is comprised of all attributes (parameters and objectives). 109

7.3 Bounds of the reference space used to calculate the pHVI for solutions to Eqn. (7.2). Note that the reference space is comprised of all attributes (parameters and objectives). 113

8.1 MO-EPO options to solve Eqn. (8.2). Refer to Algorithm 5. 124

8.2 P3GA options to solve Eqn. (8.2)...124

8.3 Bounds of the reference space used to calculate the pHVI for solutions to Eqn. (8.2). Note that the reference space is comprised of all attributes (parameters and objectives). 125

1. INTRODUCTION: PARAMETRIC OPTIMIZATION FOR ENGINEERING DESIGN

1.1 Motivation for Parametric Optimization*

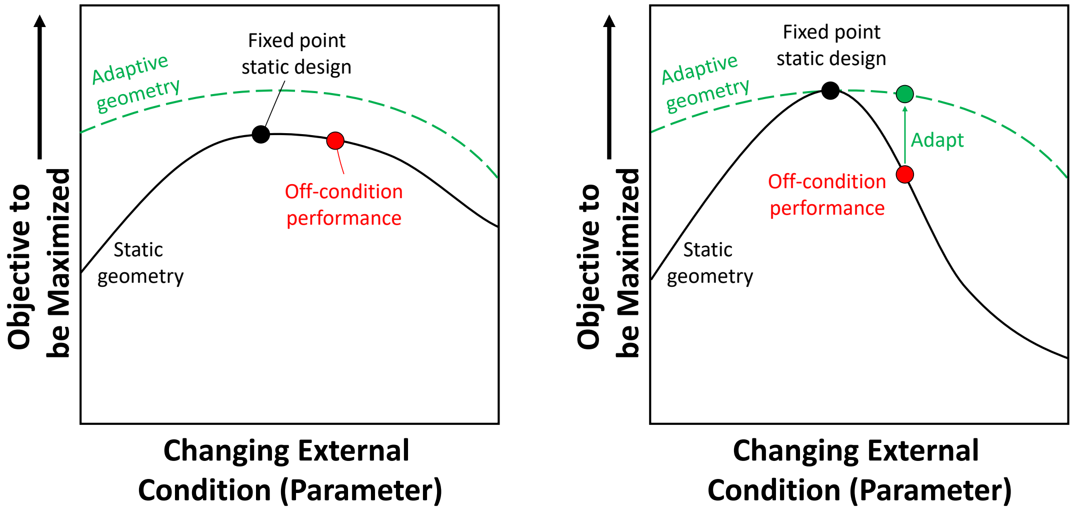
Some products and systems have long been designed to be reconfigured to meet changing needs such as flaps on an aircraft or variable suspension stiffness in an automobile. Under different conditions, one configuration may be desired over another configuration. As technology evolves, engineers are able to design systems to be increasingly reconfigurable. Ideally, systems should be adaptive: able to continuously reconfigure in situ as needs or conditions change. To some extent, such systems already exist; a pilot (or onboard computer) will configure an aircraft's flaps and slats differently if the aircraft is taking off, cruising, or landing. This "adaptation" is only a small subset of the potential configurations for an adaptive aircraft. Imagine aircraft wings and fuselage capable of continuous shape change during flight to optimize some performance or cost function. Such a design problem poses a difficult challenge: instead of designing a single aircraft capable of operating over many conditions, an engineer must find the optimal reconfigured aircraft for all possible conditions. It is no longer sufficient to optimize a system to be reliable or robust to changing conditions; adaptive systems should be ***optimal*** throughout changing conditions.

Leal et al. [3] demonstrate the optimization of a morphing airfoil for two specific conditions, cruise and landing. Gamboa et al. [4] demonstrate the optimization of a morphing airfoil for many discrete values of airspeed. Although engineers can repeat a traditional optimizer for a small set of conditions, these problems can be solved more quickly and with a more complete answer through parametric optimization [5]. Parametric optimization is an extension of traditional optimization in that it solves for the optimal design configuration as a function of parameters that are either currently unknown or changing. For adaptive systems design, changing external conditions can be represented by these parameter variables. Once the parametric optimization problem has been

*Part of this section is reprinted with permission from "Parametric optimization for morphing structures design: application to morphing wings adapting to changing flight conditions" by Weaver-Rosen, J.M., Leal, P.B.C., Hartl, D.J., and Malak, R.J., 2020. *Structural and Multidisciplinary Optimization*, 62, 2995-3007 Copyright 2020 by Springer Nature Switzerland AG [2].

solved, an adaptive system has a pre-solved optimal configuration for any parameter values in a given range. Therefore, one can almost instantly know the optimal configuration of an adaptive system given its current state. Parametric optimization of a morphing airfoil across all feasible values of airspeed, angle of attack, and altitude has been shown to be effective [2].

Although it is possible to optimize a non-morphing structure for average performance over a distribution of potential operating conditions (Fig. 1.1a), this typically results in a sub-optimal solution at any given operating condition. Traditional (non-morphing) structures may also be optimized to a specific operating condition, but changes in the external conditions will lead to sub-optimal off-condition performance. In contrast, a morphing structure can adapt its geometry to be optimal at any operating condition within some prescribed range (Fig. 1.1b).


Objective to be Maximized
Adaptive geometry
Fixed point static design
Off-condition performance
Static geometry
Changing External Condition (Parameter)
Objective to be Maximized
Adaptive geometry
Fixed point static design
Adapt
Off-condition performance
Static geometry
Changing External Condition (Parameter)

(a)

(b)

Figure 1.1: (a) Static robust design provides less sensitivity to changing conditions but accepts sub-optimality. (b) Morphing designs can adapt to changing conditions to maintain optimality.

Parametric optimization has been shown to be beneficial to engineering design [6, 7, 2, 8], but parametric optimization can also require extensive analysis to find a solution. Unfortunately, many optimization techniques (parametric optimization included) can become intractable when analysis becomes expensive. When the analysis of a single design takes 0.001 seconds, analyzing 100,000 design configurations for optimization purposes is easily done in a couple of minutes. When analysis of a single design begins to take minutes or hours (which is not uncommon for some engineering applications), analyzing even a few hundred or thousand designs becomes time consuming. The aim of this research is to make parametric optimization more amenable to complex engineering problems requiring expensive analyses by reducing the number of analyses necessary to optimize a design.

1.2 Terminology and Definitions

Analysis: Analysis is the process of calculating the performance metrics (i.e., figures of merit, attributes, etc.) of a given product or system. For example, analysis describes the calculation of maximum stress in a structural component or the weight of a product or system.

Design: Design is the process of prescribing the details for a product or system to meet a desired application. Design is often referred to as the inverse of analysis, or as an inverse problem. That is to say: analysis calculates system attributes given a system description whereas design is the process of finding an acceptable system description given desired system attributes.

Objectives: Objectives, or objective functions, are attributes of a system for which the design engineer has a preference. For example, minimize mass, minimize cost, and maximize performance all constitute objectives.

Parameters: Parameters are variables for which the design engineer has no preference. Parameters can be system attributes that do not qualify as objectives due to the lack of a preference. For example, an engineer might have no preference for the stiffness of a structure. Parameters can also be variables that impact a design problem, but are outside of the designer's control. Parameters may even be unknown to the designer or changing. For example, the planned flight conditions for an aircraft are critical to optimizing an aircraft, but such conditions are not controlled by the designer. Therefore, flight conditions are parameters.

Optimization Variables: Optimization variables are the variables that the designer has control over during an optimization procedure. Optimization variables include design variables (i.e., variables that define the design of a system) such as the sizes and dimensions of parts. Optimization variables also include control variables that define the state of a reconfigurable system such as the degree of actuation. The distinction between design variables and control variables is that design variables do not change once the product or system is built whereas control variables can change aspects of a design after it is built. Taking an aircraft as an example, the dimensions of the wing structure are design variables because they are fixed once the aircraft is built, but control variables control the deployment of slats and flaps even after the aircraft is built. Depending on the specific problem formulation, an engineer could be optimizing a system with respect to design variables, control variables, or both.

Constraints: Constraints are requirements or other conditions that must be met to solve a design problem. Constraints may apply to system attributes; for example, an engineer might be constrained to design a product that weighs less than some threshold value or satisfy a cost requirement. Constraints may also apply directly to optimization variables to preserve physical meanings of a problem and/or focus the design exploration. For example, quantities such as length or thickness cannot have negative values. Alternatively, a maximum length or thickness could be prescribed to ensure compatibility of a design with its intended application. In some cases, physical laws such as conservation of momentum are treated as constraints that must be met, but these are more often encoded into the analysis.

Traditional Optimization: Traditional optimization is the process by which an engineer seeks to find the design(s) with the most preferred objective values subject to constraints by changing the values of the optimization variables. In general, a traditional optimization problem can have any number of objectives, constraints, and optimization variables. Traditional optimization requires the assumption of parameter values to solve the problem.

Parametric Optimization: Parametric optimization is the process by which an engineer seeks to find the design(s) that solve a traditional optimization problem as a function of parameters. Instead of assuming parameter values as in traditional optimization, parametric optimization solves for the solution to the problem for all parameter values in a user-defined or possible range.

Efficient Optimization: Efficient optimization refers to the type of optimization algorithms that seek to find a solution with a minimal number of design evaluations often assisted by metamodels or surrogate models. Such techniques are usually employed when analyzing a product/system is computationally or monetarily costly. By reducing the number of design evaluations needed to solve a design problem, an engineer can reduce the expense of optimization.

Sensitivity Analysis: Sensitivity analysis is the process by which an engineer quantifies how small changes in the parameter values affect the system attributes (objectives and/or constraints). Sensitivity analysis is the process of answering a related but different question compared to parametric optimization. For sensitivity analysis, the question being asked is "How do changes in parameter values affect the performance of a given system?" whereas the question for parametric optimization is "How do changes in parameter values affect the description of the optimal system?".

1.3 Contributions

The main contribution of this work is the Multi-Objective Efficient Parametric Optimization algorithm (MO-EPO). MO-EPO is a method that brings together efficient optimization and parametric optimization to efficiently solve multi-objective, parametric optimization problems with expensive objective and/or parameter functions. Such an algorithm allows for complex optimization problems (e.g., optimal morphing, product family design, controller design, and preliminary design under uncertainty) to be solved in a more reasonable amount of time. All contributions of this work are introduced below:

1.3.1 Hypercone Heuristic for Parametric Pareto Dominance

Parametric Pareto dominance (PPD) has a very specific mathematic definition that makes it difficult to be used in optimization [9]. In prior works, researchers have used a support vector domain description (SVDD) as a predictor of PPD [9, 10, 11]. Much of this prior work has shown the effectiveness of the SVDD for predicting PPD. However, its computational expense, undesirable properties, and occasional numerical instability motivated the exploration of a new technique.

Therefore, the hypercone heuristic (HCH) was defined and implemented in Matlab code as another option for predicting PPD. Results show that this technique is faster and well-behaved compared to using an SVDD. The main limitations of this technique are the need to choose a somewhat arbitrary reference space and hypercone angle. See Chapter 5 for more.

1.3.2 Methods for Calculating the Parametric Hypervolume Indicator

Using both the SVDD and HCH for parametric Pareto dominance (PPD), two methods for calculating the parametric hypervolume indicator (pHVI) are created. These pHVI techniques provide the first convergence criteria for solving general nonlinear black-box parametric optimization problems. In prior implementations of nonlinear black-box parametric optimization, the only stopping criteria was reaching a maximum number of iterations or exhausting a maximum number of design evaluations. The pHVI measure allows for the evaluation and comparison of parametric optimization methods, for the detection of convergence of a given algorithm, and for added information that can be used by an optimization algorithm.

The SVDD-based method utilizes the SVDD prediction of the feasible attribute space (objectives and parameters). This SVDD can be used to determine the fraction of the user-defined reference space that is dominated by a set of data by determining how much of the space is enclosed by or dominated by the SVDD. The HCH-based method utilizes the newly introduced HCH for the prediction of PPD. In the same way that hypercones are used to determine whether a design is dominated or not, the volume of these hypercones can be used to approximate the fraction of the user-defined reference space dominated by a set of data. See Chapter 5 for more.

1.3.3 An Improved Implementation of the Predictive Parameterized Pareto Genetic Algorithm (P3GA)

Building upon the previous two contributions, the existing P3GA algorithm [10] for parametric optimization is improved. The previous implementation allowed for only SVDD to be used for parametric Pareto dominance (PPD) and the only stopping criteria is a maximum number of generations. The improved implementation of P3GA allows a user to select either the SVDD or

HCH methods for predicting PPD. Additionally, the improved P3GA allows for the calculation of the pHVI during optimization at each generation to: detect solution convergence and to allow a user to visualize the solution progress even when the number of objectives and parameters cannot be visualized. Due to the expense of calculating the pHVI through an SVDD (see Chapter 5), only the HCH-based pHVI calculation method is implemented (though a user could still choose to approximate PPD via the SVDD). See Section 2.1.3.2 for more details.

1.3.4 Multi-Objective Efficient Parametric Optimization (MO-EPO) Algorithm

The main contribution of this work is a new, parametric optimization algorithm specifically for solving problems with expensive functions. In engineering design, it is very common for engineers to rely on expensive procedures (either computationally or monetarily) to gather data. Some well-known examples of these expensive procedures include computational fluid dynamics (CFD), finite element analysis (FEA), and physical experiments. Although some implementations of these expensive procedures may seem relatively cheap, design optimization can require many iterations of these procedures. So an analysis that takes 10 minutes to compute would take nearly 70 days to solve an optimization procedure assuming that 10,000 designs evaluations are needed. The goal of efficient optimization is to reduce the total number of design evaluations needed to find the solution to these expensive problems.

The MO-EPO algorithm utilizes the HCH-based pHVI technique and surrogate models of expensive functions to more quickly perform parametric optimization. Using techniques similar to Ref. [12], the MO-EPO algorithm can quickly search cheap surrogate models to predict the expected improvement of taking new samples of the expensive model. This adaptive sampling technique guided by evolving surrogate models allows an engineer to solve parametric design problems faster and more efficiently. Such an optimization technique is needed to solve increasingly complex engineering design problems such as a shape morphing aircraft. See Chapter 6 for more.

1.3.5 Improved Scalable Test Problems for Performance Assessment of Parametric Optimization Algorithms

Furthermore, an existing suite of scalable parametric test problems is supplemented with variations of the same problems that allow for independent parameters in addition to the existing dependent parameter problems. The existing suite of problems allows for any number of objectives, dependent parameters, and optimization variables. However, not all parameters are dependent on optimization variables. The improved test suite allows for the same problems to be solved with any combination of dependent and independent parameters in addition to any number of objectives and optimization variables.

The example of designing a liquid metal active cooling concept is an illustration of a problem with both an independent and a dependent parameter [5]. In this example, the engineer seeks to maximize the thermal conductivity and minimize the mass of the liquid metal channel. With the ability to design the fluid channel depth, fluid channel aspect ratio, and applied voltage, the parameters outside of the designer's control are the power output of the heat source and the temperature of the hot reservoir. As a subsystem of a larger system, both the power output of the heat source and the allowable temperature of this heat source would be determined by the designers of that heat source. The power output of the heat source is considered as an independent parameter in the design of the cooling concept because no change in optimization variables will change the power output from the heat source. On the other hand, the hot reservoir temperature *is* dependent on the design of the cooling system, but the allowable value is not known yet. In this case, the engineer seeks to optimize this system before knowing these parameter values via parametric optimization.

Despite the existence of engineering problems with both independent and dependent parameters, the parametric optimization suite of test problems presented by Galvan et al. [5] does not explicitly allow for parameters to be independent of the optimization variables. Therefore, these problems have been modified to allow for the explicit inclusion of independent parameters in a manner that retains the desired properties of each problem.

2. BACKGROUND*

2.1 Parametric Optimization

Parametric optimization, also known as multi-parametric programming, is the process of solving an optimization problem as a function of one or more variables, termed parameters, over a given range [13, 14, 15, 16, 6, 7, 2]. In this context, parameters are interpreted as variables on which the optimal solution depends but that are not controlled by the optimization process. For example, an engineer may wish to perform an optimization study but believes several critical assumptions about the problem (e.g., those affecting boundary conditions, etc.) are likely to change during the course of the project. The engineer can proceed in several ways including: optimizing to current values, optimizing to perceived worst-case values, and performing optimization under uncertainty (e.g., robust design optimization [17, 18] or reliability-based design optimization [19]) based on beliefs about the likely values of the parameters. In contrast to these options, parametric optimization enables the engineer to find the optimal solution as a function of parameter values and later, once true parameter values are determined, select the relevant solution from the parametric solution set.

Many engineering and systems design scenarios can benefit from parametric optimization. For example, an engineer may wish to perform an optimization study but believes that critical design requirements are likely to change. Such requirements could be represented using parameters in a parametric optimization framework [7]. A related example is when the engineer performing the optimization lacks security clearance to know the precise values of particular variables. In this case, parametric optimization can provide the optimal solution as a function of those variables and this solution can be used by individuals with appropriate security clearance [7]. Another use of

*Part of this chapter is reprinted with permission from "Parametric optimization for morphing structures design: application to morphing wings adapting to changing flight conditions" by Weaver-Rosen, J.M., Leal, P.B.C., Hartl, D.J., and Malak, R.J., 2020. *Structural and Multidisciplinary Optimization*, 62, 2995-3007 Copyright 2020 by Springer Nature Switzerland AG [2] and from "Efficient Parametric Optimization for Expensive Single Objective Problems" by Weaver-Rosen, J.M. and Malak, R.J., 2021. *Journal of Mechanical Design*, 143(3), 031711 (9 pages), Copyright 2021 by ASME [11].

parametric optimization is to find optimal solutions for an adaptive system as a function of environmental or system conditions that change during system operation. Examples of this are flight conditions for a morphing wing [2, 4] ambient conditions for chemical process control [6, 20], and time/state for the design of a general controller [21, 22, 8]. Parametric optimization has also been used as a method for solving bi-level optimization problems by reformulating the problem as a parametric one [23]. Parametric optimization even has applications outside of engineering including to determine market equilibria as a function of exogenous parameters such as the cost of goods or labor [24].

Parametric optimization began as a form of sensitivity analysis of an optimization problem solution with respect to variable parameters [13]. Whereas typical sensitivity analysis calculates the performance variation of a fixed design with respect to changing parameters, parametric optimization solves for how the ***optimal*** solution changes with respect to changing parameters. The most general parametric optimization problem can be formulated as [11]:

$$
\begin{aligned}
\mathbf{J}^*(\boldsymbol{\theta}) = \min_{\boldsymbol{x}} \; &\mathbf{J}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) \;\; \forall \;\; \boldsymbol{\theta} \in [\boldsymbol{\theta}_{lb}, \boldsymbol{\theta}_{ub}]\,, \\
\text{subject to:}& \\
&\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) \leq \mathbf{0}, \\
&\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = \mathbf{0}, \\
&\boldsymbol{x}_{lb} \leq \boldsymbol{x} \leq \boldsymbol{x}_{ub}\,, \\
&\boldsymbol{x} \in \mathrm{I\!R}^k\,, \\
&\boldsymbol{\theta} \in \mathrm{I\!R}^m\,.
\end{aligned}
\tag{2.1}
$$

where $\mathbf{J}$ is the set of objective functions to be minimized, $\boldsymbol{x}$ is the vector of optimization variables bounded by $\boldsymbol{x}_{lb}$ and $\boldsymbol{x}_{ub}$, $\boldsymbol{\theta}$ is the vector of parameters bounded by $\boldsymbol{\theta}_{lb}$ and $\boldsymbol{\theta}_{ub}$, and $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\theta})$ represent inequality and equality constraints, respectively, as functions of both optimization variables and parameters. Although the parameters $\boldsymbol{\theta}$ are often written as independent variables, parameters can be dependent or independent variables. Independent parameters are those that do *not* depend on the optimization variables. Examples include: operational conditions or currently unknown variables outside of the system that will be sensed during optimization. In contrast, dependent parameters are system attributes dependent on the optimization variables for which a designer has no preference. In many cases, requirements on system attributes can be formulated as dependent parameters. For example, when optimizing a structure to be as light as possible while withstanding an unknown external load, the problem can be formulated such that the allowable load for the structure is the parameter [11]. This allowable load is calculated based on the geometry of the structure, and is therefore dependent on the design variables.

The solution to a parametric optimization problem is the parameterized Pareto frontier $\mathbf{J}^*(\boldsymbol{\theta})$ [9]. This parameterized Pareto frontier can inform a designer about how the traditional Pareto frontier changes according to the specific parameter values. In cases with only a single objective, an *optimal solution map* $\boldsymbol{x}^*(\boldsymbol{\theta})$ can also be found. This optimal solution map represents the optimal values for the optimization variables given any variation of the parameters. Mathematically, $\boldsymbol{x}^*(\boldsymbol{\theta}) : \mathrm{I\!R}^m \rightarrow \mathrm{I\!R}^k$ such that: $J^*(\boldsymbol{\theta}) = J(\boldsymbol{x}^*(\boldsymbol{\theta}), \boldsymbol{\theta})$. Even with only a single objective, the optimal solution map has its limitations as it may not have a unique formulation if it exists at all. This will be discussed in more detail the following subsections.

2.1.1 Single Objective Linear and Quadratic Methods

As with many optimization techniques, there are specialized solution methods for the optimization of linear or quadratic problems. Under the assumption that a designer can find a locally unique solution (usually through traditional optimization) satisfying the second order sufficient conditions for any one set of parameter values, Eqn. (2.2) can be used to estimate how the optimal value of a single objective changes with respect to changing parameter values [13]:

$$\frac{\partial J^*}{\partial \boldsymbol{\theta}} = (M(\boldsymbol{\theta}_0))^{-1} N(\boldsymbol{\theta}_0)\,, \tag{2.2}$$

where $\boldsymbol{\theta}_0$ denotes the vector of parameter values for which an optimal solution is known and:

$$M = \begin{bmatrix} \nabla^2_{\mathbf{x}} L & -\nabla_{\mathbf{x}} g_1 & \dots & -\nabla_{\mathbf{x}} g_q & \nabla_{\mathbf{x}} h_1 & \dots & \nabla_{\mathbf{x}} h_r \\ \lambda_1 \nabla^T_{\mathbf{x}} g_1 & g_1 & & 0 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots & \ddots & \vdots \\ \lambda_q \nabla^T_{\mathbf{x}} g_q & 0 & & g_q & 0 & \dots & 0 \\ \nabla_{\mathbf{x}} h_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \nabla_{\mathbf{x}} h_r & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix}, \tag{2.3}$$

$$N = \begin{bmatrix} -\nabla_{\boldsymbol{\theta}} \nabla_{\mathbf{x}} L \\ -\lambda_1 \nabla_{\boldsymbol{\theta}} g_1 \\ \vdots \\ -\lambda_q \nabla_{\boldsymbol{\theta}} g_q \\ -\nabla_{\boldsymbol{\theta}} h_1 \\ \vdots \\ -\nabla_{\boldsymbol{\theta}} h_r \end{bmatrix}. \tag{2.4}$$

In Eqns. (2.3) and (2.4), L represents the Lagrangian of the optimization problem, g_i and h_i represent the q inequality and r equality constraints of the optimization problem, respectively, λ_i represents the Lagrange multiplier associated with the i-th inequality constraint, and ∇ represents the Jacobian operator with respect to the variables in its subscript. Furthermore, one can use the same information to estimate the local change in the optimal optimization variables and Lagrange multipliers via [13]:

$$\begin{bmatrix} \mathbf{x}^*(\boldsymbol{\theta}) \\ \boldsymbol{\lambda}^*(\boldsymbol{\theta}) \\ \boldsymbol{\mu}^*(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ \boldsymbol{\lambda}_0 \\ \boldsymbol{\mu}_0 \end{bmatrix} + (M(\boldsymbol{\theta}_0))^{-1} N(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\,, \tag{2.5}$$

where $\mathbf{x}^*(\boldsymbol{\theta})$, $\boldsymbol{\lambda}^*(\boldsymbol{\theta})$, and $\boldsymbol{\mu}^*(\boldsymbol{\theta})$ represent the optimal optimization variables, inequality Lagrange multipliers, and equality Lagrange multipliers, respectively. Additionally $\mathbf{x}_0$, $\boldsymbol{\lambda}_0$, and $\boldsymbol{\mu}_0$ represent the optimal solution for the parameter values where $\boldsymbol{\theta} = \boldsymbol{\theta}_0$.

For problems that are linear or quadratic in nature, Eqns. (2.2) and (2.5) are perfect solutions so long as the set of active constraints remains constant. There are two related categories of techniques to solve linear or quadratic problems using Eqn. (2.5): those based in numerical continuation and those that utilize critical regions.

2.1.1.1 Numerical Continuation

From the early body of literature, researchers built upon the theoretical relationships from Eqns. (2.2) and (2.5) using continuation techniques also referred to as pathfollowing or homotopy methods [25]. These methods solve parametric problems by finding "paths" or "loops" of stationary points through the space of parameters. Recall from traditional optimization theory that all optima are stationary points, but not all stationary points are optima. Numerical continuation methods would, given any stationary point, use the relationship in Eqn. (2.5) to build a "path" of stationary points in a neighborhood around the initial stationary point. Note that a "loop" is simply a "path" that is continuous. Through the detection of *singularities*, one can detect certain features of a path and update the relationship defining the path as needed. Singularities include [26]:

- Folds: When a path "folds" back on itself at a point where the line tangent to the path is vertical with respect to the parameter (i.e., infinite gradient)
  - A fold results in a non-unique path where there is no longer a one-to-one mapping between the parameter and points on the path
  - In some cases, the path on one side (or even both sides) of the fold can violate one or more constraints
- Branches or Bifurcations: When a single path splits into more than one path at a point
  - In some cases, a branch may violate one or more constraints making an "infeasible branch"

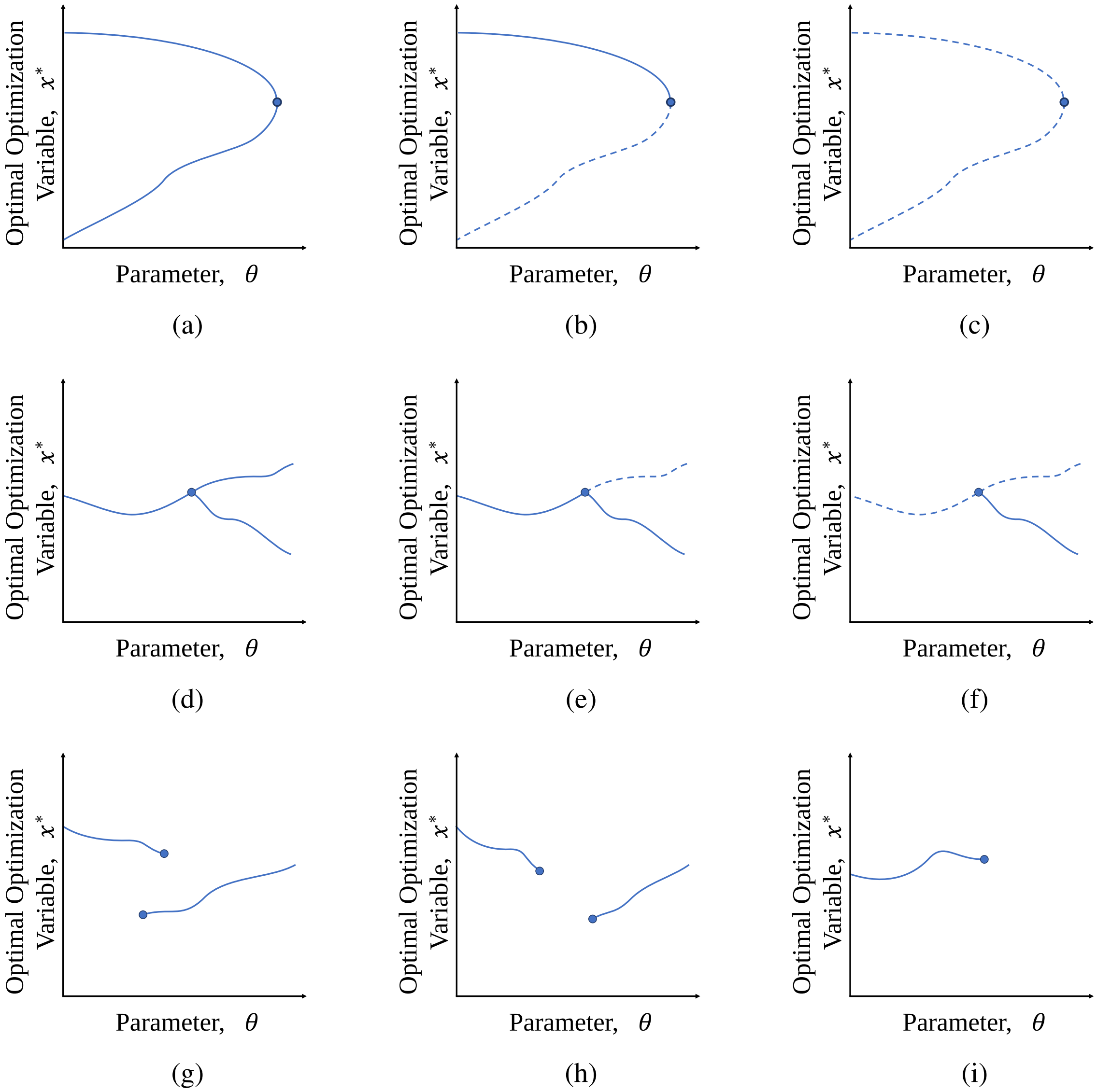


Figure 2.1: Singularity types in parametric optimization: (a)-(c) folds, (d)-(f) branches, and (g)-(i) jumps.

- Jumps: When a path (feasible or infeasible) appears to start/end in the middle of the parameter space (i.e., C^0 discontinuity)
  - There may or may not be another path to "jump" to and detecting this is not trivial

Figure 2.1 provides a visualization of these singularities. By understanding these singularities, researchers propose finding all paths or loops of stationary points in a given parameter space (feasible and infeasible). Then, the feasible path of minima solving the parametric problem can be extracted. A two-bar truss problem from Ref. [27] represents a complex example of a full path of stationary points (maxima, minima, and saddle points) for a problem with only one parameter and two optimization variables projected into two dimensions. This is a prime example of a physical engineering problem with a discontinuous, globally optimal solution for $\boldsymbol{x}^*$.

Much of the work in numerical continuation considers only a single parameter, but extensions to multi-parametric problems are possible. This is not a simple task, however, as a line path in one parameter becomes a plane or hyperplane when additional parameters are considered.

2.1.1.2 Critical Region Method

Another methodology for solving linear and quadratic parametric problems is the critical region method. This method remains somewhat agnostic to singularities presented in the continuation literature and focuses on finding a practical solution. Although there are some mathematical problems for which singularities exist, many problems involving physical systems can be solved without direct consideration of any singularities. For example, branches often involve one feasible and one infeasible path. In a practical application, following an infeasible path may not yield beneficial information.

The critical region method can be expressed by the following procedure [14]:

1. Solve the optimization problem for any value of $\boldsymbol{\theta}_0$ for which a solution does not already exist

2. Calculate the first-order approximation of the optimization variables in Eqn. (2.5) using the solution at $\boldsymbol{\theta}_0$

3. Define the critical region of $\boldsymbol{\theta}$ around $\boldsymbol{\theta}_0$ for which the active set of constraints does not change and the approximation from Step 2 is valid

4. Repeat Steps 1-3 until there is a solution for the entire feasible parameter space

5. Compile all solutions from Steps 2-3 to generate a piecewise solution for the full parameter space

Through this procedure, the exact solution to any linear or quadratic parametric problem can be found. Step 1 is simply a traditional optimization problem, and Step 2 is a calculation of the M and N matrices (Eqns. (2.3) and (2.4), respectively). Step 3 can be completed by using the formulation for optimal Lagrange multipliers (Eqn. (2.5)) to detect when the active set of constraints change, and Steps 4 and 5 are trivial. The final solution is given by a set of equations in the form of Eqn. (2.5) corresponding to specific regions of the parameter space. Although it is not stated or evaluated explicitly, the boundary of each critical region is, by definition, a singularity as defined in the numerical continuation literature. When the set of active constraints change (or when no feasible solution can be found beyond some point) a branch or jump in the path has occurred. In the critical region methodology, detecting this singularity is done implicitly by evaluating the problem constraints. It should be highlighted that a fundamental assumption of this method to ensure a globally optimal solution is that the global solution for each feasible critical region both exists and is unique.

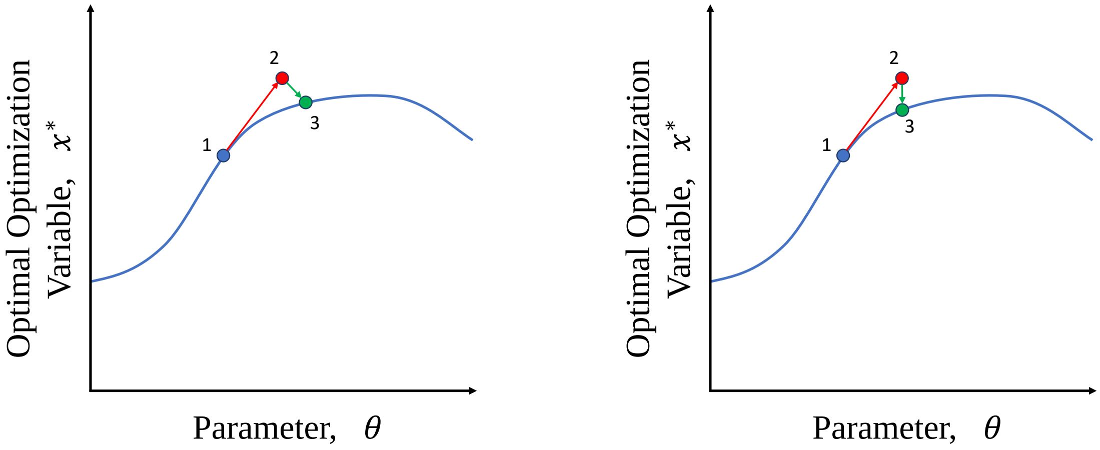
The POP (**P**arametric **OP**timization) toolbox [28] is an implementation of this procedure in the Matlab programming language. This toolbox has even been extended in recent years to solve bi-level problems through the B-POP (**B**i-level **P**arametric **OP**timization) toolbox by reformulating bi-level optimization problems in the context of parametric optimization [23].

2.1.2 Single Objective Nonlinear Methods

Although linear/quadratic theory is useful, an extension to solving nonlinear parametric problems is necessary for many engineering applications.

2.1.2.1 Numerical Continuation

Even some of the early work in numerical continuation sought to solve nonlinear problems (though typically only considering one parameter). There are two main approaches for numerical continuation of nonlinear problems: a simplex method and a predictor-corrector method. As illustrated in Fig. 2.2, the predictor-corrector method can use the formulation in Eqn. (2.5) to linearly predict the next point on the path. Then a line search or gradient optimization can be performed using the prediction as a starting point. In this way the prediction is "corrected" to find the next point on the solution path [26, 29, 27]. When using these methods, care must be taken to choose a reasonable predictor step size.


Optimal Optimization Variable, x^*
1
2
3
Parameter, θ
Optimal Optimization Variable, x^*
1
2
3
Parameter, θ

(a)

(b)

Figure 2.2: Predictor-corrector via (a) orthogonal line search and (b) fixed-parameter, traditional optimization. Starting with point 1, calculate the linear approximation of $x^*(\theta)$ to predict point 2, then "correct" back onto the true path at point 3.

Alternatively, a pathfollowing technique might rely on a simplex methodology. The POPAK (**P**arametric **O**ptimization **PA**c**K**age) software written in Fortran 90 [30] utilizes this simplicial pathfollowing to solve multi-parametric optimization problems via Brodzik's algorithm [31]. POPAK appears to work well for a small number of parameters, but does not perform well as the number of parameters grows beyond 4.

Hale [30] recommends the future work of utilizing predictor-corrector methods as described by Henderson [32, 33] to allow for expansion to more parameters. From my preliminary work in this area, Henderson's predictor-corrector algorithm appears to work well when solving parametric optimization problems, but even a small number of parameters requires many function evaluations to find the full path.

Since the work in this dissertation is focused on efficient optimization when functions are expensive, this methodology is not feasible. However, I do believe that numerical continuation via Henderson's algorithms could be useful in parametric optimization when the function evaluations are very cheap. The main advantage of an algorithm based on numerical continuation is the near-guarantee of finding a set of solutions on a single path. Since the work in this dissertation assumes that there is only one such path, I recommend that another algorithm (perhaps based in continuation) be used for problems where this assumption is not valid.

2.1.2.2 Problem Approximation

Another subset of nonlinear parametric optimization literature is based on solving an approximate problem instead of the exact problem. Depending on how nonlinear the problem is, the objective and constraint functions can be adequately approximated as linear and/or quadratic functions. As discussed by Johansen [16, 34], the exact nonlinear problem must first be partitioned such that a locally accurate quadratic problem can be defined. Once this is done, each approximate multi-parametric quadratic problem can be solved exactly using a critical region method as explained in Section 2.1.1.2.

As discussed in Refs. [16, 34, 14], the exact nonlinear problem can be partitioned adaptively through real-time error estimation. This procedure assumes that the objective and constraint functions are at least locally convex over the optimization variables and parameters. Under this assumption, one can state that the parametric solution is both continuous and convex [34]. Knowing that the true parametric solution is convex, the error of the solution to the approximate quadratic problem must be greatest at one of the corners of the current critical region. By systematically checking the error of each critical region corner (e.g., by solving the true nonlinear problem with the parameter values specified), an upper bound on the approximation error is found [14]. If any of these approximation errors are outside of a prescribed tolerance, the corners with unacceptable error can be used to create a subpartition of the space using that corner to define a new critical region. Through iterated use of this procedure, a piecewise approximate solution to the exact nonlinear problem can be generated that includes an upper bound of the error present in each critical region. Recall that these error bounds are only valid when the problem is locally convex in each critical region.

For problems that are not convex, the procedure mentioned previously would produce erroneous and misleading error estimates. Instead, another method can be used to solve the full nonlinear parametric problem [35, 14]. In this procedure an easy-to-evaluate overestimation and underestimation of the true problem are constructed and solved using one of the previous methods (i.e., quadratic or convex nonlinear). By evaluating both the over and under estimations of the nonlinear problem, the solution to the true problem is known to be within the solutions of the over/under estimators. With this knowledge, a designer can continuously partition the space, define new over/under estimators for each partition, and repeat until the solution to the over and under estimators are within a given tolerance. In this way, a different approximation of the true nonlinear problem is exactly solved for each partition of the parameter space including an estimate of the error. This method no longer relies on any assumptions of the nonlinear problem, but choosing the over and under estimators can be tricky. See Ref. [14] for a more thorough discussion.

2.1.3 Multi-Objective Nonlinear Methods

The previous subsections outline methods for single objective parametric optimization. Although any multi-objective problem can be reformulated as a single objective problem (e.g., via weighted sum, utility function, etc.), none of the previously described methods are capable of solving a multi-objective parametric optimization problem. In cases where an engineer is unsure of their tradeoffs between objectives and/or unsure of the feasible performance ranges of their objectives, a multi-objective optimization procedure can aid in design exploration to better inform a designer prior to choosing a final design.

2.1.3.1 Predictive Parametric Pareto Genetic Algorithm (P3GA)

To my knowledge, the only existing multi-objective nonlinear parametric optimization algorithms are P3GA [10, 7] and the novel MO-EPO (presented and discussed in detail in Chapter 6). P3GA, the predictive parameterized Pareto genetic algorithm, is an extension of NSGA-II [36] that uses predictive parametric Pareto dominance (PPD) via a support vector domain description (SVDD) [9] instead of traditional Pareto dominance.

The introduction of a *predictive* parametric Pareto dominance was instrumental to the introduction of evolutionary parametric algorithms. Strict parametric Pareto dominance requires that two different design alternatives have the *exact* same parameter values to determine if one dominates the other. If the parameter values are even marginally different, they are considered mutually nondominated or incomparable. When implemented in a stochastic evolutionary algorithm such as a genetic algorithm, it becomes very unlikely (impossible really) to generate two designs at the exact same parameter values without artificially forcing the parameter values to be equal. Therefore, P3GA predicts parametric Pareto dominance by generalizing the data in the attribute space (objectives and parameters) with an SVDD. The boundary of the SVDD is treated as the boundary of the feasible region in the attribute space. In this way, the nondominated boundary of the SVDD is the parameterized Pareto frontier.

Although the use of this predictive parametric Pareto dominance is not based on the mathematical theory as in Eqn. (2.2), it has been shown to be successful on a number of test problems and engineering case studies involving multiple objectives and/or multiple parameters [7, 5, 2, 8]. The P3GA methodology is used as the competition to the new algorithm presented in Chapter 6 since it is the only other multi-objective, multi-parametric optimization algorithm for nonlinear black-box functions.

2.1.3.2 P3GA: Algorithmic Improvements

Since its first publication in 2015 [10], the P3GA implementation has received multiple updates that have not yet been fully documented. These improvements include:

1. the ability to consider general nonlinear constraints,
2. the ability to evaluate members of a single generation in parallel,
3. the ability to initialize P3GA with information from an existing solution (as an alternative to specifying an initial population from a previous solution),
4. the ability to use a parametric hypervolume indicator (pHVI) as a visual progress indicator and/or convergence criteria, and
5. the ability to use the hypercone heuristic to determine dominance.

Most improvements enhance the practicality of the algorithm without changing any of its underlying methods. The only improvement that can affect the underlying algorithm is the fifth item above: the ability to use the hypercone heuristic as dominance criteria.

The first improvement is straightforward. The main P3GA function now accepts a nonlinear constraint function handle (in the same manner as “fmincon” and “ga” functions in Matlab [37]). With this function handle, all existing instances of linear constraint evaluation will now also consider the nonlinear constraints. The previous version of P3GA would attempt to “shift” an infeasible design to be feasible by solving the linear system of equations given by the linear constraints. With the consideration of nonlinear constraints, P3GA will still attempt to find a feasible point based on the linear constraints, but will not attempt to satisfy the nonlinear constraint. Instead, the design is marked as infeasible and will be unlikely to “reproduce” in a future generation via the tournament ranking of Ref. [36]. This improvement is necessary as many engineering case studies will require the satisfaction of nonlinear constraints as in Chapter 3 [2].

The second improvement is equally straightforward since genetic algorithms are regarded as “embarrassingly parallelizable”. The option structure that has long been an optional input to P3GA now contains an additional field called “Vectorized” giving the user the option to allow computations within each generation of the algorithm to be performed in parallel of one another utilizing the computer default settings.

The third improvement is practically important. The existing implementation of P3GA does allow for an initial population to be specified therefore allowing a user to initialize a new run of P3GA given the population members from an existing solution. The improvement discussed here allows all data from a previous run (not just population members) to initialize a new run of P3GA. For various reasons, an engineer might want to optimize an existing result further. For example, an engineer could have already run 50 generations of the algorithm. If the engineer has doubts about the solution quality, they may want to run an additional 50 generations to ensure that the solution is converged. With the new implementation of P3GA, an engineer can do just that. The algorithm only requires two inputs, the solution structure of the previous run (containing information such as: objective functions, previous populations and their corresponding performance, constraints, etc.) and a new option structure. The new option structure must contain a greater number of generations than the previous run (e.g., in the previous example where an engineer wants to run an additional 50 generations on top of the existing 50 generations, the option structure would specify a maximum of 100 generations), but any other option may be changed at will. The benefit of this improvement is that the information from the previous run is perfectly retained and included in the new solution rather than encoding all information from a previous run into a single initial population.

The fourth improvement is based on a new quality indicator for parametric optimization introduced in Ref. [11] and Chapter 4 and discussed in more detail in Chapter 5. The parametric hypervolume indicator (pHVI) can be used to monitor the quality of the problem solution at each generation to detect convergence. If the quality indicator remains relatively flat for many generations, one would say that the algorithm has converged because the solution is not improving with additional computation. Alternatively, the indicator can be used to tell an engineer that further generations are needed if the algorithm halts while the pHVI is continuing to increase (denoting that the solution is still improving with additional computational effort). This is especially useful for optimization problems that cannot be easily visualized.

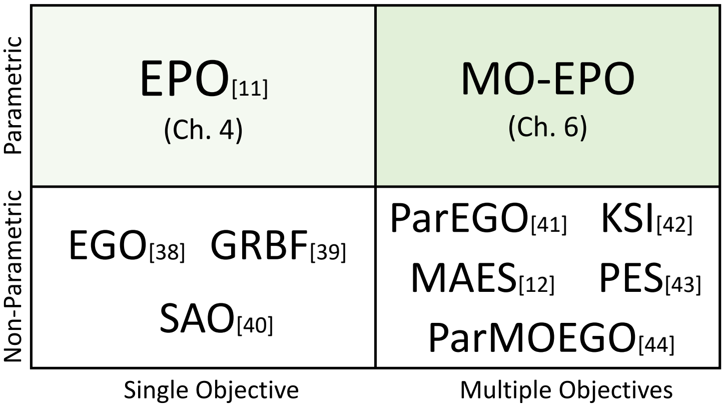
The fifth and final improvement introduces a second method to predict parametric Pareto dominance within P3GA. Introduced in Chapter 5, the hypercone heuristic (HCH) used to calculate the pHVI quality measure can also be used to detect dominance. The implementation of P3GA in Matlab now allows a user to decide between the new HCH method and the existing support vector domain description (SVDD) method for predicting parametric Pareto dominance.

2.2 Efficient Optimization

A challenge for any optimization technique is the consideration of computationally intensive analyses (e.g., finite element analyses (FEA), computational fluid dynamics (CFD), or physical experiments). For traditional optimization (i.e., non-parametric optimization), a common strategy is to combine response surface models with an adaptive sampling technique. These approaches operate with the response surface and execute expensive analyses only when the result is likely to be informative. The efficient global optimization (EGO) algorithm [38], Gutmann's radial basis function (GRBF) method [39], and sequential approximation optimizations (SAO) [40] are examples of such techniques for single-objective optimization. The Pareto EGO (ParEGO) algorithm [41], Keane's statistically based improvement (KSI) method [42], meta-model assisted evolution strategies (MAES) [12], and the predictive entropy search (PES) [43] extend these works to solve multi-objective problems. Furthermore, the parallelized multi-objective EGO (ParMOEGO) algorithm [44] seeks to improve efficiency of these methods even more by parallelizing high-fidelity computations to reduce "wall-clock" time. Another extension of these efficient algorithms, the Multi-Objective Bayesian Optimization (MOBO) algorithm [45], allows for the efficient optimization of problems with multiple information sources. Figure 2.3 categorizes these algorithms alongsie the new algorithms discussed in this work: EPO and MO-EPO.

2.2.1 Efficient Global Optimization (EGO) Algorithm

The efficient global optimization algorithm (EGO) is a well-known methodology for single objective efficient optimization. EGO iteratively updates and uses response surface models to efficiently search the design space in order to reduce the number of expensive function evaluations required to reach the global optimum [38]. To accomplish this, the expensive (or high-fidelity) function is evaluated a number of times via a design of experiments (DOE) to train an initial



	Single Objective	Multiple Objectives
Parametric	EPO[11] (Ch. 4)	MO-EPO (Ch. 6)
Non-Parametric	EGO[38] GRBF[39] SAO[40]	ParEGO[41] KSI[42] MAES[12] PES[43] ParMOEGO[44]

Figure 2.3: A list of optimization methods that rely on adaptive sampling of a response surface sorted by problem type.

response surface. Then, the less expensive response surface model is searched via branch and bound optimization to find the design with the highest expected improvement based on the response surface prediction of objective value and its uncertainty. This expected improvement formulation is [38]:

$$EI(\boldsymbol{x}) = (J_{min} - \hat{J}(\boldsymbol{x}))\Phi\left(\frac{J_{min} - \hat{J}(\boldsymbol{x})}{s(\boldsymbol{x})}\right) + s(\boldsymbol{x})\phi\left(\frac{J_{min} - \hat{J}(\boldsymbol{x})}{s(\boldsymbol{x})}\right), \tag{2.6}$$

where J_{min} is the minimum objective value found so far (from the high-fidelity samples) and $\hat{J}$ and s are the response surface model predictions of the objective value and standard error, respectively, for given optimization variables $\boldsymbol{x}$. Additionally, Φ represents the standard normal cumulative distribution function, and ϕ is the standard normal probability distribution function. This expected improvement function incorporates both *exploiting* regions with predicted optimal response and *exploring* regions with large uncertainty that might result in an optimal response [38].

Finally, the design with the greatest expected improvement is evaluated by the expensive model, and the response surface model is subsequently improved. In this way, the entire high-fidelity design space is sampled in an adaptive way using the lower-fidelity response surface model to predict beneficial and/or uncertain designs. While this method requires more steps and multiple optimizations of a response surface model, the number of times that the expensive model is sampled can be drastically reduced when compared to a direct optimization of the expensive model. See Algorithm 1 for a more compact representation of this algorithm.

Algorithm 1 EGO[38].

```
$J(\boldsymbol{x}) \leftarrow$ Expensive model
$\hat{J}(\boldsymbol{x}) \leftarrow$ Initial response surface
$N_{eval} \leftarrow$ Number of expensive evaluations performed
**input:** $N_{eval}^{max} \leftarrow$ Maximum allowable expensive evaluations
$J_{min} \leftarrow$ Minimum objective value from all expensive samples
**while** $N_{eval} < N_{eval}^{max}$ **do**
    $\boldsymbol{x}_{new} \leftarrow \text{argmax}_{\boldsymbol{x}}\ EI(\boldsymbol{x})$
    **if** $EI(\boldsymbol{x}_{new}) < tol$ **then**
        **break while**
    **else**
        $J_{new} \leftarrow J(\boldsymbol{x}_{new})$
        $\hat{J}(\boldsymbol{x}) \leftarrow$ Update response surface with $\boldsymbol{x}_{new}$ and $J_{new}$
        $N_{eval} \leftarrow N_{eval} + 1$
        **if** $J_{new} < J_{min}$ **then**
            $J_{min} \leftarrow J_{new}$
        **end if**
    **end if**
**end while**
**return** $J_{min}$
```

2.2.2 Multi-Objective Efficient Optimization

Knowles introduced a multi-objective version of EGO called Pareto EGO (or ParEGO) [41]. The ParEGO algorithm (Algorithm 2) follows the same general procedure as EGO with one minor yet consequential difference. Within each iteration of the algorithm, a trade-off vector $\boldsymbol{\lambda}$ is chosen at random to combine the multiple objectives into a single weighted sum function. Then, the response surface is trained with this weighted sum as the single output, and the general procedure of EGO is resumed. That is, the expected improvement function is maximized, and a new design is chosen to be sampled by the expensive model. The underlying idea is that given enough iterations, the whole Pareto frontier will be found by sampling enough trade-off vectors and gathering enough data to generate accurate response surface models. Results show that this method is more efficient than NSGA-II [36] on most of the problems considered in Ref. [41]. The main limitation of this work is its reliance on a weighted sum objective formulation. It is widely known that finding a full Pareto frontier through iterative weighted sum optimizations does not work well when the Pareto frontier is not convex. Additionally, some parts of the Pareto frontier may be easier to find than others as acknowledged in Ref. [41].

Algorithm 2 ParEGO[41].

```
$\mathbf{J}(\boldsymbol{x}) \leftarrow$ Expensive model
$N_{eval} \leftarrow$ Number of expensive evaluations performed
input: $N_{eval}^{max} \leftarrow$ Maximum allowable expensive evaluations
while $N_{eval} < N_{eval}^{max}$ do
    $\boldsymbol{\lambda} \leftarrow$ Choose normalized objective trade-off vector
    $\hat{f}(\boldsymbol{x}, \boldsymbol{\lambda}) \leftarrow$ Train response surface for weighted sum given $\boldsymbol{\lambda}$
    $f_{min} \leftarrow$ Identify minimal weighted sum value for given $\boldsymbol{\lambda}$
    $\boldsymbol{x}_{new} \leftarrow$ argmax$_{\boldsymbol{x}}$ $EI(\boldsymbol{x})$
    if $EI(\boldsymbol{x}_{new}) < tol$ then
        break while
    else
        $\mathbf{J}_{new} \leftarrow \mathbf{J}(\boldsymbol{x}_{new})$
        $N_{eval} \leftarrow N_{eval} + 1$
    end if
end while
return $J_{min}$
```

Emmerich *et al.* [12, 46] outline a generic multi-objective procedure for efficient optimization. This procedure follows the same mathematical background as EGO, but directly considers multiple objectives via expected improvement of the dominated hypervolume indicator (HVI) via Monte Carlo sampling. The dominated hypervolume indicator (HVI) gives an indication of how much of a reference space is “covered” or dominated by a given data set. It was first used by Zitzler and Thiele [47] to compare the solutions of evolutionary algorithms on multi-objective problems.

The hypervolume indicator is also referred to in literature as the hypervolume measure [47], hyperarea metric [48], S-metric [49], and Lebesgue measure [50, 51]. Two desireable properties of the hypervolume indicator are that the metric is sensitive to improvement, and unaffected by non-improvement [52, 46]. These properties ensure that the true solution to a multi-objective problem is the solution with the greatest hypervolume indicator.

The multi-objective expected improvement formulation presented by Emmerich *et al.* based upon this hypervolume indicator is [12]:

$$EI_{\mathcal{H}}(\boldsymbol{x}) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{H}(\mathbf{S} \cup q_i) - \mathcal{H}(\mathbf{S})\right), \tag{2.7}$$

where $\mathcal{H}(\cdot)$ is the dominated hypervolume of a given data set, $\mathbf{S}$ is the current data set, and q_i is the response of the *i*-th Monte Carlo sample of a new point $\boldsymbol{x}$ given its response distribution. Therefore, the expected improvement of a new point $\boldsymbol{x}$ is approximated by the expected increase of the dominated hypervolume indicator. Furthermore, the properties of the hypervolume indicator discussed previously guarantee that Eqn. (2.7) will yield either a value of exactly zero if the response distribution at $\boldsymbol{x}$ is fully dominated by $\mathbf{S}$ or a positive value otherwise. It is not possible for Eqn. (2.7) to have a negative result [12, 46]. With this multi-objective expected improvement formulation, the same general procedure of EGO can be used to efficiently solve multi-objective problems as outlined in Algorithm 3 [46].

The method proposed by Emmerich *et al.* [46] is used in Chapter 6 to develop an efficient multi-objective parametric optimization algorithm. The benefit of this method over a method such as ParEGO is its ability to consider all objectives independently and simultaneously. In a similar manner, the proposed procedure of MO-EPO considers all objectives and all parameters independently and simultaneously to adaptively search the region of greatest expected improvement. Refer to Chapter 6 for further details on the new MO-EPO.

Algorithm 3 Multi-objective efficient optimization via HVI-based expected improvement [46].

```
$\mathbf{J}(\boldsymbol{x}) \leftarrow$ Expensive objective models
$\hat{\mathbf{J}}(\boldsymbol{x}) \leftarrow$ Initial response surface models of objectives
$N_{eval} \leftarrow$ Number of expensive evaluations performed
$\boldsymbol{x}^{nd} \leftarrow$ store nondominated optimization variables
$\mathbf{J}^{nd} \leftarrow$ store nondominated objective
**input:** $N_{eval}^{max} \leftarrow$ Maximum allowable expensive evaluations
**while** $N_{eval} < N_{eval}^{max}$ **do**
    $\boldsymbol{x}_{new} \leftarrow \text{argmax}_{\boldsymbol{x}}\ EI_{\mathcal{H}}(\boldsymbol{x})$
    **if** $\text{all}(EI_{\mathcal{H}}(\boldsymbol{x}_{new}) < tol)$ **then**
        **break while**
    **else**
        $\mathbf{J}_{new} \leftarrow \mathbf{J}(\boldsymbol{x}_{new}^{(i)})$ for some $i^{th}$ tuple of $\boldsymbol{x}_{new}$
        $\hat{\mathbf{J}}(\boldsymbol{x}) \leftarrow$ Update response surface with $\boldsymbol{x}_{new}^{(i)}$ and $\mathbf{J}_{new}$
        $\boldsymbol{x}^{nd} \leftarrow$ update nondominated optimization variables
        $\mathbf{J}^{nd} \leftarrow$ update nondominated objective values
        $N_{eval} \leftarrow N_{eval} + 1$
    **end if**
**end while**
**return** $\boldsymbol{x}^{nd}, \mathbf{J}^{nd}$
```

3. PARAMETRIC OPTIMIZATION FOR MORPHING STRUCTURES*

3.1 Motivation

As introduced in Chapter 1, parametric optimization is uniquely suited to solving an optimal morphing problem by treating variable external conditions as parameters. This chapter details the first application of parametric optimization to the optimization of an adaptive structure. In a morphing aerostructure problem, parameter variables can include flight conditions that affect aircraft performance such as airspeed, angle of attack, and altitude. So long as the parameter values can be sensed during flight, the system can morph to the optimal solution that has been presolved via parametric optimization.

To be efficient, parametric optimization requires specialized optimization codes. Although it is possible to iterate a traditional optimization algorithm at different values for parameters, this approach does not scale well. Gamboa *et al.* [4] applies such a sampling approach to the parametric optimization of a morphing wing, but deals with only one parameter (airspeed). For problems with additional parameters (multi-parametric problems), the number of traditional optimization routines required would grow prohibitively large for this approach. This work uses the Predictive Parametric Pareto Genetic algorithm (P3GA) introduced by Galvan and Malak [10]. P3GA can handle nonlinear black-box functions and is capable of solving single- and multi-objective parametric problems. A comparative study demonstrated that P3GA is superior to iterated application of a traditional optimization algorithm [5]. Previous implementations of P3GA focus on solving problems where the parameter values are not yet fully known but will be constant during operation [7, 5]. In the case study presented here, the parameter values are continuously changing during operation, and the structure must optimally morph in response to these changing parameters.

Optimal morphing is particularly appealing in aircraft design, where performance is critical and

*This chapter is reprinted with permission from "Parametric optimization for morphing structures design: application to morphing wings adapting to changing flight conditions" by Weaver-Rosen, J.M., Leal, P.B.C., Hartl, D.J., and Malak, R.J., 2020. *Structural and Multidisciplinary Optimization*, 62, 2995-3007 Copyright 2020 by Springer Nature Switzerland AG [2].

often costly. Morphing aero-structures are defined in this work as those that can C^1-continuously change their outer mold line (OML). Morphing expands the space of traditional aircraft control and trim methods, such as flaps and slats, to include smooth configurations with improved aerodynamic performance [3]. Aviation can benefit from this technology to minimize drag [3] and noise [53] or to improve aircraft maneuverability [54]. However, the benefits of continuous shape change are not intuitive for each flight condition and have not yet been fully explored.

Although there has been extensive work in optimizing an aircraft OML to improve overall performance, most methods have one or more of the following drawbacks: (*i*) no structural constraints are considered when defining geometry [55, 56, 57], (*ii*) no analytical C^1 geometric representation exists [57, 58], and/or (*iii*) the design is optimized for only a limited number of flight conditions [3, 55, 56, 57, 59, 60]. Optimized configurations that do not consider (*i*) will result in thin airfoils that are infeasible to manufacture while (*ii*) hinders the use of gradient-based methods or sensitivity analyses and results in high-drag OML's. Drawback (*iii*) leads to designs that perform well at specific flight conditions but that underperform for all other conditions. Some have addressed (*i*) using thickness [57, 59] and volume [58, 60] constraints on the OML that reproduce the inner structure restrictions. Recently, Leal and Hartl have addressed both (*i*) and (*ii*) using a structurally-consistent approach based on Class/Shape Transformation (CST) equations [61]. Additionally, parametric optimization addresses (*iii*) by treating flight conditions as parameters. This current work combines the CST approach with parametric optimization to yield a comprehensive approach to optimal morphing wing design. This work is the first application of parametric optimization to morphing wings in particular and morphing structures in general.

Section 3.2 gives a recap of parametric optimization and the specific techniques used in this study, and section 3.3 contains a brief description of the structurally consistent CST-based morphing wing model. Next, the morphing wing optimization problem formulation is presented (section 3.4) along with results (section 3.5). Finally, the parametric results are benchmarked against solutions obtained from a traditional optimization applied at samples of the parameter values (section 3.6). Comparison results indicate that parametric optimization using P3GA is an effective approach for determining the optimal morphed structure across a range of operating conditions surpassing a traditional approach.

3.2 Parametric Optimization

3.2.1 Theory

As discussed in Section 2.1, parametric optimization solves a traditional optimization problem while directly considering parameter variables. Recall the general parametric problem formulation in Eqn. (2.1). In this case study, one objective is considered with three independent parameters. Therefore, the problem solution is the *performance model* $J^*(\boldsymbol{\theta})$ (i.e., the single objective version of the parameterized Pareto frontier) and the optimal solution map $\boldsymbol{x}^*(\boldsymbol{\theta})$.

The performance model can be used to determine how parameters affect optimal performance and assess which parameters are most critical. The optimal solution map provides a designer with the optimal design configuration given a vector of parameter values. In a morphing context, this optimal solution map removes the need to perform a computationally expensive optimization algorithm during operation as conditions change. Instead, the optimal solution map $\boldsymbol{x}^*(\boldsymbol{\theta})$ is directly used to determine the optimal morphed shape during operation under changing conditions.

In the case of morphing aircraft, optimization variables include the variables that control wing OML (i.e. shape variables) during operation, while parameters could include variables such as airspeed, angle of attack, altitude, temperature, air density, and others. In most cases, it is unlikely that a pilot will know all of these parameter values for a given flight plan at take-off, and many parameters are subject to change during flight. Using the proposed parametric optimization technique, a pilot or a controls system is always informed of all optimal lift-to-drag ratios and corresponding shape variables for a vector of parameters describing the flight condition.

3.2.2 Methods

The optimization algorithm used in this work, P3GA, directly outputs the discrete configurations (i.e., set of optimization variables and corresponding parameters) found to be optimal in addition to an implicit representation of the performance model. A Gaussian based process such as a Kriging interpolation [62] can approximate the optimal solution map from the optimal configurations found with P3GA as demonstrated by Hartl *et al.* [7] under the assumption that the optimal solution map both exists and is unique. Therefore, $\hat{\boldsymbol{x}}^*(\boldsymbol{\theta})$ will denote the approximate optimal solution map via Kriging interpolation. For the case study herein, the DACE toolbox [63] is used to train $\hat{\boldsymbol{x}}^*(\boldsymbol{\theta})$ using the optimal designs from P3GA and a Gaussian correlation function and a quadratic regression model (see Lophaven *et at.* [63] for details on training a Kriging model).

3.3 Camber Morphing Wing Description

The design problem considers the cross section of an infinite wing (i.e., an airfoil) undergoing camber morphing as illustrated by Fig. 3.1. The CST method is used to mathematically define all generated airfoils [64]. This method has been shown to accurately represent any realistic airfoil geometry with a minimum number of variables [65]. Leal and Hartl [61] modified the CST equations so that children airfoils (i.e., morphed configurations) can be generated from any parent airfoil (i.e., original configuration) while considering kinematic constraints related to the internal structure; thus, all of the explored designs herein are known to be physically feasible for low strains ($\leq 4\%$), even without recourse to full structural analysis.

In the context of real aircraft, the parent airfoil and much of its internal structure are fixed and known. Therefore, the children airfoils are only a function of shape coefficients $\mathbf{P}^c$, used herein as the optimization variables. For this work, the shape coefficients of a NACA 0012 airfoil using fourth-order Bernstein polynomials are used and denoted in Table 3.1. A symmetric airfoil is chosen to demonstrate that morphing is effective even for an airfoil with initially no camber, even as compared to other examples of non-morphing airfoils as will be shown. The non-dimensional

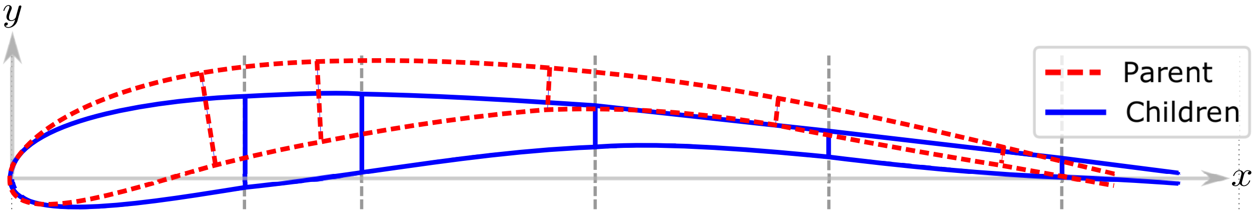


Figure 3.1: Morphing airfoil example.

location of spars are also fixed at 0.1, 0.3, 0.6, and 0.8. Refer to Appendix B for further details on the CST equations.

Table 3.1: Shape coefficients for parent (reference) NACA 0012 airfoil considered in this work from which all morphed configurations are generated.

P_0^p	P_1^p	P_2^p	P_3^p	P_4^p
0.1828	0.1179	0.2079	0.0850	0.1874

A_0^p	A_1^p	A_2^p	A_3^p	A_4^p
0.1702	0.1528	0.1592	0.1195	0.1651

3.4 Parametric Formulation of a Morphing Aerostructure Design Problem

The optimal morphing aero-structure problem can be formulated as a multi-parametric optimization problem, where the single objective is to maximize the lift-to-drag ratio. This ratio is determined by calculating the coefficients of lift and drag using XFOIL [66], a subsonic panel-method aerodynamic solver. The operational parameters considered are angle of attack α, airspeed V, and altitude H, all of which vary during flight. For this work, the optimization variables are the shape coefficients of the child (morphed) airfoil P_i^c for $i = [1, 2, 3, 4]$, where each coefficient is spatially predominant in a different chordwise location. Together, the four shape coefficients define the airfoil shape (see Table 3.1 for parent shape coefficients $\mathbf{P}^p$). P_1^c and P_4^c dominate respectively the leading and trailing edges, while P_2^c and P_3^c are most relevant for intermediate chordwise coordinates. All other aspects of the problem, such as the initial NACA 0012 shape and spar locations are held constant to allow clarity of presentation. CST equations of order $n = 4$ are utilized based on a convergence study performed in Ref. [61]. Depending on the requirements of the design exploration, n can be increased to allow more subtle topological manipulations and structural constraints (e.g., spars).

The mathematical problem formulation is thus:

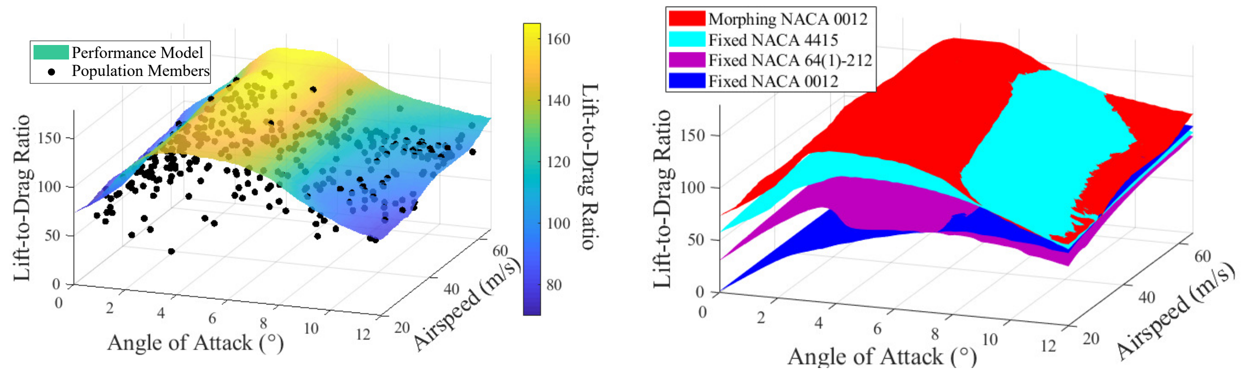
$$
\begin{aligned}
&J^*(\alpha, V, H) = \max_{\mathbf{P}^c} \; J(\mathbf{P}^c; \alpha, V, H), \\
&\qquad\qquad \text{Subject to:} \\
&\qquad\qquad\qquad \mathbf{P}^c \in \mathbb{P}, \\
&\begin{bmatrix} 0^\circ \\ 20 \text{ m/s} \\ 5000 \text{ ft} \end{bmatrix} \leq \begin{bmatrix} \alpha \\ V \\ H \end{bmatrix} \leq \begin{bmatrix} 12^\circ \\ 65 \text{ m/s} \\ 40000 \text{ ft} \end{bmatrix},
\end{aligned}
\tag{3.1}
$$

where the objective function J represents the lift-to-drag ratio, $J^*(\alpha, V, H)$ represents the maximum lift to drag ratio as a function of the parameters, $\boldsymbol{x} = \mathbf{P}^c$, and $\boldsymbol{\theta} = [\alpha, V, H]^T$.

To prevent generating shapes that either are unreasonable or cannot be accurately studied with XFOIL, the shape coefficients are conservatively constrained based on feasible, existing airfoil shapes. This feasible set is denoted $\mathbb{P}$. To quantify this constraint, 1,636 shape coefficient vectors are calculated for all the airfoils contained in a given database [67]. A thorough investigation of this four-dimensional data revealed that the feasible space would not be easily constrained via linear constraints. As shown by Malak and Paredis [68], a Support Vector Domain Description (SVDD) can be trained to fit the data and provide a single, nonlinear constraint. This SVDD is a one-class classifier that, once trained, predicts whether a new data point lies inside or outside of the class [69]. In this case, the class being trained is that of feasible airfoil shapes.

3.5 Results

Since the solution of a single-objective three-parameter problem cannot be readily visualized in 4-D space, results are herein presented in two phases. Section 3.5.1 presents a two-parameter subset of the solution where altitude is fixed at 10,000 feet while angle of attack and airspeed are still treated as parameters. Then, section 3.5.2 conveys the full, three-parameter solution.



(a)

(b)

Figure 3.2: Results considering altitude held constant at 10,000 ft. (a) Population of structurally feasible morphed designs projected into this 3-D domain and lift-to-drag ratios of optimal performance across changing flight conditions. (b) Comparison between the performance of the morphing NACA 0012 configurations and that of a fixed NACA 0012, a fixed NACA 4415 and a fixed NACA 64(1)-212.

3.5.1 Two Parameter Problem

Considering a fixed altitude of 10,000 ft, the performance model over changing angle of attack and airspeed is illustrated by the surface in Fig. 3.2a. Depending on the parameter values, lift-to-drag ratios of up to 169 are achieved. Overall, the optimal lift-to-drag ratio monotonically increases as airspeed V increases. An increase of angle of attack for smaller angles ($\alpha \leq 5°$) results in an increase of this objective. However, for higher angles of attack ($\alpha > 5°$), the lift-to-drag ratio decreases because of sensitivity to stalling [66]. Fig. 3.2b shows a comparison between the performance for the camber *morphing* NACA 0012 and three *fixed* airfoils for the same flight conditions (including the fixed NACA 0012). The benefits of using a morphing airfoil are apparent at smaller angles of attack where significant increases in achievable lift-to-drag ratios are obtained. For higher angles of attack, the morphing optimization seems to be less relevant for this parent airfoil. Although there is one region ($\alpha \in (7°, 11°)$) where the fixed NACA 4415 dominates, the morphing NACA 0012 appears to outperform all considered fixed airfoils for the majority of these flight parameters. The maximum benefit seen by the fixed NACA 4415 over the morphing NACA 0012 is an increase of 18.8 of the lift-to-drag ratio. However, the lift-to-drag ratio of the morphing

NACA 0012 is on average 9 points higher than that of the fixed NACA 4415. Additionally, the morphing NACA 0012 always outperforms the fixed NACA 4415 for the lower angles of attack that airplanes more commonly operate. This result shows that the morphing NACA 0012 is physically unable to match the performance of the NACA 4415 shape due to the structurally consistent constraints. This gives evidence to support analyzing a *morphing* NACA 4415 concept especially when looking at higher angle of attack applications.

The optimal morphed configuration clearly changes with flight condition, but four distinct families can be clustered based on their topology [70]. Fig. 3.3 illustrates how the performance model and optimal solution map can be used together to display these four families of airfoils. The depicted airfoils are the cluster centers [70] for each family of solutions. At lower angles of attack ($\alpha < 4°$), the morphed configurations (design A) are similar to traditional high-lift airfoils [71]. As angle of attack increases ($4° \leq \alpha < 7°$), the maximum camber location tends towards the leading

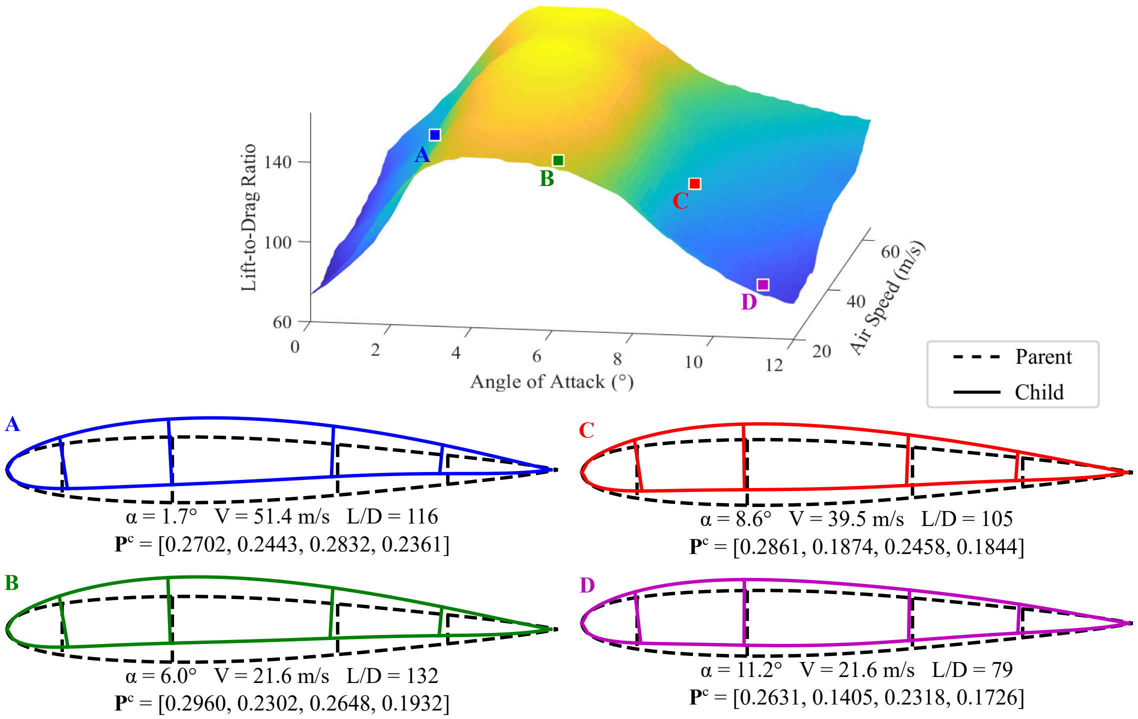


Figure 3.3: Four structurally consistent morphed configurations from the optimal solution map and their relation to the performance model are depicted along with airfoil shape coefficients, designed flight condition, and lift-to-drag ratio.

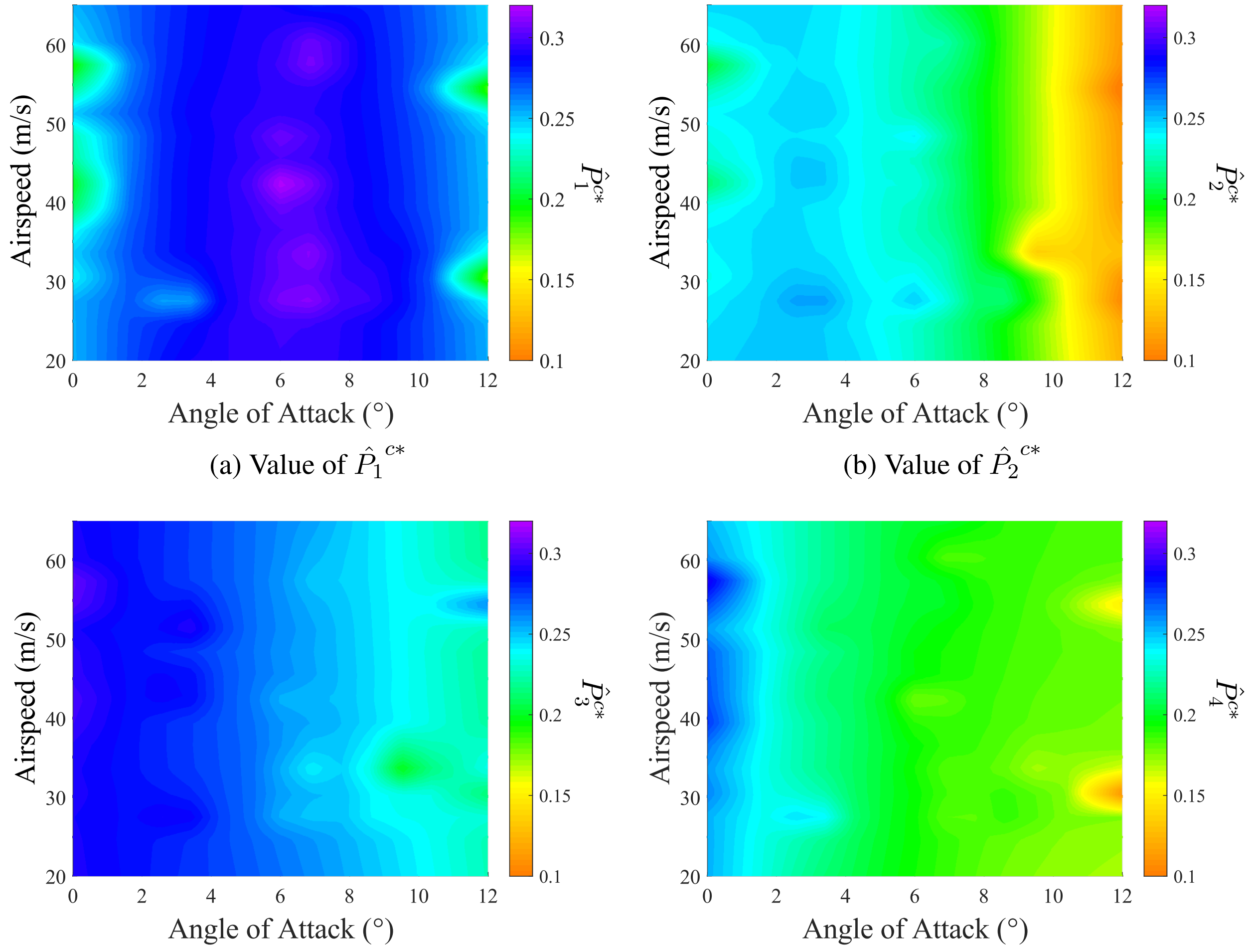
edge, resulting in configurations (design B) similar to avian airfoils [72]. As angle of attack continues to increase (7° $\leq \alpha <$ 9°), the high camber at the front of the airfoil decreases, leading to traditional airfoil shapes (design C). At the highest angles of attack, camber is decreased to delay boundary layer separation, leading to unorthodox designs (design D) known as reflexed airfoils [73, 74]. Each of the identified designs are not acceptable for all flight conditions, but are viable as one of many states of a morphing airfoil.

Fig. 3.4 illustrates the optimal solution map approximation, $\hat{\mathbf{P}}^{c*}(\alpha, V, H)$, for each shape coefficient over the subset of the parameter space where $H =$ 10,000 ft. For lower angles of attack ($\alpha <$ 7°), $\hat{P_1}^{c*}$ increases, $\hat{P_2}^{c*}$ fluctuates near its highest value, and $\hat{P_3}^{c*}$ and $\hat{P_4}^{c*}$ decrease to augment camber and relocate the maximum camber location towards the leading edge. At higher angles of attack ($\alpha \geq$ 9°), all shape coefficients decrease to curtail camber, generating children configurations similar to the parent airfoil.

3.5.2 Three Parameter Problem

Although the four-dimensional performance model (three parameters and one objective) can be difficult to visualize directly, depicting multiple two-dimensional subsets for a fixed aircraft speed and altitude is enlightening (Fig. 3.5). Fig. 3.5a displays the two-dimensional solution subsets at four different airspeeds while fixing altitude at 20,000 ft. Likewise, Fig. 3.5b displays the two-dimensional solution subsets at four different altitudes with airspeed fixed at 35 m/s.

Visualizing these reduced subsets of the four-dimensional performance model reiterates the trend seen in the two-parameter solution, but also elucidates how this model is affected by altitude variations. The lift-to-drag ratio monotonically increases as airspeed increases while the effect of angle of attack is not as straightforward. Angle of attack continues to maximize the lift-to-drag ratio near 5 °, but variations in altitude change this peak slightly. While the curves for 10,000 and 20,000 ft are similar, the performance begins to decrease more rapidly for altitudes of 30,000 and 40,000 ft. This is the expected response as air density decreases at higher altitudes, leading to varying Reynolds numbers (i.e., different flow regimes). Note that the performance model somewhat deviates from these trends when the angle of attack is near 2° and 8°, which may be



(a) Value of $\hat{P}_1{}^{c*}$

(b) Value of $\hat{P}_2{}^{c*}$

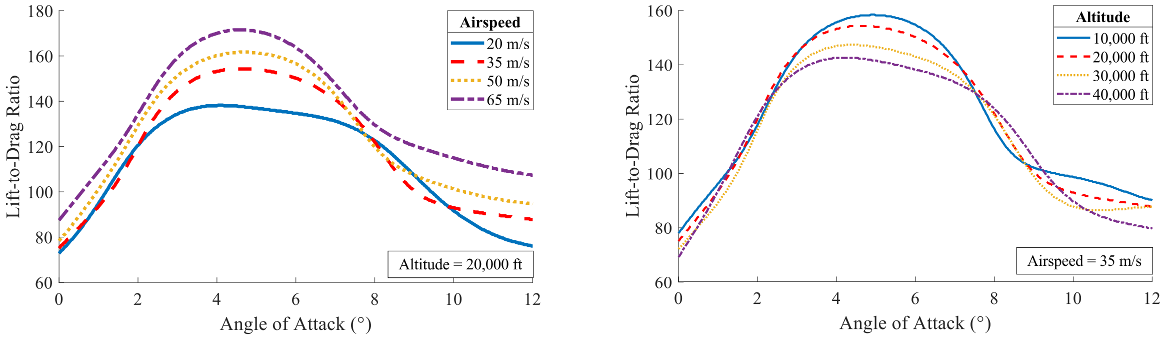
(c) Value of $\hat{P}_3{}^{c*}$

(d) Value of $\hat{P}_4{}^{c*}$

Figure 3.4: Approximate optimal solution map providing the value of each shape coefficient for the camber morphing NACA 0012 when altitude is fixed at 10,000 ft. Such a map could be used to schedule a controller such that morphing structures automatically adapt during operation.

an artifact of aerodynamic instabilities in these regions.

Fig. 3.6 displays the shape coefficient values (cf. Fig. 3.4) with angle of attack and altitude featured on the horizontal and vertical axes, respectively, and airspeed fixed at 35 m/s. The subset of the optimal solution map represented by Fig. 3.6 shows that altitude has very little effect on the optimal morphed airfoil shapes, even though the optimal *performance* of these airfoils is affected (cf. Fig. 3.5). In the interest of brevity, only a small subset of the full optimal solution map is visualized.



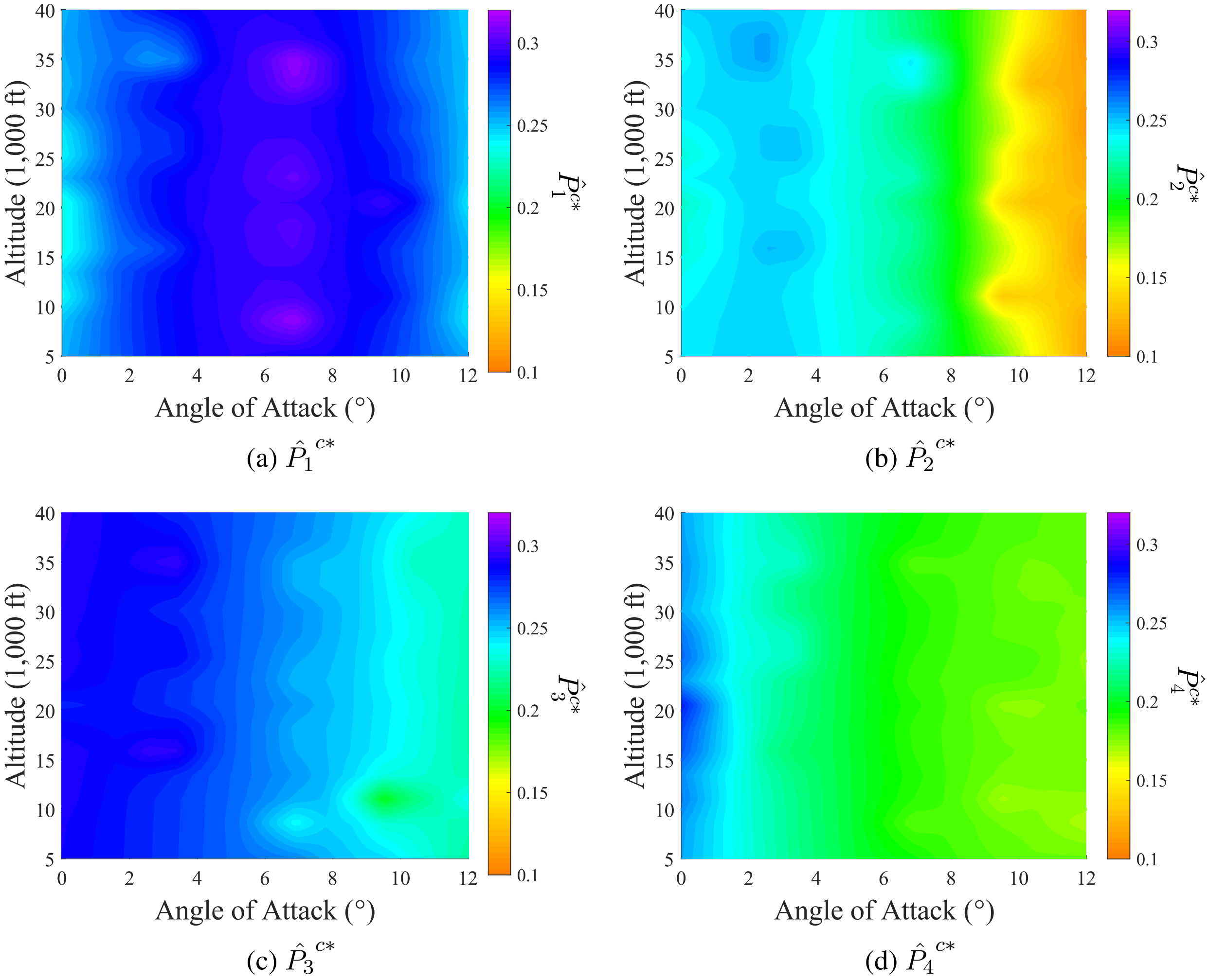
(a) (b)

Figure 3.5: Two-dimensional subsets of the four-dimensional performance model for the morphing airfoil: (a) sections of this model where altitude is fixed at 20,000 ft and airspeed is sampled at 20, 35, 50, and 65 m/s; (b) sections of the same model fixing airspeed at 35 m/s and aircraft altitude at 10, 20, 30, and 40 thousand feet.

With respect to computational cost, the runtime for the full parametric optimization loop that solves Eqn. (3.1) for all three parameters using P3GA is approximately 117 hours for 400 generations with a population size of 150 individuals (Intel i7 core processor with 8 GB of RAM). The majority of this runtime is spent evaluating the objective function via XFOIL, which requires 6 to 20 seconds depending on the aerodynamic complexity. In this case, the computational cost of the methods used within P3GA (i.e., machine learning) are negligible compared to objective function cost.

3.6 Benchmarking

To quantify any errors in the performance model, its output is compared to that from multiple executions of a traditional genetic algorithm (each with a different set of fixed parameter values). 64 parameter combinations selected via a Latin Hypercube sampling (LHS) for the morphing NACA 0012 parent shape are considered and represent 64 single-objective individual optimizations to obtain the maximum lift-to-drag ratio, $J^\flat$, and optimal shape coefficients, $P_i^{c\flat}$, using the built-in GA in Matlab [37]. Default algorithmic settings are used, and the results are taken as truth solutions as convergence to within a small tolerance occurs. Note that the solutions to the 64 genetic optimizers required 1,344 hours in total, though direct comparison to P3GA's runtime of



(a) $\hat{P}_1^{c*}$

(b) $\hat{P}_2^{c*}$

(c) $\hat{P}_3^{c*}$

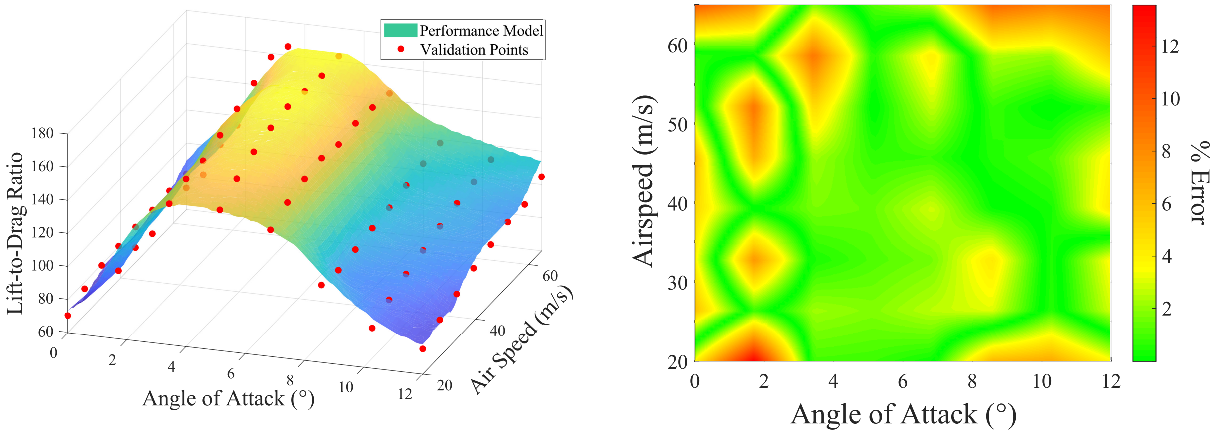
(d) $\hat{P}_4^{c*}$

Figure 3.6: Approximate optimal solution map providing the value of each shape coefficient for the camber morphing NACA 0012 when airspeed is fixed at 35 m/s.

117 hours is inconclusive since P3GA did not enforce any stopping criteria beyond the maximum number of generations[1]. All optimizations performed in this study (via P3GA and a traditional genetic algorithm) are done using a quad-core computer with a 2.93 GHz Intel i7-870 processor and 8 GB of RAM. The error of the performance model is quantified using Eqn. (3.2).

$$\text{Error}\left(\boldsymbol{\theta}\right) = \frac{\left|J^{*}(\boldsymbol{\theta}) - J^{\flat}(\boldsymbol{\theta})\right|}{\left|J^{\flat}(\boldsymbol{\theta})\right|} \tag{3.2}$$

[1] Although genetic algorithms (including P3GA) are highly parallelizable, complications with the specific implementation of XFOIL used herein required serial calculations. Therefore, all times presented in this chapter do not consider potential time savings for running analyses in parallel.



(a)

(b)

Figure 3.7: Results of parametric optimization via P3GA contrasted with the results of the traditional genetic algorithm [1]: (a) the performance model (surface) as compared to the 8 x 8 grid of traditional optimization solutions (points), (b) percent error between the performance model and validation points as calculated via Eqn. (3.2).

The average and standard deviation of the error in performance among the 64 fixed parameter sites considered are 2.85% and 3.30%, respectively. To provide a graphical representation, another study is performed where a two-parameter subset of the full solution is explored to allow for visualization in three dimensions (two parameters and one objective). Fixing the altitude parameter at 10,000 ft, an 8 x 8 grid of combinations of angle of attack and airspeed are chosen to run a total of 64 *additional* iterations of a traditional genetic algorithm. These 64 combinations are plotted in red over the performance model in Fig. 3.7a for this fixed altitude. P3GA is able to represent the trends of the true solution very well. The errors (defined in Eqn. (3.2)) of this subset are largely within 5%, as shown in Fig. 3.7b.

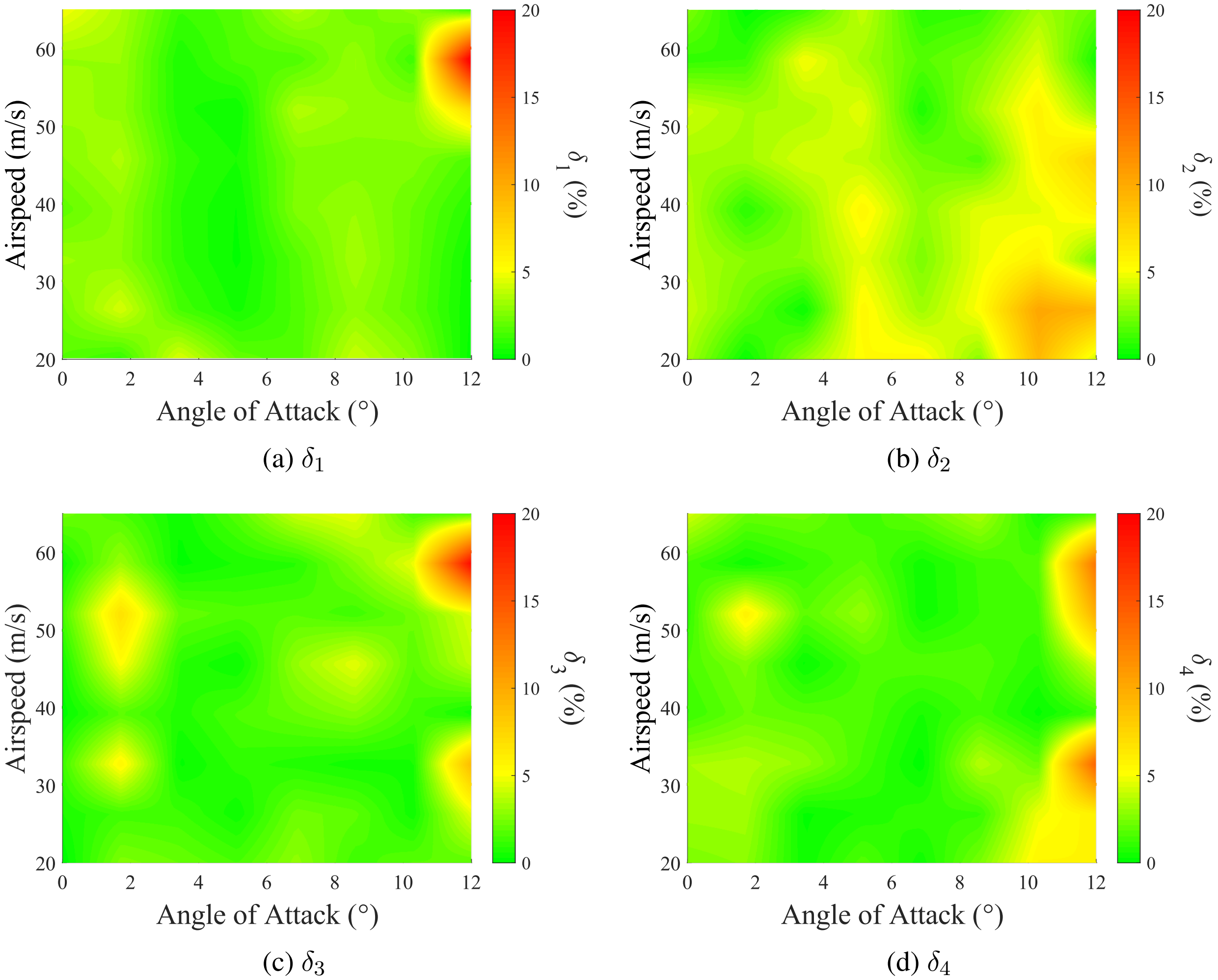
Finally, the full optimal solution map from parametric optimization is benchmarked against the discrete optimal configurations for the original 64 points in parameter space via LHS. The differences between the shape coefficients from the optimal solution map and those from fixed-parameter optimizations are given by:

$$\delta_i\left(\boldsymbol{\theta}\right) = \frac{\left|\hat{P}_i^{c*}(\boldsymbol{\theta}) - P_i^{c\flat}(\boldsymbol{\theta})\right|}{x_{ub,i} - x_{lb,i}} \ \forall \ i \in [1, 2, 3, 4]\,, \tag{3.3}$$

where $\hat{P}_i^{c*}(\boldsymbol{\theta})$ is the optimal i-th shape coefficient predicted by the Kriging model of the optimal solution map and $P_i^{c\flat}(\boldsymbol{\theta})$ is the optimal i-th shape coefficient predicted by the fixed-parameter optimizations via a common GA [1]. This difference is normalized by the bounds of shape coefficients which are determined by the 1600 feasible airfoils and introduced via the nonlinear constraint. All shape coefficients are allowed to vary from ± 0.5; thus, the denominator of Eqn. (3.3) equals one. The average differences for each of the four shape coefficients are 2.38%, 3.48%, 1.85%, and 2.34%, respectively. To allow for visualization of these differences, a subset of the optimal solution map at $H = 10,000$ ft is visualized for each shape coefficient based on the same 8 x 8 grid over angle of attack and airspeed from before (Fig. 3.8). This agreement gives evidence to support the use of parametric optimization in a morphing context as a preliminary step in calculating the optimal shape variables for many sets of external conditions.

3.7 Conclusion

This work shows the potential of parametric optimization as applied to the design of a morphing structure. P3GA is shown to be an effective tool for optimizing a flight performance objective function over a range of parameters, making it a useful method for optimizing morphing structures. The results of parametric optimization using P3GA are found to be within a reasonable error tolerance as compared to traditional methods, with the advantage of capturing a more detailed understanding of the parameter space at a reduced computational cost. This accuracy, coupled with the decreased runtime compared to traditional methods [5], supports the use of parametric optimization for solving morphing problems over traditional optimization methods. Visualizing the parametric solutions (performance model and optimal solution map), even when the solution dimensionality is greater than three, is demonstrated by viewing subsets. The full performance model is found to have an average error of approximately 2.85%, and the optimal solution map has average errors of less than 3.5% for each of the four optimization variables (shape coeffi-



(a) δ_1

(b) δ_2

(c) δ_3

(d) δ_4

Figure 3.8: Difference between the results of the optimal solution map and the traditional genetic algorithm [1] from the four optimization variables (P_1^c, P_2^c, P_3^c, and P_4^c) using Eqn. (3.3) for the two-parameter subset where altitude is fixed at 10,000 ft.

cients). The two-parameter performance model describing the morphing NACA 0012 lift-to-drag ratios compared to those from non-morphing baseline designs motivates morphing technologies and shows that morphing airfoils can drastically improve wing performance and achieve similar or better performance across changing operational conditions than fixed, specialized airfoils.

Now that parametric optimization has been shown to effectively solve adaptive structure optimization problems, a more efficient method is needed. In this work, parametric optimization of the three-parameter, single objective problem required 60,000 design evaluations for a total runtime of 117 hours. This is an expensive optimization problem considering that the evaluation of a single design takes between 7 and 20 seconds. Chapter 4 presents the first efficient parametric optimization algorithm to allow problems with more expensive analyses to be performed in a more reasonable amount of time.

4. EFFICIENT PARAMETRIC OPTIMIZATION*

In previous chapters the motivation for parametric optimization has been explained and demonstrated for the design of a morphing airfoil across changing conditions (cf. Chapter 3). However, parametric optimization like many traditional optimization techniques can become computationally intractable when exposed to problems requiring expensive evaluations (e.g., CFD, FEA, etc.). In traditional optimization, there are multiple methods that have been employed to allow for the more efficient optimization of such problems with expensive analyses (cf. Section 2.2). However, the special role of parameter variables precludes the direct use of these techniques for parametric optimization.

In this chapter, the well-known EGO algorithm (cf. Algorithm 1) is extended to parametric optimization problems. The new technique formulates a parametric version of the expected improvement function and uses the existing parametric optimization technique, the Predictive Parameterized Pareto Genetic Algorithm (P3GA) [10], to perform parametric expected improvement maximization on cheap surrogate models. The first parametric version of the hypervolume indicator (HVI) metric is also presented as a means for comparing solutions and evaluating optimizer progress in parametric optimization. The new algorithm, Efficient Parametric Optimization (EPO), is benchmarked on two test problems with known solutions prescribing a fixed number of high-fidelity function evaluations and demonstrated on the parametric optimization of a structural stiffened plate requiring FEA. Figure 2.3 relates EPO to other efficient metamodel-based optimization algorithms introduced in Section 2.2 by the type of problems they can solve.

4.1 Efficient Parametric Optimization (EPO)

The novel method explored herein is an extension of EGO to be applicable to parametric optimization problems. Although there are newer and more specialized techniques for adaptive

*This chapter is reprinted with permission from "Efficient Parametric Optimization for Expensive Single Objective Problems" by Weaver-Rosen, J.M. and Malak, R.J., 2021. *Journal of Mechanical Design*, 143(3), 031711 (9 pages), Copyright 2021 by ASME [11].

sampling in an optimization context, such techniques rely on the same underlying principles of exploitation and exploration accomplished by the expected improvement function of EGO [75]. Therefore, this work showcases the key novelty of including parameters in efficient optimization by extending the seminal work, EGO which is widely known in the community. In the same manner as EGO, EPO iteratively updates and uses response surface models to efficiently search the design space to find the parametrically nondominated frontier. The basic procedure is to initialize the response surface model via DOE, parametrically maximize the expected improvement function to sample the expensive model, update the response surface, and repeat until stopping criteria are met. While this is very similar to the procedure of EGO, these steps are performed differently to take parameters into account.

First, the presence of parameters introduces the potential for increasing numbers of necessary response surface models. In EGO, the objective function is considered to be expensive. In EPO, any combination of objective and parameter functions could be expensive. One response surface is trained for each expensive function predicting the objective or parameter value given a set of optimization variables.

Next, the expected improvement function (Eqn. (2.6)) must be redefined such that it is dependent on the parameters. The response surface model predictions $\hat{J}$ and s become functions of both the optimization variables and parameters. Furthermore, J_{min} *must* become a function of the parameters since each location in parameter space ought to have a different (but not entirely independent) baseline. Therefore, the parametric expected improvement function is given by:

$$\begin{aligned} EI(\boldsymbol{x},\boldsymbol{\theta}) = (J_{min}(\boldsymbol{\theta}) - \hat{J}(\boldsymbol{x},\boldsymbol{\theta}))\Phi\left(\frac{J_{min}(\boldsymbol{\theta}) - \hat{J}(\boldsymbol{x},\boldsymbol{\theta})}{s(\boldsymbol{x},\boldsymbol{\theta})}\right) \\ +s(\boldsymbol{x},\boldsymbol{\theta})\phi\left(\frac{J_{min}(\boldsymbol{\theta}) - \hat{J}(\boldsymbol{x},\boldsymbol{\theta})}{s(\boldsymbol{x},\boldsymbol{\theta})}\right). \end{aligned} \tag{4.1}$$

In the case when parameter functions are also expensive, $\boldsymbol{\theta}$ is replaced by its response surface model prediction $\hat{\boldsymbol{\theta}}$ as a function of the optimization variables.

Unlike the scalar J_{min} in EGO, the function $J_{min}(\boldsymbol{\theta})$ is not trivial to compute. For the expected improvement formulation to be valid, a J_{min} value is necessary for all parameter vectors. Therefore, EPO utilizes P3GA to find $J_{min}(\boldsymbol{\theta})$ by parametrically optimizing the cheap response surface model prediction only. Then, EPO uses this $J_{min}(\boldsymbol{\theta})$ to parametrically optimize for maximum expected improvement, Eqn. (4.1), via P3GA to determine new designs with which to sample the expensive, high-fidelity model. In this way, EPO requires *two* runs of parametric optimization (specifically P3GA in this work) on the response surface models in every loop without any expensive function calls.

While doubling the number of analytic optimizations performed per loop (compared to EGO) is not ideal, this disadvantage is balanced and possibly outweighed by the advantage of finding multiple designs with high expected improvement across the parameter space per loop. Since P3GA results in a parametrically nondominated set of designs in terms of maximum expected improvement, multiple different designs can be selected to sample the expensive model and retrain the response surface model before repeating. This is beneficial for parametric problems because the engineer is interested in the entire parametric solution, while the non-parametric EGO only aims to find the single best design. The benefit of choosing multiple designs per adaptive sampling step is better explained by a comparison to the procedure of ParEGO [41] where the solution is a Pareto frontier. In ParEGO, a specific trade-off vector (between multiple objectives) is chosen each time the response surface is optimized for expected improvement. In the ParEGO framework, choosing a new trade-off vector would require an additional optimization. In contrast, multiple parameter locations can be chosen via the parametric optimization of EPO *without* requiring additional optimizations.

The number of designs chosen to sample the expensive model is essentially a meta-parameter of the algorithm. It is not necessary to choose only one (as in EGO), but it should also not be excessive such that the algorithm rarely updates the response surface. Choosing this part of the algorithm is a designer choice largely dependent on the problem being solved and the computational capability of the designer. If a designer is able to evaluate designs with the expensive model in parallel, a larger number of designs might be chosen each iteration to take advantage of the parallel capability. Alternatively, if sampling the expensive model is incredibly expensive or parallel analyses are not possible, a lower number of designs may be chosen each iteration to use the best possible surrogate model before sampling expensive models. Results in this chapter indicate that the algorithm is somewhat robust to the choice of this algorithmic parameter.

The procedure of EPO is thus: *i*) perform DOE to train the initial response surface, *ii*) parametrically optimize the response surface prediction to find $J_{min}(\boldsymbol{\theta})$, *iii*) parametrically maximize the expected improvement (via Eqn. (4.1)), *iv*) select n points to evaluate with the high-fidelity model, *v*) update the response surface model, and *vi*) repeat steps *ii*-*v* until the maximum number of high-fidelity samples have been taken or the maximum expected improvement is below some threshold. This procedure is displayed in Algorithm 4 for the case of dependent parameters.

Algorithm 4 EPO (dependent parameters).

```
$F(\boldsymbol{x}) = [J(\boldsymbol{x}), \boldsymbol{\theta}(\boldsymbol{x})] \leftarrow$ Expensive model
$\hat{F}(\boldsymbol{x}) = [\hat{J}(\boldsymbol{x}), \hat{\boldsymbol{\theta}}(\boldsymbol{x})] \leftarrow$ Initial response surface
$N_{eval} \leftarrow$ Number of expensive evaluations performed
input: $N_{eval}^{max} \leftarrow$ Maximum allowable expensive evaluations
while $N_{eval} < N_{eval}^{max}$ do
    $J_{min}(\hat{\boldsymbol{\theta}}) \leftarrow \min_{\boldsymbol{x}} \hat{J}(\boldsymbol{x}) \; \forall \; \hat{\boldsymbol{\theta}}$
    $\boldsymbol{x}^*(\hat{\boldsymbol{\theta}}) \leftarrow \text{argmax}_{\boldsymbol{x}} \; EI(\boldsymbol{x}, \hat{\boldsymbol{\theta}})$
    if all($EI(\boldsymbol{x}^*(\hat{\boldsymbol{\theta}}), \hat{\boldsymbol{\theta}}) < tol$) then
        break while
    else
        Define $n$ $\boldsymbol{x}_{new}$ tuples such that: $\boldsymbol{x}_{new} \subseteq \boldsymbol{x}^*(\hat{\boldsymbol{\theta}})$
        $J_{new}, \boldsymbol{\theta}_{new} \leftarrow F(\boldsymbol{x}_{new})$
        $\hat{F}(\boldsymbol{x}) \leftarrow$ Update response surface via $\boldsymbol{x}_{new}$, $\boldsymbol{\theta}_{new}$, and $J_{new}$
        $N_{eval} \leftarrow N_{eval} + n$
    end if
end while
return all high-fidelity samples
```

4.2 Benchmarking

Before analyzing an engineering case study, analytic test functions with known solutions are evaluated to determine the viability of the new algorithm. The test functions used to benchmark EPO are parametric variants of the DTLZ test suite of problems [76]. The two scalable test functions chosen from Galvan et al. [5] have been shown to be non-trivial to solve and are used to assess the performance of other parametric optimization algorithms. Furthermore, these problems are chosen because the mathematic formulation scales to any combination of objectives and parameters. For both test problems, a parametric hypervolume indicator (pHVI) is used to compare the results to the true solution by the following:

$$\delta = \mathcal{H}_p^{true} - \mathcal{H}_p^{calc}\,, \tag{4.2}$$

where $\mathcal{H}_p^{true}$ is the parametric hypervolume indicator of the true solution, $\mathcal{H}_p^{calc}$ is the parametric hypervolume indicator calculated from the data, and δ is simply the difference between the true and calculated pHVI values. The first method of calculating the HVI for parametric problems is discussed briefly in the next subsection.

4.2.1 Parametric Hypervolume Indicator

As introduced in Section 2.2.2, the multi-objective hypervolume indicator (HVI) is a popular measure of evolutionary algorithm performance in multi-objective optimization. The hypervolume indicator (HVI) essentially gives an indication of how much space is "covered" by (or dominated by) the data [52]. In the context of multi-objective optimization, this indicator is used to compare solution sets by determining which set dominates more of the reference space or to check for convergence to a single HVI value. In general, solutions with greater HVI values are closer to the true solution and can dominate solutions with lesser HVI values. The HVI can be directly calculated given a data set by summing the mutually exclusive sizes of the hypercube dominated by each point with respect to some reference point. See Refs. [52, 77, 78] for more information on calculating multi-objective hypervolume indicators. The benefit of calculating the HVI metric in evolutionary optimization is two-fold. First, an algorithm or engineer can monitor this value to detect convergence (or the lack thereof). This allows an optimization routine to be shortened or extended as needed. The second benefit is realized when comparing various solutions. Knowing that the true solution maximizes the HVI metric, an engineer should prefer solutions yielding a greater HVI value.

While this indicator is widely used in multi-objective optimization, the original publication of this work is the first to use the concept in parametric optimization [11]. It is important to note that the hypervolume indicator is conceptually the same for both parametric and non-parametric data. In both cases, the hypervolume indicator measures the size of the dominated space. Despite this, the HVI cannot be calculated for parametric data in the same way that it is calculated for multi-objective data. This is because the space dominated by parametric data is not as simple as overlapping hypercubes. Instead, one must rely on the space that is believed to be dominated based on the data. P3GA uses a machine learning technique (specifically a support vector domain description, or SVDD) to predict parameterized Pareto dominance (PPD) [10]. By sufficiently sampling the reference space (e.g., a Monte Carlo technique as in Bader et al. [79]) and predicting whether the points are dominated or not via PPD, a parametric HVI (pHVI) value can be estimated as the fraction of the reference space dominated by the data. This pHVI estimation is bounded from 0 to 1 where a value of 0 means that none of the space is dominated and a value of 1 means that the entire space is dominated. In this section, a Latin hypercube sampling of 4,000 points (chosen through a convergence study) is used to sample the attribute space (objective and parameter space) to predict the pHVI of parametric data. Note that these 4,000 points are *not* evaluated by the expensive or cheap functions. These points are generated directly in the attribute space for the sole purpose of estimating the pHVI value.

Discussion and analysis of this SVDD-based pHVI calculation method compared to another method is presented in Chapter 5.

4.2.2 Test Problem #1

The first example analyzed is test problem C from Galvan et al. [5] with one objective and five optimization variables. This scalable problem is evaluated for one and two parameters. This test function is known to have a discontinuous solution, a non-convex solution, and a biased solution density. The specific problem formulation is Eqn. (A.1) with 5 optimization variables ($N_{var} = 5$).

4.2.2.1 One Parameter

To demonstrate the new framework, EPO is performed 20 times (randomly restarting). For the single parameter variant, an initial DOE of 50 designs is used and the response surface model is updated with two or five designs per sampling step (i.e., $n = 2$ or $n = 5$; cf. Algorithm 4). The maximum number of high-fidelity function evaluations is set to 150 (i.e., $N_{eval}^{max} = 150$; cf. Algorithm 4). Then, P3GA is performed 20 times (also randomly restarting) using the high-fidelity function directly prescribing 10 generations of 15 population members (to match 150 high-fidelity samples per run). To visualize the results, all parametrically nondominated points from all runs are displayed graphically in Fig. 4.1 along with the known true solution. To quantitatively compare the results, the hypervolume indicator is calculated for each run from all methods and its difference from the true solution is displayed in Tab. 4.1 where the pHVI of the true solution is calculated to be 0.8803 (i.e., $\mathcal{H}_p^{true} = 0.8803$ cf. Eqn. (4.2)).

Table 4.1: Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #1 with one parameter (Eqn. (A.1)). Theoretically, negative δ values are not possible, but there is a small error inherent in the calculated pHVI. Each method evaluates the high-fidelity model only 150 times.

Method	Mean δ	Min δ	Max δ
P3GA	0.182	0.117	0.282
EPO (n=2)	0.010	0.000	0.042
EPO (n=5)	0.022	-0.001	0.145

Theoretically, a value of δ equal to zero would mean that the data perfectly captures the true

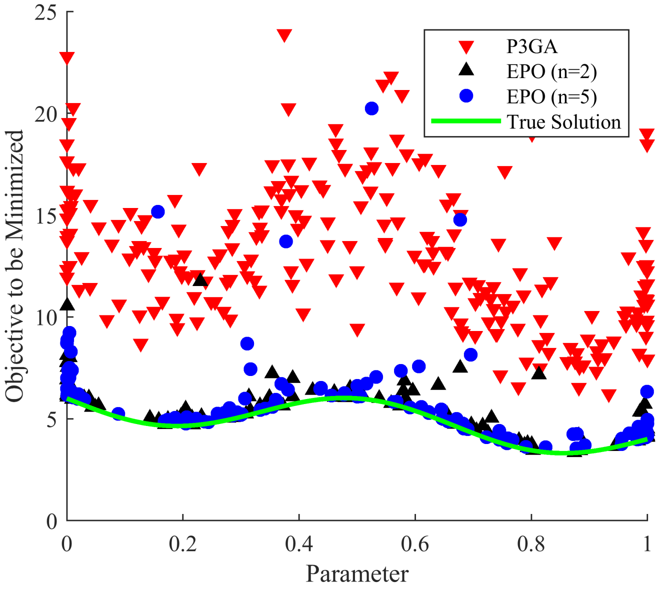


Figure 4.1: Parametrically nondominated solutions found from all 20 runs for each method against the true solution for test problem # 1 with one parameter.

solution, and negative values of δ are not possible. However, the pHVI calculated here relies on a prediction of the dominated space and is thus subject to some inherent error. Note that this error is relatively small and solution sets with higher pHVI values (or smaller δ) are closer to the true solution. Performing a statistical t-test on the final pHVI values reveals that the results of EPO and P3GA are statistically different (99% significance level), but the results of the two EPO runs are not statistically different.

The pHVI calculated throughout each run is plotted against the number of high-fidelity function evaluations in Fig. 4.2a. The point denotes the average value, the whiskers denote the maximum and minimum values, and the boxes denote the interquartile range (middle 50%) for all 20 runs. EPO approaches the true solution in fewer function evaluations compared to P3GA alone. EPO appears to perform better when fewer high-fidelity samples are taken each iteration (i.e., small n value), but this increased performance is coupled with increased wall-clock time. However, one should note that when the high-fidelity function becomes increasingly time-consuming, the increased time of the EPO algorithm becomes negligible.

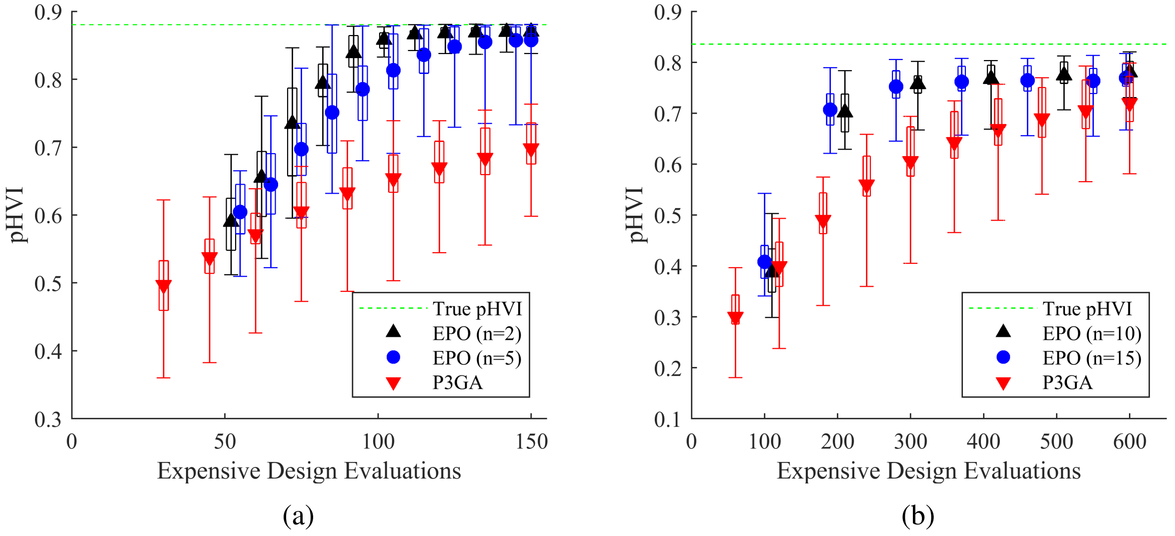


Figure 4.2: Average pHVI versus the number of high-fidelity function evaluations for test problem #1 with (a) one parameter and (b) two parameters.

4.2.2.2 Two Parameters

The same procedure is repeated scaling test problem #1 up to two parameters. The only differences are that: the initial response surface is trained by a DOE of 100 designs, the number of high-fidelity samples evaluated per sampling step is increased to 10 or 15, and the maximum number of high-fidelity samples is increased to 600 (i.e., $n = 10$ or $n = 15$ and $N_{eval}^{max} = 600$). To account for the increase in allowable function evaluations, the benchmarking P3GA directly evaluates 20 generations of 30 population members (for a total of 600 high-fidelity evaluations). The hypervolume indicator is calculated for each run from all methods and its difference from the true solution is displayed in Tab. 4.2 where the pHVI of the true solution is calculated to be 0.8356. Furthermore, the hypervolume indicator is plotted against the number of high-fidelity function evaluations in Fig. 4.2b. For both the one and two parameter variants of this benchmarking problem, the new EPO approaches the true solution in fewer function evaluations than the existing P3GA.

Table 4.2: Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #1 with two parameters (Eqn. (A.1)). Each method evaluates the high-fidelity model 600 times.

Method	Mean δ	Min δ	Max δ
P3GA	0.116	0.037	0.254
EPO (n=10)	0.056	0.015	0.105
EPO (n=15)	0.067	0.018	0.168

4.2.3 Test Problem #2

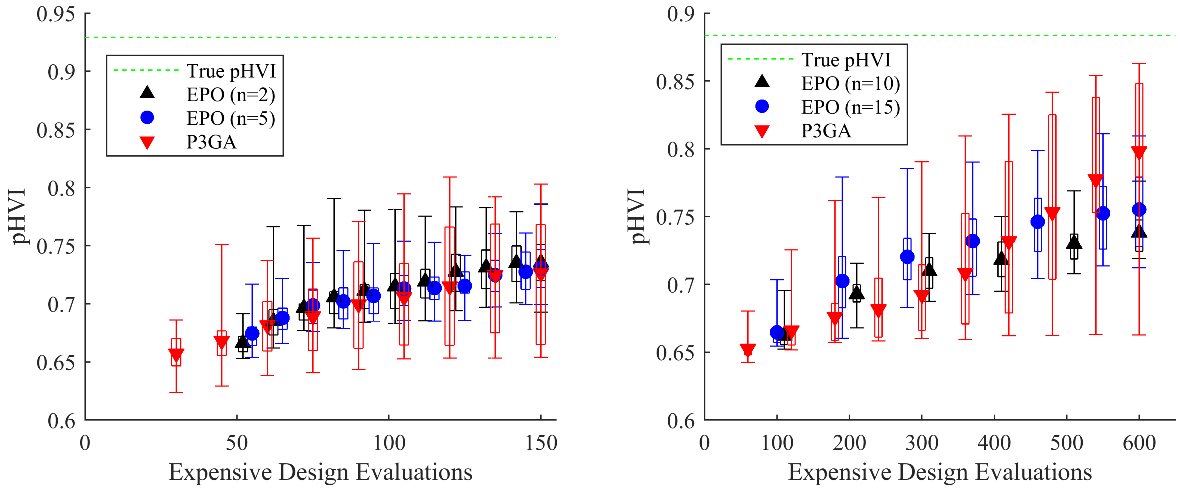
The second example analyzed is test problem B from Galvan et al. [5] with one objective and five optimization variables. This scalable problem is also evaluated for one and two parameters. This multimodal test function is known to have a biased solution density and a variable solution density. The specific problem formulation is Eqn. (A.2) with 5 optimization variables ($N_{var} = 5$).

In the same way as the previous test problem, 20 random restarts are performed for each method. The results of the single parameter formulation of test problem #2 can be found in Tab. 4.3 and Fig. 4.3a (where $\mathcal{H}_p^{true} = 0.9292$), and the results for the two-parameter formulation can be found in Tab. 4.4 and Fig. 4.3b (where $\mathcal{H}_p^{true} = 0.8833$).

In both cases of test problem #2, the best result of the existing method (P3GA) outperforms the best result of the new method (EPO). However, the worst result of P3GA is also worse than the worst result of EPO. For this problem, EPO results in less solution variance when compared to the existing P3GA. This benchmarking problem does not show EPO outperforming the existing P3GA, but rather the performances of EPO and P3GA are very similar. Through an inspection of how EPO

Table 4.3: Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #2 with one parameter (Eqn. (A.2)). Each method evaluates the high-fidelity model only 150 times.

Method	Mean δ	Min δ	Max δ
P3GA	0.203	0.126	0.275
EPO ($n = 2$)	0.194	0.143	0.237
EPO ($n = 5$)	0.199	0.144	0.230



(a) (b)

Figure 4.3: Average pHVI versus the number of high-fidelity function evaluations for test problem #2 with (a) one parameter and (b) two parameters.

Table 4.4: Final mean, minimum, and maximum difference (Eqn. (4.2)) from true pHVI for 20 random restarts of each method on test problem #2 with two parameters (Eqn. (A.2)). Each method evaluates the high-fidelity model only 600 times.

Method	Mean δ	Min δ	Max δ
P3GA	0.085	0.021	0.221
EPO ($n = 10$)	0.145	0.107	0.164
EPO ($n = 15$)	0.128	0.074	0.171

chooses new designs to sample for this problem, we hypothesize that dependent parameters defined by highly multi-modal functions are not sampled efficiently with EPO. The algorithm was not able to quickly find the nondominated designs as seen in the previous benchmarking problem because it took many samples to develop a trustworthy response surface. This is not surprising for noisy problems, but it is promising that the novel EPO does not perform worse than P3GA despite this disadvantage.

4.3 Finite Element Case Study

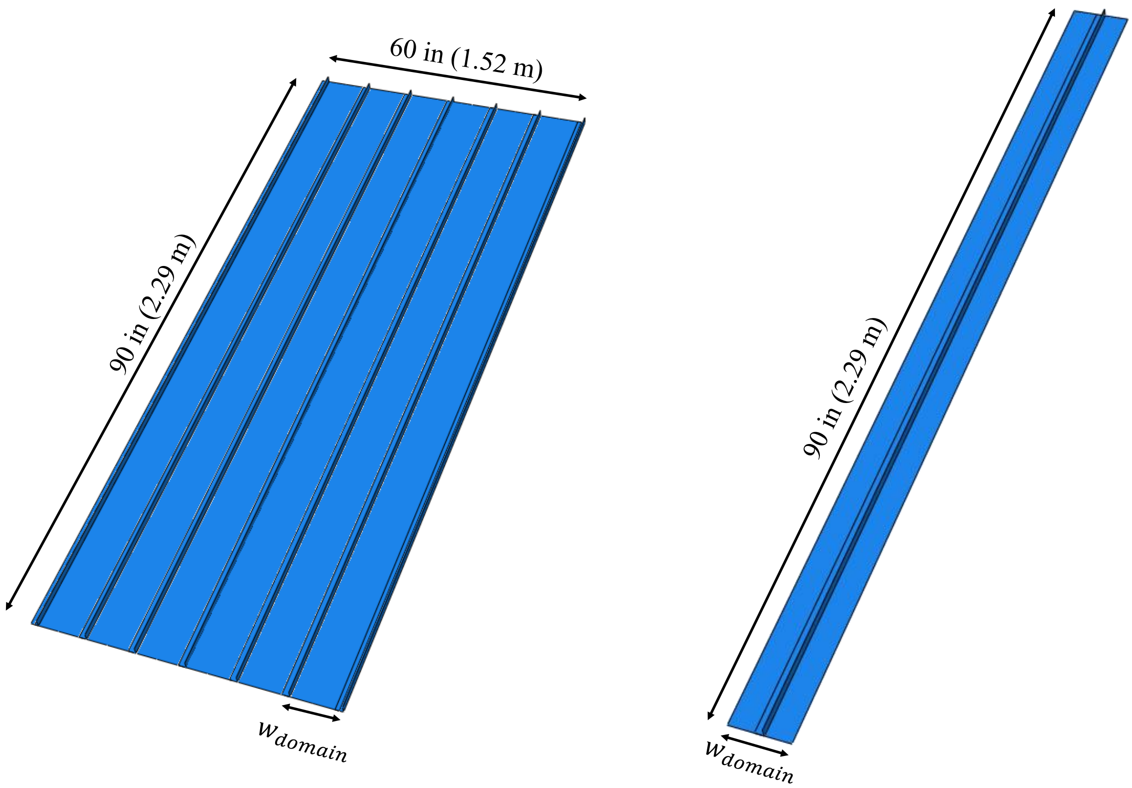
The engineering case study presented in this section is the design of a panel consisting of a thin plate stiffened by L-beams under compression where the primary failure mode is buckling. As such, an eigenvalue buckling analysis is required to calculate the load magnitude that causes the structure to buckle. The design problem is to find the stiffened panel of minimum mass for various load magnitudes prescribing a safety factor of 1.5. In this way, the objective is to minimize mass, and the unknown parameter is the allowable loading magnitude. In a practical sense, an engineer may believe that the allowable loading requirement may change over the course of the project. Another possibility is that the engineer is interested in developing a product platform for which multiple scaled designs are required, each for different loading magnitudes which have not yet been decided. In either case, parametric optimization is well suited to solving this problem.

The stiffened panel is shown in Fig. 4.4a. The outer dimensions of the plate are fixed at 60 in (1.52 m) wide and 90 in (2.29 m) tall. Additionally, the entire assembly is assumed to be aluminum with the following material properties: $\sigma_y = 40$ ksi (275.79 MPa), $\nu = 0.3$, $\rho = 0.0975$ lbs/in^3 (2698.79 kg/m^3), and $E = 10.4E3$ ksi (71.71 GPa). The relevant optimization variables are: the plate thickness, the stiffener thickness, the number of stiffeners per 60 in (1.52 m) panel (n_{stiff}), and the stiffener cross-sectional dimensions. Note that one of these variables (n_{stiff}) is discrete while the other four are continuous.

For simplicity in the FEA model, only one segment of the panel is analyzed as shown in Fig. 4.4b where the width of a single panel segment, w_{domain}, is defined by the full panel width and number of stiffeners (assuming that all stiffeners are equally spaced). This is simply:

$$w_{domain} = \frac{60}{n_{stiff}}\,, \tag{4.3}$$

where 60 represents the total panel width in inches. To improve the accuracy of this reduced order model, two boundary conditions on the long edges are considered separately, and the worst case scenario is used to determine the allowable load. The first boundary condition is symmetry



(a) (b)

Figure 4.4: The (a) full stiffened panel assembly and (b) a single section of the assembly for Abaqus analysis.

along the two long edges, and the second boundary condition is anti-symmetry along the two long edges. Therefore, two FEA evaluations are required to evaluate each design. For both cases, the loading is evenly distributed along one of the short edges while the other short edge is fully encastered. This case study demonstrates the ability of EPO to learn and predict the underlying physics of a black-box model to identify and exploit trends that could be unknown to an engineer. Since the demonstration herein is focused on solving general black-box models, the FEA model is not simplified further. In practice, however, an engineer should simplify their model as much as possible without inhibiting their trust in the model (e.g., using theoretical relationships [80], CAD feature suppression [81], etc.) as doing so can reduce the expense incurred for every expensive function call.

The parametric problem is then:

$$W^*(\theta) = \min_{\boldsymbol{x}} \; \rho V(\boldsymbol{x})$$

where

$$\begin{aligned} \theta(\boldsymbol{x}) &= \min[\theta_1(\boldsymbol{x}), \theta_2(\boldsymbol{x})]\,, \\ \theta_1(\boldsymbol{x}) &= \frac{\lambda_1(\boldsymbol{x})}{1.5(w_{domain})}\,, \\ \theta_2(\boldsymbol{x}) &= \frac{\lambda_2(\boldsymbol{x})}{1.5(w_{domain})}\,, \end{aligned} \tag{4.4}$$

where $W^*(\theta)$ denotes the minimum weight as a function of the parameter, and $V(\boldsymbol{x})$ denotes the volume of the entire assembly (cf. Fig. 4.4a). Furthermore, the parameter $\theta(\boldsymbol{x})$ denotes the allowable compressive edge load for a safety factor of 1.5, and λ_1 and λ_2 denote the critical buckling eigenvalues for the symmetric and anti-symmetric boundary conditions, respectively, calculated via Abaqus simulation of a single section (as shown in Fig. 4.4b).

For this problem, only one run of each method is performed with the *same* initializing DOE of 50 designs. Then, P3GA performs 20 generations with a population size of 5 (yielding 100 additional function evaluations). Further, EPO is used to find 100 additional designs by two procedures. In one run, two high-fidelity samples are taken per iteration, and the other run takes five (i.e., $n = 2$ and $n = 5$; cf. Algorithm 4). The attributes of all parametrically nondominated designs are displayed in Fig. 4.5, and the pHVI is plotted against the number of high-fidelity function evaluations in Fig. 4.6. Given the same design initialization, both runs of EPO are able to find designs that can withstand larger loads while P3GA is unable to find any design that can withstand an edge load larger than 800 lbs/in (140.1 N/mm) resulting in inferior pHVI values. This highlights the advantage of EPO. Because it is a global optimization technique, one would expect P3GA to find solutions for larger edge loads if given sufficient runtime. In contrast, EPO is able to identify these solutions quickly due to its use of a response surface model and adaptive sampling strategy.

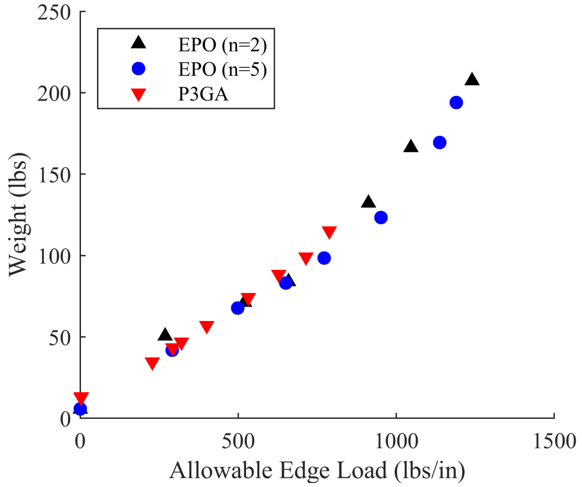


Figure 4.5: Performance of parametrically non-dominated solutions for each method for the finite element case study.

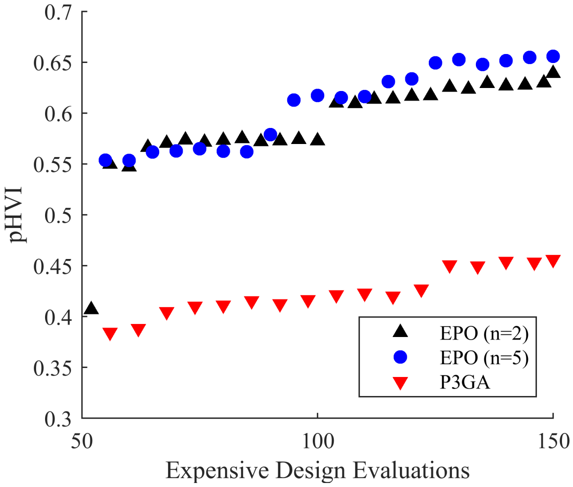


Figure 4.6: pHVI versus the number of expensive design evaluations for the finite element analysis case study.

4.4 Discussion

This work details a new methodology that extends the efficient global optimization (EGO) framework to parametric optimization. The novel EPO shows promise for the problems considered. For the analytical test problem #1 and the engineering case study, EPO consistently outperformed the existing P3GA. The expected improvement formulation allows EPO to locate promising designs with less exploration of the high-fidelity design space compared to the alternative, P3GA. For the analytical test problem #2, the performances of EPO and P3GA were largely the same. For each problem considered, two variations of the EPO algorithm are performed updating the response surface with a different number of high-fidelity samples per iteration, but no statistical difference is found in these results. The problems considered in this chapter are only a small subset of problems applicable for EPO, but it is promising that EPO performed as well as or better than the competition for all problems considered.

4.5 Limitations and Conclusions

Although the EPO methodology represents an important first step towards efficient optimization of parametric problems, there are some limitations. The most obvious limitation is that it is built off of a single-objective method, EGO, and is therefore only valid for single-objective problems. A multi-objective method would, at the very least, need to evaluate the expected improvement differently because Eqn. (4.1) fundamentally assumes that only a single objective is considered. Additionally, the method for determining $J_{min}(\boldsymbol{\theta})$ in the expected improvement formulation, cf. Eqn. (4.1), is ad hoc. The methodology used in EPO appears to work under certain circumstances, but it has the potential to skew results. Since predictions can be wrong and the entirety of Eqn. (4.1) relies on this prediction, EPO is at risk of outputting very poor predictions of points with expected improvement since its baseline is a poor prediction of the true baseline.

There is a single solution to fix both of these major limitations of EPO. In single objective optimization, the improvement function is given by the improvement of the single, scalar objective. However, by drawing on the multi-objective efficient optimization literature, the improvement function can be given by the improvement of any other scalar measure (e.g., the hypervolume indicator) that accounts for all objectives. In the case of parametric optimization, the measure should account for all objectives and all parameters (e.g., the parametric hypervolume indicator). By reformulating the improvement function as the improvement of the pHVI measure, an efficient multi-objective parametric optimization algorithm can be developed that is based on a meaningful quality indicator.

Chapter 5 discusses the SVDD-based pHVI metric from this chapter in more detail and introduces a more efficient calculation method based on a newly defined hypercone heuristic (HCH). Chapter 6 then introduces and benchmarks the multi-objective extension of EPO through the use of this HCH-based pHVI metric for a new expected improvement formulation.

5. PARAMETRIC HYPERVOLUME INDICATOR

As introduced in Sections 2.2.2 and 4.2.1, the hypervolume indicator (HVI) is a scalar indication of quality of any given Pareto set in multi-objective optimization. Since more data closer to the true Pareto froniter yields a greater HVI value, one might prefer a solution with the greatest HVI value. The uses of the HVI highlighted in this dissertation are:

1. To quantify the quality of an optimization solution,
2. To compare solutions from different optimization algorithms or different runs of the same algorithm,
3. To detect convergence of an evolutionary optimization algorithm, and
4. To enable multi-objective efficient optimization through expected improvement of the HVI.

Despite its many uses in multi-objective optimization, the HVI has only recently been introduced to parametric optimization problems as mentioned in Chapter 4 [11]. By introducing a parametric hypervolume indicator (pHVI), the uses of the multi-objective HVI listed above can be used to benefit parametric optimization.

As a result of considering parameters (attributes for which a designer has no preference) in the optimization process, the multi-objective HVI cannot be directly used in a parametric optimization context. However, the general concept of the HVI as a measure of the dominated space has been extended into parametric optimization. Chapter 4 introduces the use of the parametric HVI (pHVI) to compare solution sets from different parametric optimization algorithms [11]. Building upon prior work, this chapter introduces a new and faster technique for calculating pHVI in addition to describing the previous method in more detail. These two techniques are applied to multiple sets of data acquired from an evolutionary parametric optimization algorithm, the predictive parameterized Pareto genetic algorithm (P3GA) [10, 5].

The layout of the chapter is as follows. Section 5.1 gives a more detailed background on the multi-objective HVI, and Section 5.2 explains why the multi-objective HVI is not valid for parametric data. Then, Section 5.3 introduces the new hypercone heuristic (HCH) for parametric Pareto dominance and the corresponding HCH-based pHVI technique. Additionally, more detail is presented on the existing SVDD-based pHVI technique. Section 5.4 compares the two techniques applied to data from parametric optimization with discussion. Section 5.5 provides some intuition on how a designer should choose the reference space and hypercone angle for the HCH. Finally, Section 5.6 closes the chapter with conclusions and how the pHVI metric will be used for the novel MO-EPO algorithm and for parametric optimization in general.

5.1 Multi-Objective Hypervolume Indicator

Used for more than two decades in multi-objective optimization, the hypervolume indicator (HVI) is a scalar measure of the size of space dominated by data [47]. Figure 5.1 illustrates the dominated space for data in two objective space given a reference point. Originally, the HVI was used to assess the performance of multi-objective optimization algorithms by enabling the direct comparison of solution sets [47, 52, 50, 49]. Typically, solution sets with greater HVI value are preferred because if one solution set dominates another, then it must have a greater HVI value. While the opposite cannot always be said (i.e., one cannot say that a solution with greater HVI value *certainly* dominates another solution), Fleischer [51] presents a proof that maximization of the HVI constitutes the necessary and sufficient conditions for finding the true Pareto set. This result further motivates one to prefer solution sets with greater HVI value. This is especially true in evolutionary optimization. In evolutionary optimization where nondominated data are not removed from consideration, one can prove that any increase in HVI value from one iteration to the next is indicative that a more preferred solution has been found [52].

In a detailed study, Knowles and Corne [82] recommend the HVI as a viable method for performance assessment in multi-objective optimization citing its Pareto compatibility and intuitive meaning. According to this study, the main limitations of the HVI are the large computational overhead and need to define a reference point [82]. To address the computational over-

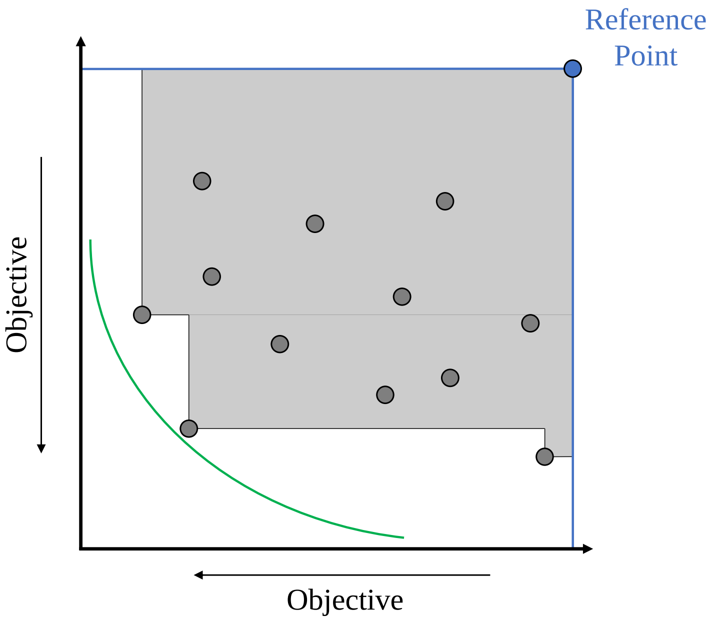


Figure 5.1: Notional representation of dominated space in two objectives. Arrows represent objective preference direction. Green curve represents a notional Pareto frontier.

head, many researchers have proposed innovative methods of calculating the HVI that are less demanding (e.g., the walking fish group algorithm [77] and convex hull assumption [83]). The HVI has also been extended to metamodel-based multi-objective optimization whereby expected improvement functions are formulated to measure the expected improvement of the HVI (sometimes referred to as EHVI) [46, 12, 45, 84, 78, 85]. The HVI has also been used to reformulate multi-objective optimization problems [51], detect convergence, or to develop new optimization algorithms [86, 87, 45, 85].

5.2 Parametric Incompatibility of the Multi-Objective Hypervolume Indicator

Multi-objective HVI calculations rely on the fundamental assumption that each attribute has a preference (i.e., that each attribute is an objective). Since parameters are, by definition, attributes for which a designer has no preference, existing multi-objective calculation methods are flawed in this context and cannot be directly used when parameters are present. Instead, parametric Pareto dominance must be considered to account for parameter attributes without preference. Parametric

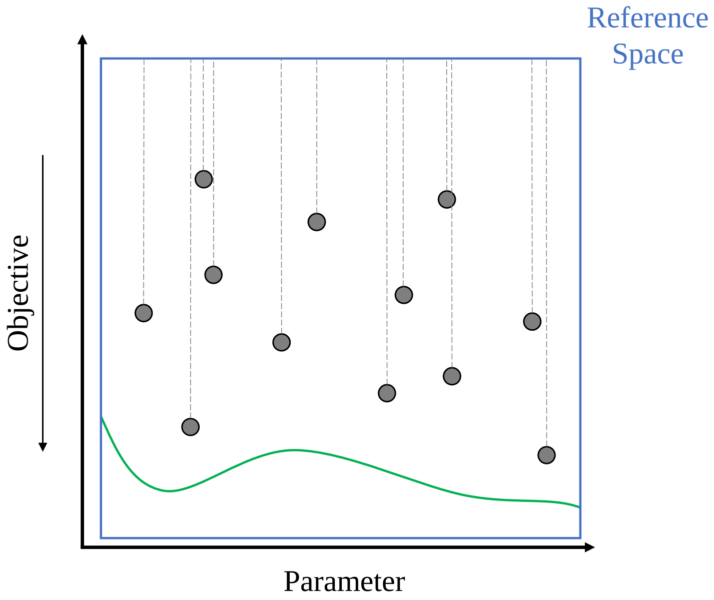


Figure 5.2: Parametric data in two dimensions (one objective and one parameter) demonstrating parametric Pareto dominance. Arrow represents objective preference direction. Green curve represents a notional parameterized Pareto frontier.

Pareto dominance (PPD) states that parametric data can only dominate other data if they share the exact same parameter values [9]. Figure 5.2 shows parametric data in two dimensions (one objective and one parameter). Following the rule of parametric Pareto dominance (PPD), all data are considered nondominated because no data share the *exact* same parameter value.

Consider the mathematical definitions of Pareto dominance and parametric Pareto dominance. Let $\mathbf{P}$ and $\mathbf{Q}$ be vectors of performance values for which an engineer prefers to minimize all entries of each vector (i.e., each entry is an objective to be minimized). These vectors represent points in an objective space. $\mathbf{P}$ is said to Pareto dominate $\mathbf{Q}$ iff:

1. $P_i \leq Q_i \, \forall \, i$, and
2. $\exists \, i \, (P_i < Q_i)$.

In words, $\mathbf{P}$ is said to Pareto dominate $\mathbf{Q}$ if and only if every objective in $\mathbf{P}$ is at least as good as every objective in $\mathbf{Q}$ so long as at least one objective in $\mathbf{P}$ is strictly better than that of $\mathbf{Q}$. If

$\mathbf{P}$ does not Pareto dominate $\mathbf{Q}$ and $\mathbf{Q}$ does not Pareto dominate $\mathbf{P}$, both points are considered to be mutually nondominated or incomparable. Graphically, Fig. 5.1 shows three points that are mutually nondominated whereas the remaining points are Pareto dominated by at least one of these three nondominated points.

Now let $\mathbf{P}$ and $\mathbf{Q}$ be vectors of objectives *and* parameters. Let i denote the objective indices of each vector, and let j denote the parameter indices of each vector. $\mathbf{P}$ is said to parametrically Pareto dominate $\mathbf{Q}$ iff:

1. $P_j = Q_j \,\forall\, j$,

2. $P_i \leq Q_i \,\forall\, i$, and

3. $\exists\, i\, (P_i < Q_i)$.

The second and third criteria above are identical to the conditions for Pareto dominance. However, parametric Pareto dominance also requires that all parameter values of each point be identical. Graphically, Fig. 5.2 illustrates 12 mutually nondominated points via parametric Pareto dominance rules because no two points share the exact same parameter values (cf. the first condition of parametric Pareto dominance above).

Since existing methods of calculating multi-objective HVI do not consider parametric Pareto dominance (PPD) rules above, they are not compatible with parametric data. Consider the true solution to a parametric problem in Fig. 5.3a. If we have infinite data making up the true solution, we can look closely at a portion of the true solution in Fig. 5.3b. Since the data is infinite, we can visualize a grid of points sharing the exact same parameter values. Unlike Fig. 5.2, some data is now dominated and a dominated space can be visualized. However, if the multi-objective HVI is calculated of this data, the parameter would be incorrectly given a preference direction. If the parameter (which has no preference by definition) is assigned a preference, the dominated space could look like the illustration in Fig. 5.4. Treating a parameter as an objective is fundamentally incorrect when determining dominance and when measuring the size of the dominated space via

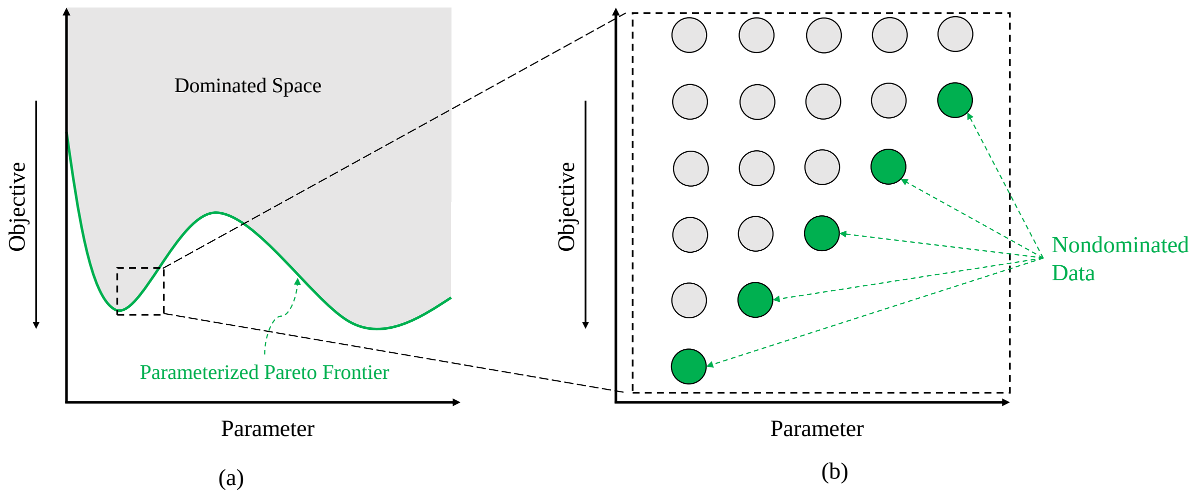


Figure 5.3: (a) True parametric solution in two dimensions (one objective and one parameter) for a notional parametric problem. (b) Application of parametric Pareto dominance to data in a grid. Arrow represents objective preference direction. Parameter has no preference direction.

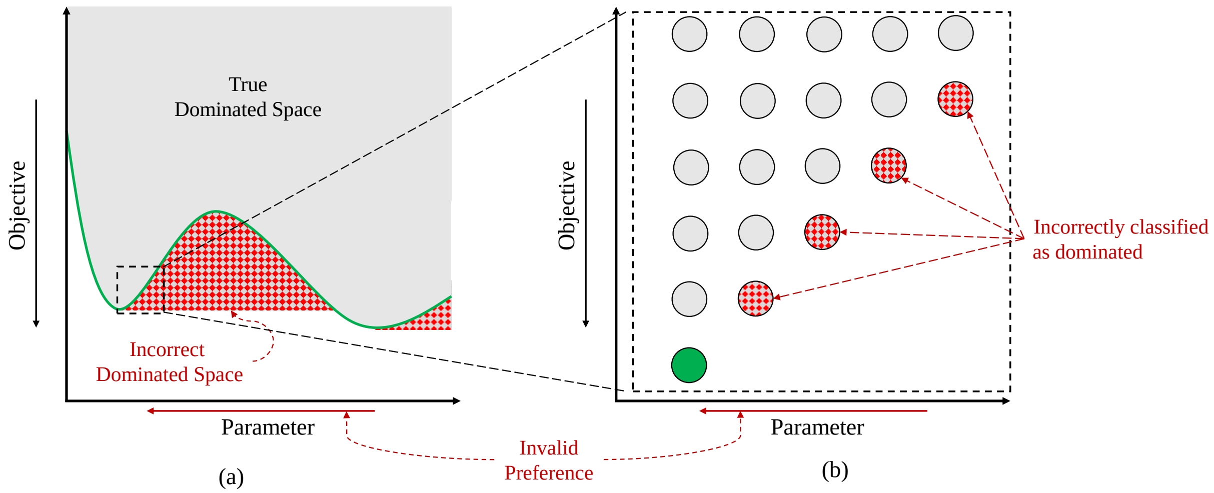


Figure 5.4: The application of Pareto dominance to parametric data is fundamentally incorrect as (a) the dominated space can be overestimated and (b) parametrically nondominated data are incorrectly classified as dominated. Black arrow denotes true objective preference direction while red arrow denotes the invalid preference given to the parameter.

something like the HVI measure. Therefore, another method is needed that will properly consider parameters.

5.3 Parametric Hypervolume Indicator Methods

As motivated in the previous section, a method for calculating the parametric hypervolume indicator is needed as existing multi-objective methods are not directly applicable. Unfortunately, calculating the size of the space dominated by PPD rules is problematic. If one takes the literal meaning of PPD, the dominance "line" in Fig. 5.2 should have no thickness, and thus the size of the dominated space is effectively zero. Relaxing the conditions of PPD slightly to add some thickness to these "lines" will result in a measurable dominated space, but this method is still limited especially when additional objectives and parameters are present. Figure 5.3 is only able to show a measurable dominated space under the assumption that an infinite amount of data is present.

Until recently, there has been no attempt to develop a HVI or similar measure for parametric optimization. This is due to the relatively new introduction of evolutionary algorithms for parametric optimization [10]. As engineers continue to solve increasingly difficult problems, the use of parametric evolutionary algorithms is becoming more common. Recently, such algorithms have been used to optimize a liquid metal cooling concept [7] and a morphing airfoil concept [2]. Other applications of parametric evolutionary algorithms include dynamic control design [8], product family/platform optimization (alluded to in Ref. [11]) and other systems engineering contexts.

Prior to the parametric HVI (pHVI), parametric optimization solutions were primarily compared against the result of iterating traditional optimizers for various combinations of parameter values [7, 2]. Although this provides strong evidence of a correct solution, it is not efficient because repeatedly solving traditional optimization routines can become very costly [5]. The primary benefit of parametric optimization is the overall reduction in computational expense by solving a single all-encompasing parametric optimization problem compared to solving multiple traditional optimization problems [5]. Galvan et al. [5] also demonstrate the only other method of comparing parametric results through the two-sided mean Hausdorff distance. The main disadvantage of this method is that it requires two solutions for comparison. A metric like the pHVI is needed for independent quality assessment for detecting convergence or to inform an algorithmic optimization search.

5.3.1 SVDD-Based Method

The SVDD-based pHVI method presented in Ref. [11] defines the size of the dominated space through the use of a support vector domain description (SVDD) [69]. An SVDD is a one-class classifier that has been used to predict parametric Pareto dominance in the evolutionary predictive parameterized Pareto genetic algorithm (P3GA) [10]. The SVDD-based pHVI method generalizes all data in the attribute space (objectives and parameters) to predict the feasible region [10]. Then, the dominated space is defined as the space within the SVDD boundary or dominated by any point within the SVDD boundary. Figure 5.5a demonstrates a two-dimensional (one objective and one parameter) illustration of this technique. The SVDD boundary is represented by the dotted line and the dominated space is represented by the shaded region. This method generalizes within the data, but may not predict any dominated area for parameter values outside of the existing data. The implication of this is that data at extreme parameter values (near the upper or lower bounds) can be valuable regardless of their objective values.

Training the SVDD requires solving a separate quadratic programming problem to find the model that best represents the data [10]. Once the SVDD model is known, regions of the space are easily determined to be inside or outside of the SVDD boundary with a linear function. Any point within the SVDD is easily classified as dominated, but points outside of the SVDD must undergo further evaluation. In this implementation any point outside of the SVDD is added to the SVDD incrementally at little additional cost. Then a small step taken in the direction of objective preferences is evaluated to be determine whether the current location is on the nondominated or dominated side of the SVDD as in Ref. [10] to determine whether the space is dominated by the SVDD or not. Therefore, this method requires training an initial SVDD as well as training a number of incrementally different SVDDs to determine the pHVI value. To enable a wider use of the pHVI, a faster method is needed.

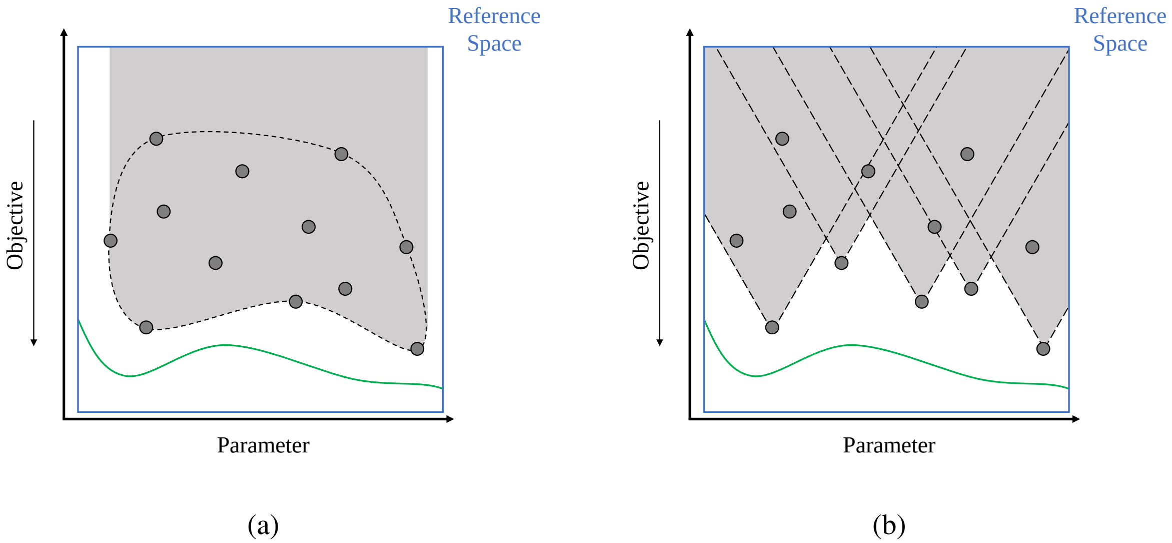


Figure 5.5: Dominated space of two-dimensional data (one objective and one parameter) illustrated by shaded regions for the (a) SVDD-based method, and (b) HCH-based method. Arrows represent objective preference direction. Green curves represent notional parameterized Pareto frontiers.

5.3.2 HCH-Based Method

The new method presented in this chapter is based on a newly defined hypercone heuristic (HCH) drawing inspiration from the multi-objective cone-based HVI [88]. The most obvious difference from the SVDD-based method is that no quadratic programming problems must be solved. Instead, any point in the space can be classified as dominated or nondominated via linear projections through parameter space and traditional Pareto dominance. This should result in a noticeable difference in computational cost between the two methods. The novel hypercone heuristic (HCH) is a relaxed version of parametric Pareto dominance rules. The HCH states that data point $\mathbf{Q}$ could be practically considered as dominated if:

1. There exists another data point $\mathbf{P}$ with similar parameter values,

2. All objective values of point $\mathbf{P}$ are comparable to or preferred to that of point $\mathbf{Q}$, and

3. $\mathbf{P}$ sufficiently outperforms $\mathbf{Q}$ in at least one objective.

Unlike parametric Pareto dominance rules, the hypercone heuristic is not explicitly decisive. Phrases such as “similar”, “comparable”, and “sufficiently outperforms” are not definitive, and the specific meaning of these phrases will change depending on the context. Therefore, this heuristic is mathematically defined by a hypercone across the parameter space and the hypercone angle provides a concrete definition for the unspecific phrases mentioned previously. Any data falling within this hypercone is practically considered dominated, thereby defining a calculable and meaningful dominated space.

Though the HCH can be mathematically scaled to any number of objectives and parameters, Fig. 5.5b illustrates this concept on a two-dimensional set of data. The dashed lines in the figure represent the dominating hypercones of the nondominated data, and the shaded region represents the dominated space. Following intuition, data with very similar parameter values and much worse objective values compared to other data is considered dominated (as opposed to strict PPD rules where all data is nondominated; cf. Fig. 5.2). To fully implement the HCH, a user must define the reference space and the hypercone angle. Note that an angle of zero degrees would constitute the strict PPD rules. As the hypercone angle increases from zero, the PPD rules are more and more gradually relaxed. A hypercone angle of 180 degrees would negate the consideration of parameters as strict Pareto dominance would be employed regardless of parameter values in the data. Both of these extremes are impractical, but a hypercone angle in between can be useful. It is important to note that the HCH will never violate Pareto dominance rules because the dominating hypercone exists across parameter space only to enable the comparison of designs with differing parameter values. Given any two designs with the exact same parameter values, strict Pareto dominance rules are followed.

5.3.3 Current Implementations

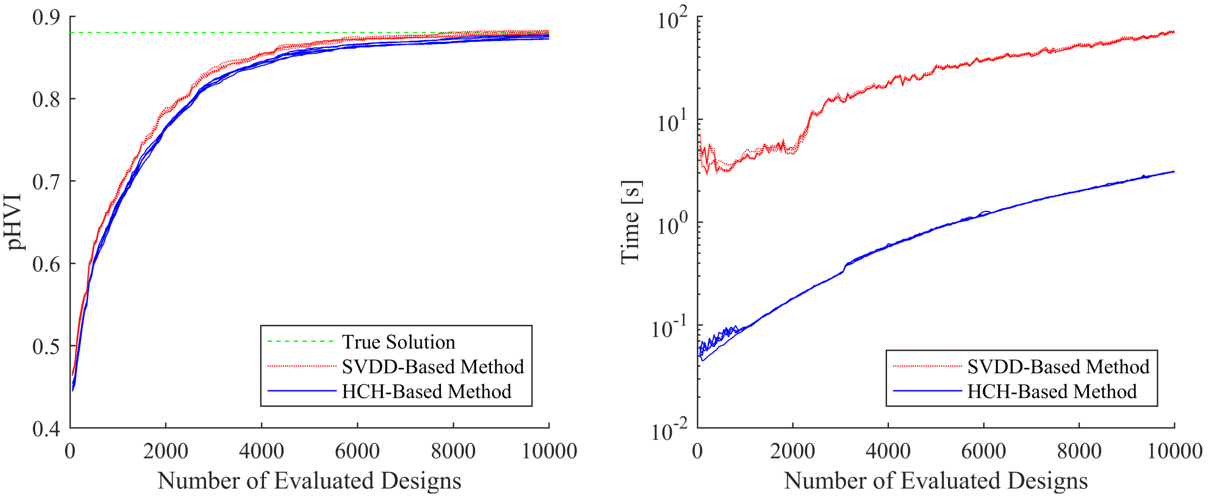
The specific implementations for both methods demonstrated in this chapter use a quasi-Monte Carlo sampling approach within the reference space to estimate the size of the dominated space (based on Ref. [79]). Specifically, 4,000 samples generated via Latin hypercube sampling are used as in Chapter 4 [11].

The SVDD-based method considers no data as outliers and uses an SVDD width parameter equal to 4 (see Ref. [69] for details). The HCH-based method also considers no data as outliers and uses a hypercone angle ϕ_{hc} equal to 90 degrees. As a heuristic method there is no "correct" value for this angle. A 90 degree angle is chosen because it is the midpoint of the two extremes (0 and 180 degrees). Section 5.4.3 provides evidence to support the use of a 90 degree hypercone angle in addition to giving some insight into the effect of this hypercone angle for optimization. Section 5.5 provides more intuition on the effect of both the hypercone angle and reference space on HCH results.

5.4 Results

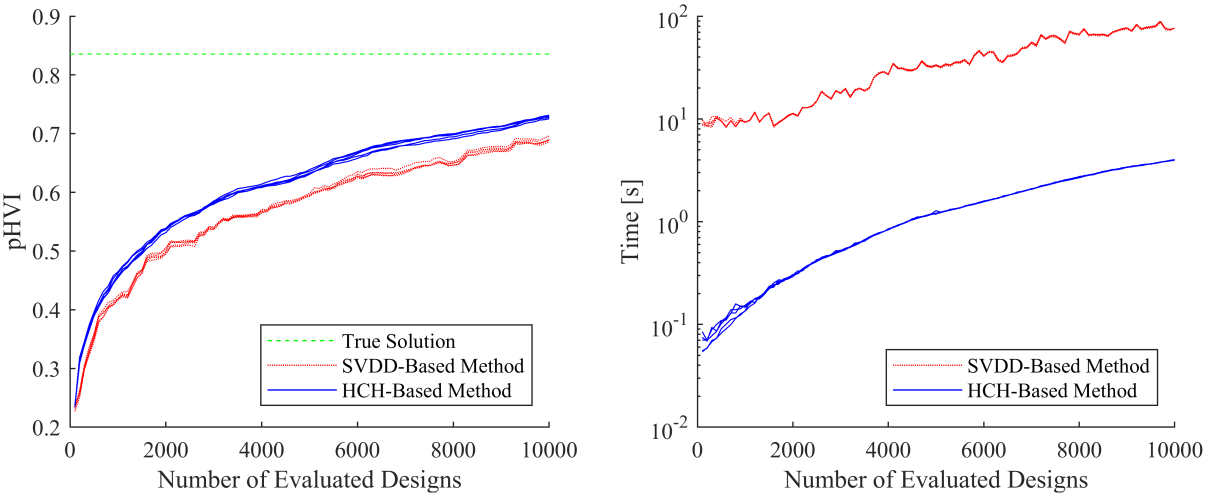
5.4.1 Scalable Test Problems

The data used to calculate the pHVI is gathered from parametric optimization of test problems #1 and #2 (Equations (A.1) and (A.2), respectively, in the Appendix) each with 12 optimization variables ($N_{var} = 12$) from Galvan et al. [5]. Since these are scalable test problems, various numbers of objectives and parameters can be considered within these two problems. For each set of data, the pHVI is calculated for a progressively increasing number designs to emulate what could be seen during an evolutionary optimization procedure. Unlike what would ideally be done during optimization, each pHVI calculation is started from scratch to get an accurate representation of computational time for a given number of designs (ideally, there would be some transfer of knowledge from one iteration to the next to improve computational speed). To promote comparison between the methods, five sets of Latin hypercube samples are prescribed and each method calculates the pHVI value using the same five sets of Latin hypercube samples to predict the size of the dominated space for the *same* data obtained from optimization via P3GA [10]. Note that all calculations are done on the same machine with an Intel(R) Core(TM) i7-9700 CPU at 3.00 GHz on 64-bit OS. The slower SVDD-based calculations utilize eight (8) parallel processes in Matlab while the faster HCH-based calculations are performed in series. Additionally the "true" pHVI value calculated by an extensive optimization run of 300 generations of 300 population members via P3GA is shown as the dashed green line in the following figures.



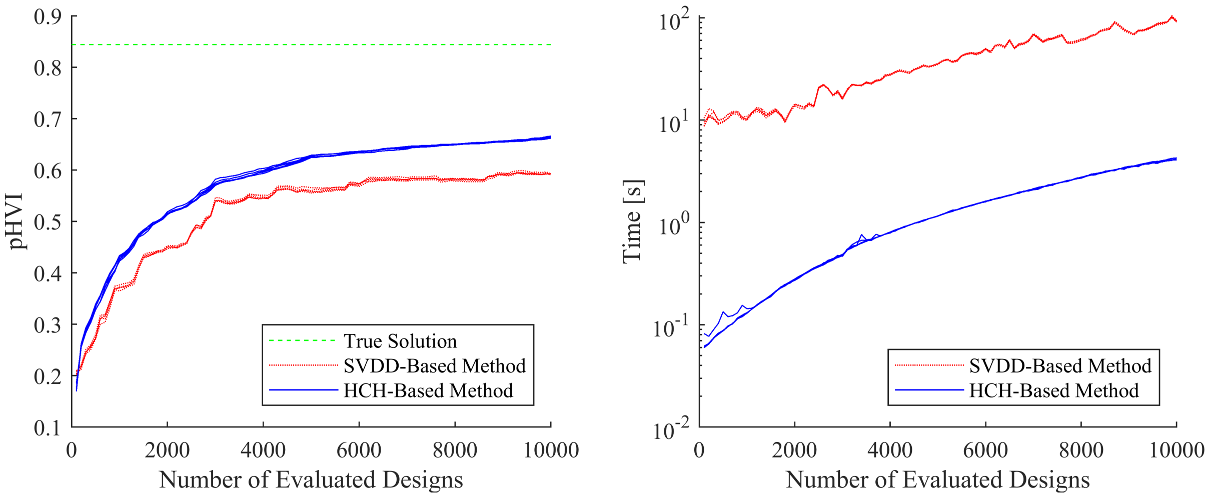
(a) (b)

Figure 5.6: Test problem #1 (a) pHVI and (b) time results for one objective and one parameter.



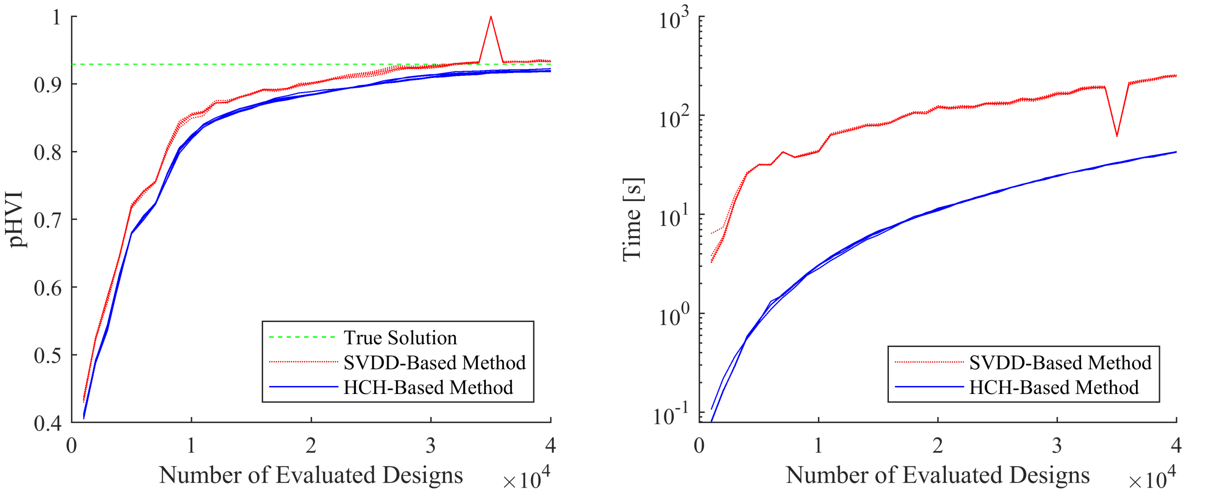
(a) (b)

Figure 5.7: Test problem #1 (a) pHVI and (b) time results for one objective and two parameters.



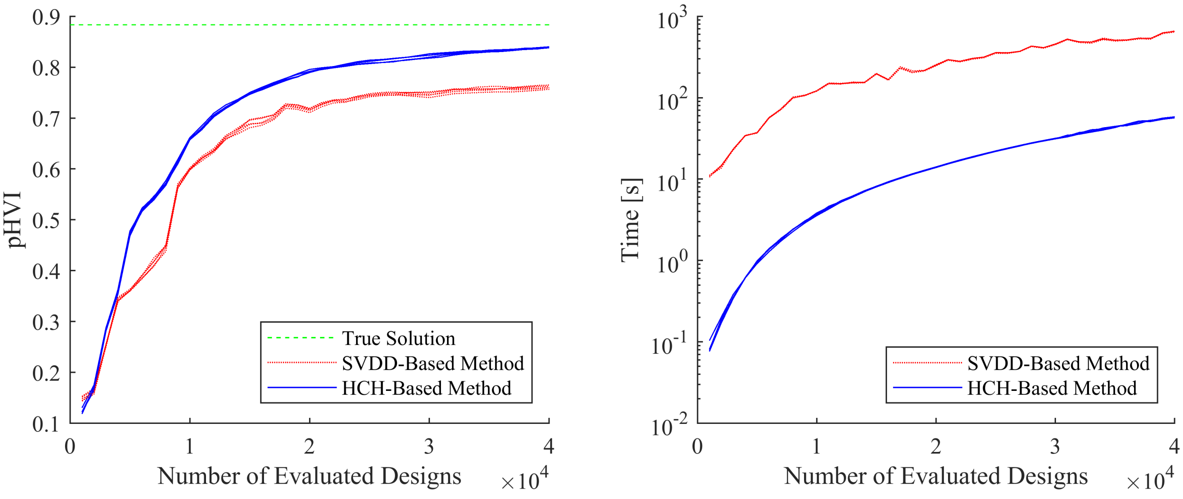
(a) (b)

Figure 5.8: Test problem #1 (a) pHVI and (b) time results for two objectives and one parameter.



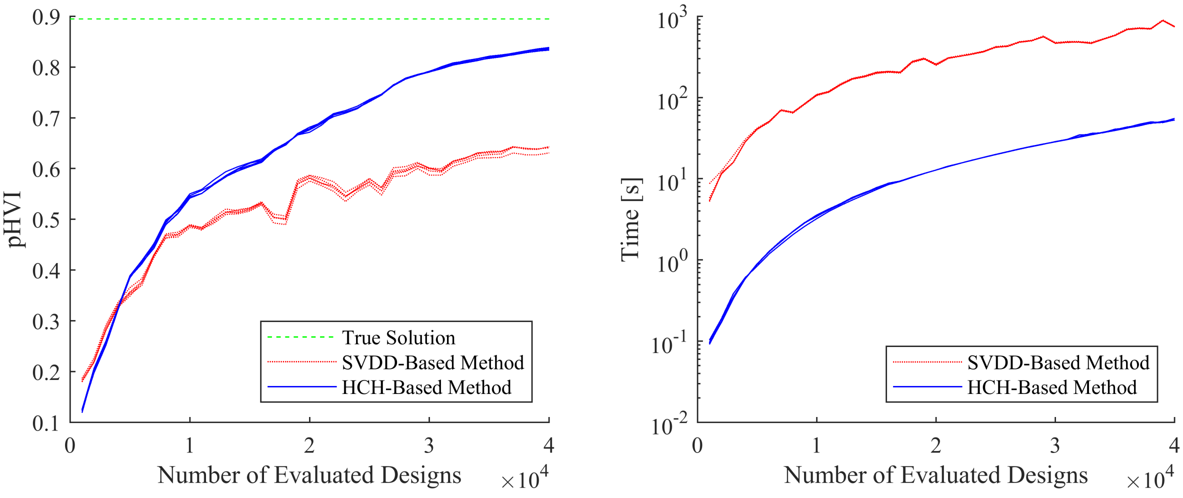
(a) (b)

Figure 5.9: Test problem #2 (a) pHVI and (b) time results for one objective and one parameter.



(a) (b)

Figure 5.10: Test problem #2 (a) pHVI and (b) time results for one objective and two parameters.



(a) (b)

Figure 5.11: Test problem #2 (a) pHVI and (b) time results for two objectives and one parameter.

Figures 5.6-5.11 show the results from each test problem with a different formulation (number of objectives and parameters) considering both the SVDD-based and HCH-based pHVI calculation methods. Each figure plots five curves for each method, one per prescribed set of Latin hypercube samples. As one might expect, the precise pHVI values given by each method vary. This is because the SVDD-based method generalizes data more within the existing data whereas the HCH-based method generalizes data equally within and beyond the existing data (cf. Fig. 5.5). Depending on the context, one assumption might be preferred over the other. For these benchmarking examples, the important thing to note is the similarity in relative trends of pHVI. In two cases, a larger deviation in trend is found. For test problem #2 with one objective and one parameter, the SVDD-based method predicts a strange spike in pHVI up to a value of 1 (cf. Fig. 5.9a). This spike is a result of training an invalid SVDD model due to numerical instability when solving the quadratic programming problem that defines the model. This model is quickly corrected as more data is added and the SVDD is retrained. Such a gross mischaracterization of the data is not common overall, but similar behavior of the SVDD for data from this problem was found in three out of three random repeats. This behavior was not observed for any other problem formulation considered here. In the results for test problem #2 with two objectives and one parameter, the pHVI trends between the SVDD- and HCH-based methods is greatest (cf. Fig. 5.11). In this case, the SVDD defined a more conservative boundary between the dominated and nondominated space likely due to a lack of nondominated data in extreme regions of the parameter space.

Despite their differences in pHVI value, both methods provide some indication that the parametric data is converging or beginning to converge to a final solution. However, through the pHVI value of the data and the true solution, it is clear that the data illustrated by Fig. 5.8 is converging to a sub-optimal solution. This is a reflection of the underlying data gathered from the optimization routine, not a failure of the pHVI calculation.

As hypothesized, the biggest difference between these two methods is the computational effort. In all cases, the SVDD-based method scales poorly with increasing number of evaluated designs. Though this time may be tractable for a small number of pHVI calculations, the SVDD-based method is less suited to direct implementation within an evolutionary algorithm (e.g., as convergence criteria, as a single objective, or otherwise as information for the optimizer).

5.4.2 Discussion of the Two pHVI Methods

Understanding how the dominated space changes in each method is important to properly compare these two techniques. In the SVDD-based method, the boundary of the dominated space is almost constantly changing as new data is used to better predict the SVDD boundary. Initially this sounds ideal, but some problems can arise with these predictions. As discussed in multi-objective HVI theory, the HVI value should be increased when adding nondominated data but should remain unchanged when adding dominated data [52]. When updating the SVDD model, it is possible to violate both of these properties. In most cases, adding nondominated data has the desired effect: expand the SVDD boundary to predict a larger dominated space. However, adding dominated data can actually shrink parts of the SVDD boundary to predict a smaller dominated space overall. If newly added, dominated data reveals that the previous SVDD boundary was over-predicting the dominated space, a new SVDD boundary could actually result in a lower pHVI value. This phenomenon is clearly seen in Figs. 5.11a and 5.8a where the pHVI value drops noticeably in multiple locations. Therefore, the SVDD-based method can violate the multi-objective HVI properties because adding information should not reduce the size of the dominated space [52].

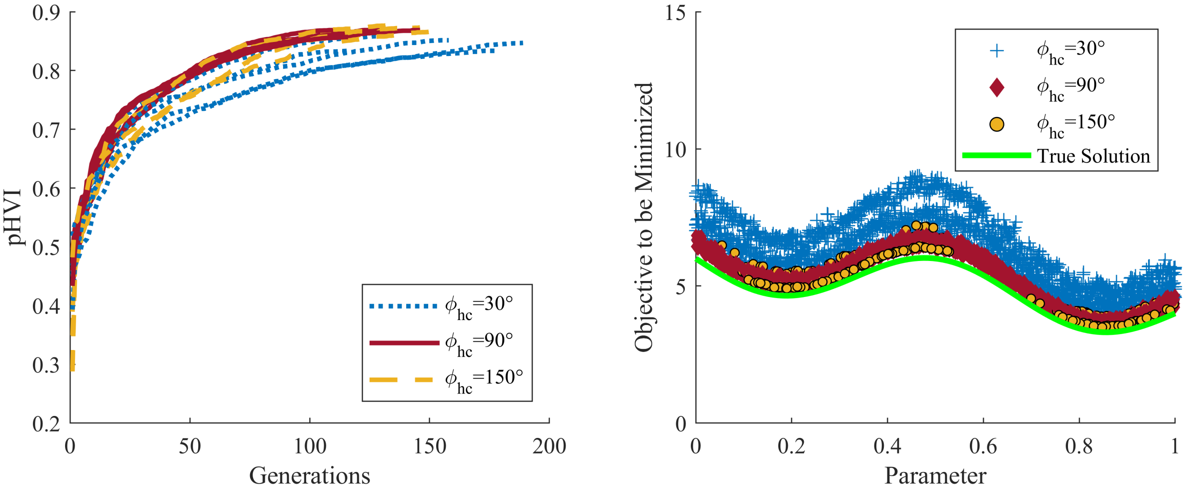
In contrast, the definition of the dominating hypercone (i.e., the hypercone angle and vertex) remains constant in the HCH-based method. Therefore, new data that is dominated by existing data has no effect on the pHVI while new data that is nondominated by existing data will increase the pHVI (so long as the Latin hypercube samples remain unchanged). This adheres to the multi-objective HVI properties [52]. These properties are important since any violation can disrupt certain applications of the pHVI. For example, if an optimizer were being used to maximize the pHVI value or to find designs that maximize the expected improvement of the pHVI value, such algorithms could halt prematurely or actively reject data that does not agree with the current SVDD model even if that model is incorrect.

Another benefit that the HCH-based pHVI method shares with the multi-objective HVI is the ability to reduce computational time when implemented in an iterative context (e.g., evolutionary optimization). Since the hypercone definition remains unchanged, if a point in the reference space is determined to be dominated at iteration i, it must also be dominated at iteration $i + 1$. Therefore, designs and quasi-Monte Carlo samples that have been classified as dominated do not need to be considered again in the future because they will certainly remain dominated (a statement that cannot be said of the SVDD-based method). This allows for the reduction in computational cost in two ways: 1) the effective number of evaluated designs considered is reduced and 2) the effective number of quasi-Monte Carlo samples needed to accurately predict the size of the dominated space is reduced. This means that the computational times of the HCH-based method presented in Figs. 5.6-5.11 represent the worst-case calculation times.

5.4.3 Parametric Optimization with the HCH

In addition to quantifying the quality of parametric data, one of the desired applications of the HCH-based pHVI method is for use during optimization for predicting parametric Pareto dominance (PPD), detecting convergence, etc. Therefore, a modified version of P3GA [10] employing the hypercone heuristc is used to solve a parametric optimization problem. In contrast to results presented in the previous subsection, these results will *not* use the same underlying data. Instead, three separate optimization routines are run that each consider a different hypercone angle ($\phi_{hc} = 30°$, $90°$, and $150°$) to better understand the algorithmic hyperparameter effect on optimization solutions[1]. Each optimization routine assumes a population size of 50 and is run until convergence of the pHVI value is achieved. In this work, convergence is defined by a less than 0.5% change in pHVI value over 10 consecutive generations. To account for the stochastic nature of the optimization algorithm, five repeats are performed for each hypercone angle. Figure 5.12a shows the pHVI value as a function of generations until convergence is met for each hypercone angle applied to a simple one objective, one parameter problem (Eqn. (A.1)) for visualization purposes. Note that the exact value of the pHVI cannot be directly compared for different cone angles (since a larger cone

[1] All runs consider the same reference space bounded by 0 and 1 in the parameter dimension and bounded by 0 and 40 in the objective dimension.



(a)

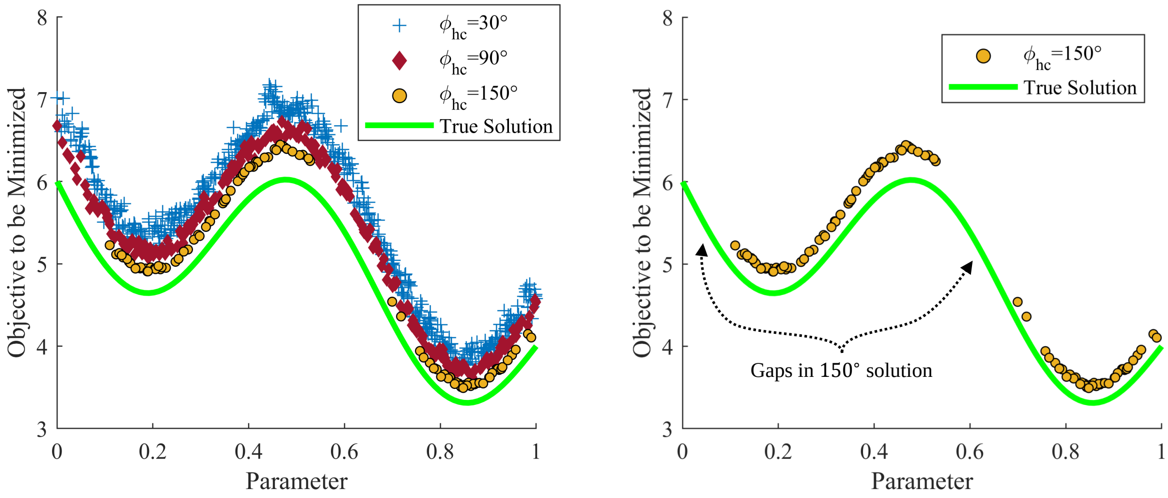
(b)

Figure 5.12: (a) Convergence of pHVI and (b) nondominated designs for three implementations of the hypercone heuristic as dominance criterion in modified P3GA applied to test problem #1 with one objective and one parameter. Note that the true solution is given by the green line in (b).

angle will produce a larger pHVI value for the same underlying data). Instead, an engineer can gain a qualitative sense of how the solution is improving and how many generations are needed to converge.

The purpose of this study is to gain a qualitative sense of how the cone angle affects the optimization solutions. Since only five repeats are performed for each hypercone angle, there is no statistical difference in the number of generations required for convergence. Qualitatively, however, the solutions are quite different. Figure 5.12b shows the nondominated data produced by each method for a qualitative comparison against the true solution. The main takeaway from this figure is that all three cone angles produce solutions near the true frontier (represented by the solid green curve), but solutions for the optimizer using a 30° cone angle are consistently farthest from the true frontier.

To aid in visualization, Fig. 5.13a shows a close-up plot of only one solution set (that which has the greatest resulting pHVI value) per cone angle from Fig. 5.12b. Through this image, the differences in optimization results using different hypercone angles are more apparent. First, the



(a)

(b)

Figure 5.13: Nondominated set of solutions for (a) a single run of each hypercone angle and (b) a single run of $\phi_{hc} = 150°$ only. Result using the 30° cone angle is farthest from the true solution, and the result using the 150° cone angle misses parts of the parameter space.

trend is seen that larger hypercone angles result in solutions marginally closer to the true solution. However, optimizing with a hypercone angle of 150° in this case misses parts of the solution (see the gaps pointed out in Fig. 5.13b). Although Fig. 5.13a shows only a single solution for each cone angle, these trends are consistent across all five repeats. These results agree with intuition of the hypercone heuristic. For a hypercone angle that is too small, more data is considered to be nondominated (recall the extreme case where $\phi_{hc} = 0°$ and most, if not all, data is classified as nondominated). This can detract from the performance of an evolutionary algorithm such as P3GA since more designs that might be practically considered as dominated will be classified as nondominated instead. For a hypercone angle that is too large, more data is considered to be dominated including data that might truly be nondominated. In this example, the genetic algorithm searches more heavily near designs with more desireable objective values regardless of parameter values leaving certain areas of the parameter space undersampled. It is interesting to note that the optimizer is somewhat robust to the HCH angle ϕ_{hc}, but there is a noticeable effect. In each case, the nondominated designs produced are close to the true frontier, but one might qualitatively say

Table 5.1: Example of Reference Space Description.

	θ_1	J_1
Lower	0	0
Upper	1	40

that the result at $\phi_{hc} = 90°$ is better than the others. These results support the use of implementing a moderate hypercone angle near $90°$. For this problem, a cone angle between 90° and 150° could result in a slightly better performance.

5.5 Choosing the Reference Space and Hypercone Angle

Although the previous section provides some insight on how the hypercone angle affects optimization results of a single problem, a more fundamental understanding of how the angle affects dominance predictions can be useful. Additionally, discussion on the reference space chosen in previous sections is largely neglected.

The reference space, much like the reference point for multi-objective HVI, is chosen somewhat arbitrarily. However, there are some guidelines that can be used to choose a meaningful reference space. First of all, the reference space is defined in the full attribute space including both objectives and parameters. Consider Table 5.1 in the following discussion as an example reference space description for a single objective and single parameter used for test problem #1. Oftentimes, the extent of the parameter space of interest is known (cf. $\boldsymbol{\theta}_{lb}$ and $\boldsymbol{\theta}_{ub}$ in Eqn. (A.1)). These upper and lower bounds of the parameter space in consideration can be directly used as part of the reference space description since an engineer is not concerned with designs outside of this parameter range. Choosing the bounds on objective dimensions is not always as easy. In this case, the objective should be minimized so the lower bound on this objective is most important[2]. The upper bound can be chosen somewhat arbitrarily such that much data will be below this bound (it is not required that all data be within the upper bound). The lower bound, however, must be a lower bound on all possible data. If the objective is a physical property, engineering intuition can be used to choose

[2] Note that if maximization is the preference, the discussion on upper and lower bounds is opposite.

Table 5.2: Reference Space Descriptions used in Figs. 5.14 and 5.15.

		Parameter	Objective
RS #1	Lower	0	0
	Upper	1	40
RS #2	Lower	0	10
	Upper	1	30

this bound. For example, if the objective is to minimize mass, there is no reason to have the lower bound be below zero (since mass cannot be negative).

Figure 5.14 illustrates the effect of choosing a reference space in coordination with the hypercone angle on a sparse data set (20 data points) from test problem #1. Cone angles of 30°, 90°, and 150° are considered along with two references spaces defined in Table 5.2. For both reference space descriptions, the bounds on the parameter are the same since they are prescribed in the problem formulation. For the objective space, RS #1 considers a larger range of objective values whereas RS #2 considers a narrow range of objective values. Both reference space descriptions are valid for the given data because the full parameter space is considered and no data falls below the lower bound on the objective value. However, the results are quite different depending on the chosen hypercone angle and reference space. As discussed previously, the smaller angle will naturally lead to more nondominated data and less dominated space and vice versa for larger angles. Since the hypercone angle is defined by the *normalized* reference space, predictive dominance is different for different reference spaces even when the same cone angle is used. In Fig. 5.14, one may notice that for hypercone angles of 90° and 150° the set of nondominated data is different for the two different reference space descriptions.

Figure 5.15 illustrates the same effect the reference space and hypercone angle on a more dense data set (200 data points) from test problem #1. What is interesting to note here is that less data is considered nondominated as the hypercone angle increases and as the range of objective values considered decreases (i.e. RS #2). To adequately perform parametric optimization, an algorithm should balance strict PPD with practical considerations. For test problem #1 the results presented

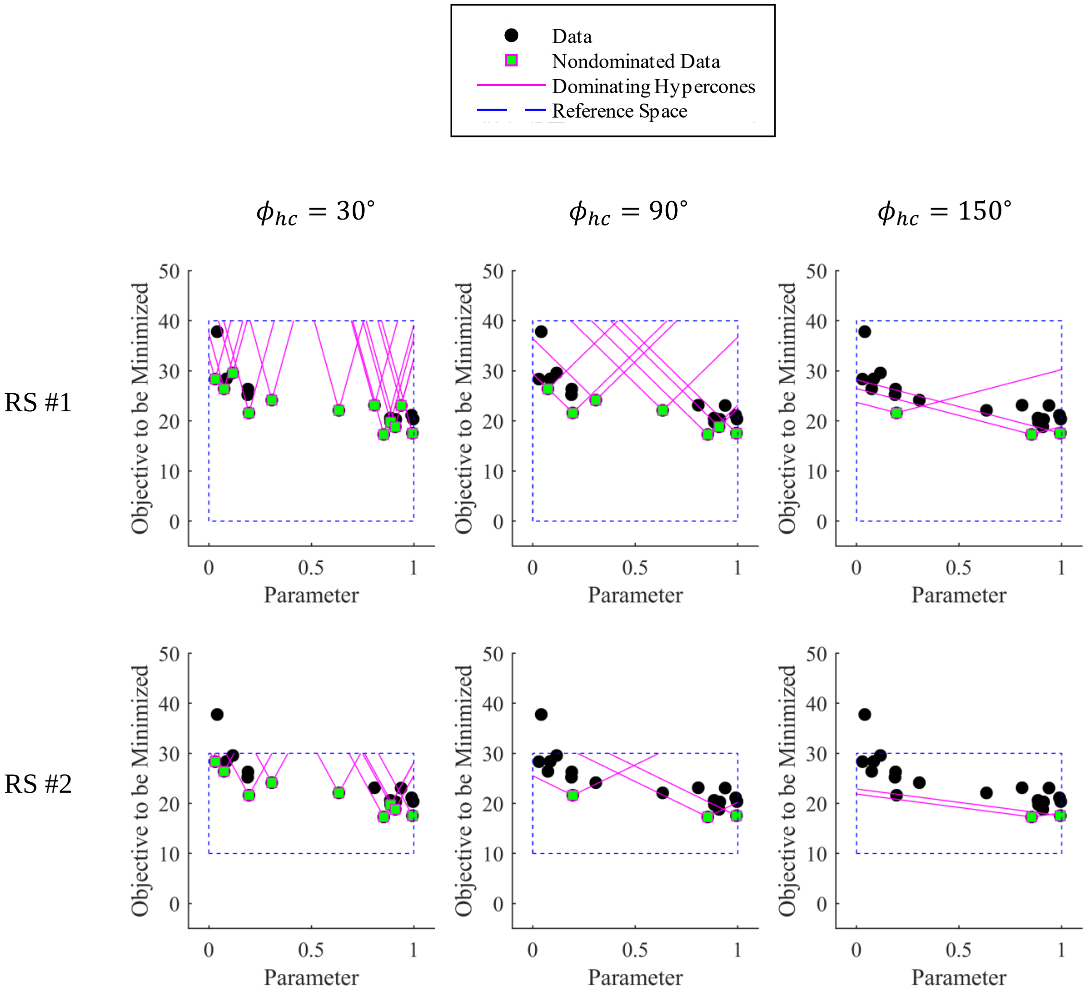


Figure 5.14: HCH visualization for different combinations of hypercone angle and reference space description (cf. Table 5.2) on a sparse data set of 20 data points from test problem #1.

in Figs. 5.14 and 5.15 suggest that RS #1 with $\phi_{hc} = 90°$ or RS #2 with $\phi_{hc} = 30°$ would yield the best predictive PPD results. For this reason, RS #1 was used in the previous sections for the HCH method.

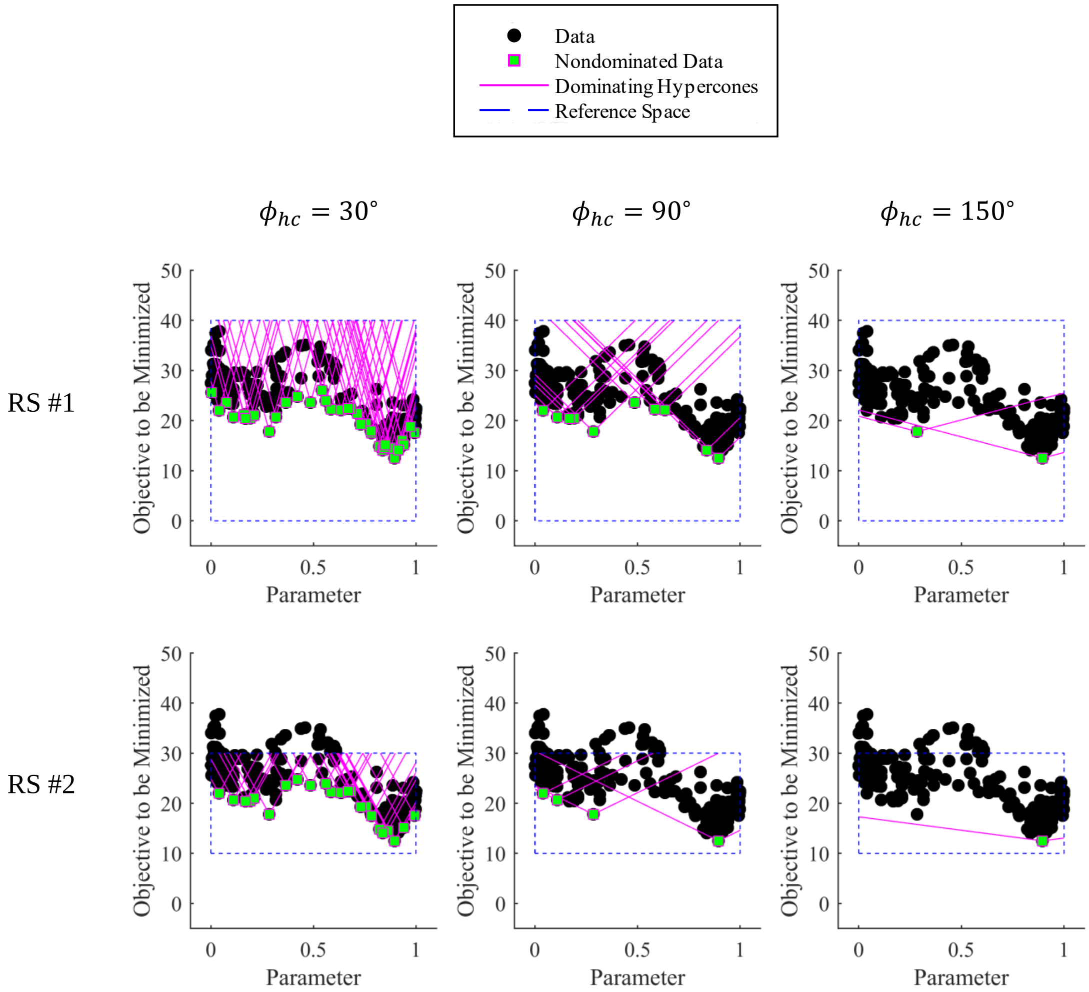


Figure 5.15: HCH visualization for different combinations of hypercone angle and reference space description (cf. Table 5.2) on a more dense data set of 200 data points from test problem #1.

5.6 Conclusions

This work introduces a new technique for calculating the parametric hypervolume indicator (pHVI) and compares the novel HCH-based method to the existing SVDD-based method. To the author's knowledge, these are the only two methods for pHVI calculation. Results indicate that pHVI trends calculated through each method are similar, but assumptions within each method understandably result in different values. pHVI values calculated from these methods should only be used to compare against other values calculated using the same method. Depending on the specific problem and data, either method could artificially produce a larger pHVI value even if the data are identical.

The HCH-based method is shown to be faster by at least an order of magnitude compared to the SVDD-based method. In addition to the reduction of computational cost, properties of the HCH-based method make it more ideal for use in evolutionary optimization algorithms because it upholds the desired multi-objective hypervolume properties of being sensitive to improvements yet unaffected by nonimprovement. This allows for further improvements in computational cost via implementation choices and allows applications of the multi-objective HVI to be implemented in parametric optimization via the pHVI.

The application of the pHVI measure to engineering design gives a designer more confidence in their solutions. In prior parametric optimization, no convergence criteria is used to halt an optimization process. Instead, only a maximum number of design evaluations is prescribed. Implementing a proper convergence criteria (such as convergence of the pHVI value; cf. Fig. 5.12a) is necessary to avoid wasting computational resources in parametric optimization by performing unnecessary analyses. More importantly, a proper convergence criteria can tell a designer that a solution is not converged and more analysis is required. Furthermore, the pHVI can be used within optimization algorithms to directly inform the search and improve an algorithm's efficiency. For example, in the existing P3GA, all nondominated data is treated equally when creating the generation of designs. With pHVI information, a new algorithm could further rank nondominated designs by their contribution to the total pHVI value thereby influencing the search more heavily toward the true parameterized Pareto frontier. In Chapter 6, the HCH-based pHVI measure is used as the basis of the novel Multi-Objective Efficient Parametric Optimization (MO-EPO) algorithm that builds upon the single objective variant [11] (cf. Chapter 4) and the expected improvement formulation of Ref. [46].

6. MULTI-OBJECTIVE EFFICIENT PARAMETRIC OPTIMIZATION

6.1 Preliminaries

The single-objective EPO algorithm provided an important first step into efficient parametric optimization. However, its limitations proved to be concerning at the least. The expected improvement formulation is extended from that of EGO with the most notable difference being the change from the scalar J_{min} in EGO to the parametric function $J_{min}(\boldsymbol{\theta})$ in EPO. Although this equation does not violate any principles of the original equation, the $J_{min}(\boldsymbol{\theta})$ is calculated in an ad hoc manner that is potentially flawed (using one approximation to make another approximation).

Consider instead a more general formulation of the expected improvement applicable when multiple objectives are present [12]:

$$EI(\boldsymbol{x}) = \int I(\mathbf{J})\, \phi_{\boldsymbol{x}}(\mathbf{J})\, d\mathbf{J}\,, \tag{6.1}$$

where $EI(\boldsymbol{x})$ is the expected improvement given the optimization variables $\boldsymbol{x}$. $I(\mathbf{J})$ is a function that outputs a scalar improvement value given a point in objective space. Finally, $\phi_{\boldsymbol{x}}(\mathbf{J})$ is the probability density function for any point in objective space given the optimization variables $\boldsymbol{x}$. Therefore, this equation states that the expected improvement is the integral of the product of the improvement of a point in objective space and the probability of achieving that point in objective space given the optimization variables $\boldsymbol{x}$ and current beliefs. The integral is taken with respect to the objectives over the feasible objective space. The relationship in Eqn. (6.1) is powerful because it makes no assumptions about the underlying relationships. Both the improvement and probability functions are up to the discretion of the designer.

Since the previous relationship is so general, one can redefine it to fit a parametric context without losing any generality. Therefore, let us define $\mathbf{F}$ as the attribute space of both objectives and parameters (i.e., $\mathbf{F} = [\boldsymbol{\theta}, \mathbf{J}]$). Then it is easy to write:

$$EI(\boldsymbol{x}) = \int I(\mathbf{F})\, \phi_{\boldsymbol{x}}(\mathbf{F})\, d\mathbf{F}\,, \tag{6.2}$$

so that the general expected improvement function now considers both objectives and parameters by considering all attributes with or without preference. All that is left is to define an improvement function that is valid for spaces with both objectives and parameters. For single objective, non-parametric problems (as in EGO), the improvement function is usually given by the reduction of the single objective value (assuming minimization):

$$I(J(\boldsymbol{x})) = (J_{min} - J(\boldsymbol{x}))\,. \tag{6.3}$$

However, the presence of additional objectives and/or parameters requires a more sophisticated improvement function.

As proposed by Emmerich *et al.* [12], a potential improvement function for multi-objective data is the HVI. Therefore, let us define a **parametric** improvement function based on the pHVI from Chapter 5:

$$I(\mathbf{F}(\boldsymbol{x})) = \mathcal{H}_p\big(\mathbf{S} \cup \mathbf{F}(\boldsymbol{x})\big) - \mathcal{H}_p(\mathbf{S})\,, \tag{6.4}$$

where $\mathcal{H}_p(\cdot)$ denotes the pHVI calculation and $\mathbf{S}$ denotes the set of existing data. Equation (6.4) then defines improvement as the amount of increase of the parametric hypervolume indicator value. Recall that the hypervolume indicator should be strictly monotonic [52, 46], so a positive value from Eqn. (6.4) indicates that $\mathbf{F}(\boldsymbol{x})$ is not dominated by any data in $\mathbf{S}$ whereas a value of exactly zero indicates that $\mathbf{F}(\boldsymbol{x})$ is dominated by at least one point in $\mathbf{S}$. A negative improvement value is not possible so long as the set S retains all existing nondominated data.

With a valid improvement function, one must now consider how to solve the expected improvement from Eqn. (6.2). From multi-objective literature, we can see that the general expected improvement formulation can be approximated by Monte Carlo integration whereby N samples in the attribute space are produced from the probability distribution $\phi_{\boldsymbol{x}}$ [12]:

$$EI(\boldsymbol{x}) \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}_p\Big(\mathbf{S} \cup \big(q_i \,|\, \phi_{\boldsymbol{x}}\big)\Big) - \mathcal{H}_p(\mathbf{S}) \tag{6.5}$$

where q_i represents the i-th Monte Carlo sample selected from the probability distribution of the attribute space given the optimization variables. Alternatively, if we assume that the probability distribution $\phi_{\boldsymbol{x}}$ follows a Gaussian distribution, Gauss quadrature rules can be used to approximate the numerical integral from Eqn. (6.2) as [89]:

$$EI(\boldsymbol{x}) \approx \sum_{i} w_i \bigg(\mathcal{H}_p\Big(\mathbf{S} \cup \big(q_i \,|\, \phi_{\boldsymbol{x}}\big)\Big) - \mathcal{H}_p(\mathbf{S}) \bigg) \tag{6.6}$$

where w_i is a weight corresponding to sample q_i dependent on the probability density function $\phi_{\boldsymbol{x}}$. Unlike a Monte-Carlo sampling, Gauss quadrature techniques prescribe the weights and samples in a structured manner. In this work, the Gauss-Legendre quadrature is used to determine the samples and weights for Eqn (6.6) [90].

6.2 An Algorithm for Multi-Objective Efficient Parametric Optimization (MO-EPO)

Building upon the structure of the single objective EPO and Eqns. (6.4) through (6.6), the Multi-Objective Efficient Parametric Optimization (MO-EPO) algorithm can be defined. The basic procedure of MO-EPO after initialization is to:

1. Train a response surface model for each expensive model,

2. Evaluate PPD via HCH to determine which data are parametrically nondominated and compute pHVI value of data,

3. Find new samples of $\boldsymbol{x}$ that parametrically maximize the expected improvement function from Eqn. (6.5) or Eqn. (6.6),

4. Evaluate some of these samples using the expensive models and add them to the full data set,

5. Repeat 1-4 until stopping criteria is met, and

6. Output information for all expensive samples including values of optimization variables, parameters, and objectives as well as the ranks of all expensive samples.

A more detailed pseudocode for MO-EPO is presented in Algorithm 5.

6.2.1 Necessary Assumptions

6.2.1.1 Smooth and Continuous Functions

The first assumption made by MO-EPO is that the expensive models can be approximated by a *smooth* and *continuous* response surface model. The most important of these is the need for the function to be continuous. This assumption is made implicitly when fitting a response surface model to the data. For most engineering applications where expensive functions are representative of or dependent on some physical phenomenon (i.e., mass, energy, etc.), this assumption is valid. In cases where the expensive models are continuous but not smooth (e.g., a piecewise linear relationship where C^0 continuity is maintained), the response surface approximation will be disadvantaged. Despite this disadvantage, the model will still approach the true solution given enough training data.

6.2.1.2 Predictive Parametric Pareto Dominance

The second assumption is that parametric Pareto dominance (PPD) can be *predicted* via an SVDD [9, 10] or the HCH (cf. Chapter 5). Although these heuristic methods are not theoretically proven, empirical evidence supports the use of these predictions of PPD under the right circumstances discussed below.

The SVDD method could be flawed if any numerical inconsistencies are experienced or if the width hyperparameter is too “loose” or too “tight”. As mentioned in Chapter 5, numerical inconsistencies are not common and could be reflective of the specific implementation used here instead of a pitfall for the method as a whole. Tax and Duin [69] illustrate and describe the effect of the width hyperparameter on the resulting SVDD. If a small value is used, the SVDD is trained to be a tight description around the data. For its application to parametric Pareto dominance (PPD), this means that there is little generalization from the training data and dominance will look similar to the strict

Algorithm 5 MO-EPO.

```
input: $\mathbf{F}(\boldsymbol{x}) = [\mathbf{J}(\boldsymbol{x}), \boldsymbol{\theta}(\boldsymbol{x})] \leftarrow$ Expensive models
input: $N_{eval}^{max} \leftarrow$ Maximum allowable number of expensive evaluations
input: $n \leftarrow$ Maximum number of expensive evaluations taken per iteration
initialize: $i = 1 \leftarrow$ Iteration counter
$\boldsymbol{x}_i \leftarrow$ Define some initial samples of $\boldsymbol{x}$ via design of experiments
$\mathbf{F}_i = \mathbf{F}(\boldsymbol{x}_i) \leftarrow$ Evaluate inital samples using expensive models
$N_{eval} = |\boldsymbol{x}_i| \leftarrow$ Count the number of expensive evaluations performed
while $N_{eval} < N_{eval}^{max}$ do
    $\hat{\mathbf{F}}(\boldsymbol{x}) = Kriging\_Interp(\boldsymbol{x}_i, \mathbf{F}_i)$
    $\mathcal{H}_p^i,\ ranks = Update(\mathbf{F}_i)$
    $\boldsymbol{x}^*(\hat{\boldsymbol{\theta}}) = \text{argmax}_{\boldsymbol{x}}\ EI(\boldsymbol{x}) \,|\, \mathcal{H}_p^i$
    if all$(EI(\boldsymbol{x}^*(\hat{\boldsymbol{\theta}})) \,|\, \mathcal{H}_p^i < tol)$ then
        break while
    else
        Define $\boldsymbol{x}_{new}$ as $n$ tuples such that: $\boldsymbol{x}_{new} \subseteq \boldsymbol{x}^*(\hat{\boldsymbol{\theta}})$
        $\mathbf{F}_{new} = \mathbf{F}(\boldsymbol{x}_{new})$
        $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i \cup \boldsymbol{x}_{new}$
        $\mathbf{F}_{i+1} = \mathbf{F}_i \cup \mathbf{F}_{new}$
        $N_{eval} = N_{eval} + n$
        $i = i + 1$
    end if
end while
$\mathcal{H}_p^i,\ ranks = Update(\mathbf{F}_i)$
return $\boldsymbol{x}_i, \mathbf{F}_i, ranks$

definition $Update(\mathbf{F})$
    $pHVI = \mathcal{H}_p(\mathbf{F}) \leftarrow$ Calculate pHVI value
    $ranks \leftarrow$ Rank all points in F via HCH
    return $pHVI, ranks$
end definition

definition $Kriging\_Interp(\boldsymbol{x}, \mathbf{F})$
    $\hat{\mathbf{F}}(\boldsymbol{x}) \leftarrow$ Train a Kriging model of each expensive model given the known data in $\{\boldsymbol{x}, \mathbf{F}\}$
    return $\hat{\mathbf{F}}(\boldsymbol{x})$
end definition
```

PPD rules whereby much data will be considered as nondominated. On the other hand, a large width hyperparameter results in a loose description of the SVDD. This means that the description over-generalizes the data. For PPD in this case, the danger is that much of the nondominated data will be incorrectly classified as dominated. Despite this potential pitfall, Galvan *et al.* [5] demonstrate the method's effectiveness using a moderate value for the width hyperparameter.

The HCH method also has hyperparameters that factor into its trustworthiness. Similar to the width hyperparameter of the SVDD, the HCH requires the value for the hypercone angle. In similar fashion to its analogous hyperparameter in the SVDD method, too small of a cone angle will result in little generalization where much data is considered as nondominated. For a large cone angle (with 180° being its upper bound), the parameters become less important in determining dominance as objective values become more and more important. As the cone angle nears 180°, the predictive parametric Pareto dominance begins to look more and more like traditional Pareto dominance (which negates the consideration of parameters altogether). An additional limitation to the HCH method is the need for a user-defined reference space (similar to the reference point chosen for multi-objective HVI calculations). Both the reference space and the hypercone angle are needed to fully define the HCH to predict PPD. A user may avoid these pitfalls by choosing a hypercone angle and reference space consistent with the intuition discussed in Section 5.5.

6.2.2 Implementation-Specific Details

Although Algorithm 5 provides a general overview of MO-EPO, there are some details specific to the current implementation worth mentioning. First of all, the algorithm allows for the input of both expensive *and* inexpensive models (cf. line 1 in Algorithm 5). Inexpensive models are evaluated exactly when necessary (i.e., there are no response surface models for inexpensive functions). The sample initialization (cf. line 5) is either input by a user or generated via Latin hypercube sampling (LHS) though this could be updated to any number of sampling techniques. As expressed in line 9, the response surface model used in this implementation is a Kriging interpolation via the DACE toolbox [63]. Then, the Kriging predictions of both value and standard error are used to define a Gaussian probability distribution ϕ throughout the attribute space $\mathbf{F}$ for a given set of optimization variables $\boldsymbol{x}$.

The optimization of the response surface model in line 12 is performed via P3GA [10] to generate a number of design configurations with high expected improvement across the parameter space. Depending on the number of expensive functions, the expected improvement is calculated via Eqn. (6.5) or Eqn. (6.6). When the number of expensive functions is less than or equal to 4, the Legendre-Gauss quadrature is used to define a grid with a resolution of 5 samples in each dimension (of the expensive functions) along with their corresponding weights using the implementation of Greg von Winckel [91] to evaluate Eqn. (6.6). If the number of expensive functions is greater than 4, a random sampling of 1,000 points (i.e., $N = 1,000$) are taken from the probability density function to evaluate Eqn. (6.5). In either case, the improvement function used is the improvement of the parametric hypervolume indicator presented in Eqn. (6.4).

The n tuples chosen from P3GA solution is done on the basis of a score calculated as the product of the expected improvement of the point and its euclidean distance to the nearest neighbor via Eqn. (6.7). $EI(\boldsymbol{x})$ represents the expected improvement of design $\boldsymbol{x}$, $\mathbf{X}$ represents the set of new designs that have already been chosen, and $d_{NN}(\cdot)$ represents the euclidean distance between $\boldsymbol{x}$ and its nearest neighbor in $\mathbf{X}$. Therefore, this crowding metric penalizes similar data and promotes a more diverse set of new samples. If $n = 1$ (cf. line 15 in Algorithm 5), the point with greatest expected improvement is chosen since the set $\mathbf{X}$ is empty. If $n > 1$, each chosen point is added to $\mathbf{X}$ and scores are updated before choosing a new point. Once all n points are chosen in this way, x_{new} is simply the set of points in $\mathbf{X}$.

$$\text{score}(\boldsymbol{x}|\mathbf{X}) = EI(\boldsymbol{x}) \times d_{NN}(\boldsymbol{x}, \mathbf{X}) \tag{6.7}$$

6.2.3 User Options

In its current implementation, MO-EPO has many options that can be set by the user. Aside from the inputs labeled in Algorithm 5, a user can choose to modify the conditions for training the response surface model. As the algorithm currently accomodates a Kriging model only via the DACE toolbox [63], one could easily change the correlation or regression functions. Default options include using the Gaussian correlation function and a second order regression function.

Since MO-EPO hinges on the improvement of the parametric hypervolume indicator, options such as the reference space and hypercone angle can also affect the algorithm performance. The default value of the hypercone angle is 90 degrees. There is an option to automatically define the reference space based on the initial population, but this is not recommended since the bounds of the initial population are not always representative of the entire feasible space. Instead, a user should input reference bounds based on the parameter ranges of interest and objective goals. For example, in Chapter 3, the parameter ranges are set in the problem formulation (cf. Eqn. (3.1)) and can be used directly for the reference space. For the lift-to-drag ratio objective, the lower bound of the reference space can be zero (since negative lift-to-drag ratios are not meaningful in this context) and the upper bound can be placed using some intuition based on engineering knowledge. For example, if the maximum lift-to-drag ratio for the *fixed* airfoil is about 100, the reference space will need to sufficiently extend beyond 100 to account for the maximum performance of the *morphing* airfoil (which is currently unknown). Knowing the true solution, we can see that an upper bound of 200 on lift-to-drag reference space is sufficient, but a larger upper bound would still be valid. The upper bound only becomes invalid when the true solution exceeds the bounds on the reference space. For example, if the morphing airfoil is capable of achieving lift-to-drag ratios up to 180 under certain conditions, having the upper bound of lift-to-drag equal to a number less than 180 would result in misleading pHVI values, though dominance calculations would remain in tact.

Another user input is related to stopping criteria. By default, the MO-EPO algorithm will run until the full budget of expensive evaluations has been spent. However, one may choose to detect convergence by modifying the maximum amount of expected improvement necessary to continue (cf. tol on line 12 of Algorithm 5). The default value of this tolerance is zero to promote spending the whole computational budget, but a user could input a small positive tolerance that could allow MO-EPO to stop when expectation of improvement falls below the user-defined threshold.

Finally, the last user input affects the transfer of information from one iteration to the next (cf. line 8). When P3GA is used to find data with high expected improvement (line 11), only n of these points are used to improve the solution and/or response surface model (lines 15-18). Some or all of the leftover data from P3GA at iteration i can be used to seed the maximization of expected improvement at iteration $i + 1$. In some cases, the expected improvement will drop due to the new information. However, in other cases, there are points in design space that will continue to have high expected improvement until there is a new expensive sample taken in its vicinity. By transferring this information from one iteration to the next, MO-EPO can consider promising designs in one iteration when the designs were unable to be sampled in the previous iteration. The trade-off of this is increased computational time of the MO-EPO algorithm. The default value of this is zero meaning that no information is shared from one iteration to the next, but a user could increase this number to be any positive integer.

6.3 Benchmarking

To demonstrate the MO-EPO algorithm performance, the three test problems in Appendix A are considered. Of interest is the the solution accuracy given by the parametric hypervolume indicator (pHVI) as a function of both the number of design evaluations and the computational time assuming an evaluation time for the analytic test problems[1]. Since the true pHVI value can be calculated for these analytic problems, the pHVI value is presented as a percentage of the true value. For each problem, all combinations of the number of objectives and parameters are considered up to a four-dimensional attribute space. Purely multi-objective formulations without consideration of parameters are ignored since the MO-EPO method is built to be used for multi-objective parametric problems. Although MO-EPO will solve non-parametric problems, there are more efficient methods that should be used when parameters are not present (cf. Section 2.2).

In the following results, both methods are initialized by a Latin hypercube sampling. P3GA is run with 50 generations of 50 population members for a total of 2,500 design evaluations along

[1] All calculations are performed on the same machine: Intel(R) Core(TM) i7-9700 @ 3.00 GHz CPU with 32 GB of RAM

with the options listed in Table 6.1. MO-EPO, on the other hand, is allowed to take 10 expensive samples per iteration with a maximum budget of 1,000 design evaluations (including the initial Latin hypercube samples). Other specifics for each test problem are given in the following subsections.

Table 6.1: P3GA options for all three test problems.

Population Size	Number of Generations	SVDD Width Parameter	Crossover Fraction	Mutation Fraction
50	50	4	0.80	0.01

6.3.1 Scalable Test Problem #1

The first test problem considered is given by Eqn. (A.1) with five optimization variables. This test problem proposed by Galvan *et al.* [5] is characterized by a discontinuous and nonconvex solution. Furthermore, the analytical problem has a biased solution density. This means that it is very unlikely to find the true solution by a random sampling. These three characteristics provide a meaningful benchmarking example.

By treating all objectives and parameters as expensive, MO-EPO can be employed to solve the problem. For both P3GA and MO-EPO, the reference space is bounded by 0 and 40 for the final objective. All other functions are bounded from 0 to 1. Additionally, both methods use a hypercone angle of 90° (note that although P3GA does not use the HCH for dominance assessment, calculation of the pHVI value requires a hypercone angle and reference space to be defined). For MO-EPO, 0 points of interest will carry over from one iteration to the next (cf. Section 6.2.3) and the algorithm will run until either: the maximum number of design evaluations has been met or no point with positive expected improvement can be found.

Figure 6.1 presents the results from five replications of both MO-EPO and the existing P3GA [10] using the SVDD for PPD. The figure plots the solution pHVI as a function of the number of design evaluations. Note that a single design evaluation consists of evaluating all objective and parameter functions for a single vector of optimization variables $\boldsymbol{x}$. Each row of Fig. 6.1 denotes the dimensionality of the attribute space (i.e., the number of objectives plus the number of parameters) whereas each column denotes the number of parameters. The trend shows that MO-EPO consistently finds a better solution with fewer design evaluations compared to P3GA for all problem formulations here. Figure 6.2 shows the same data focused on the regions of fewer design evaluations.

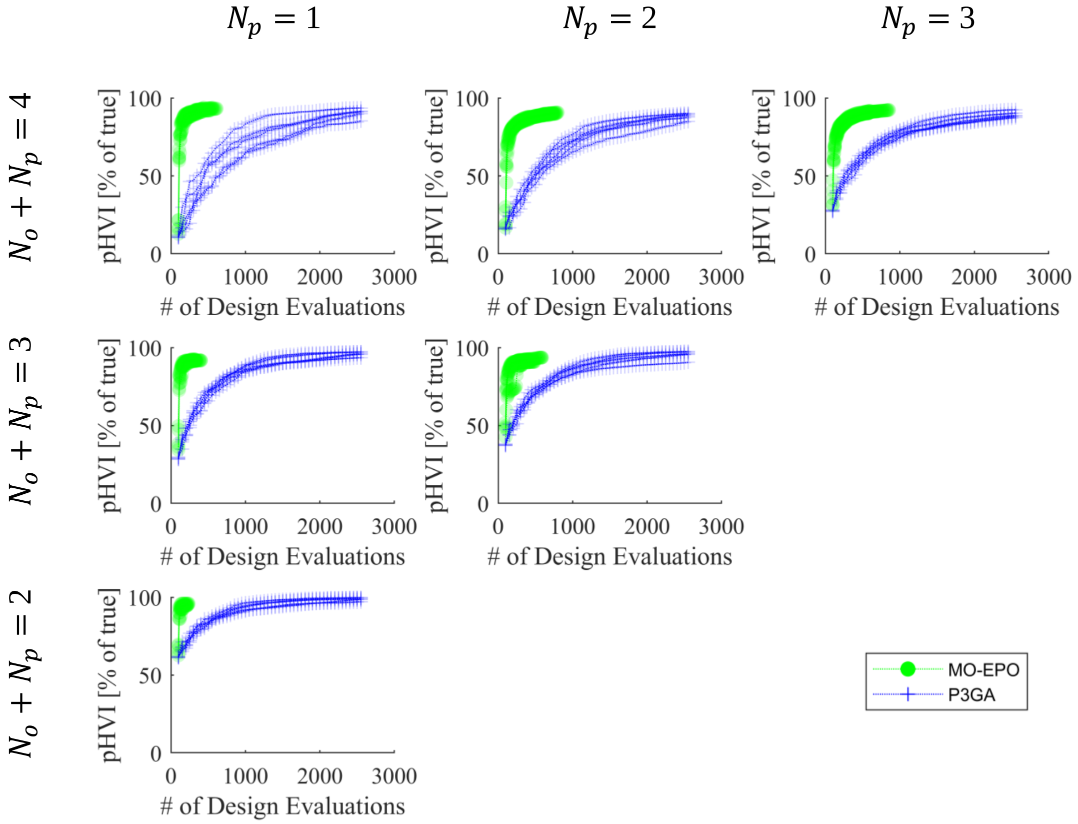


Figure 6.1: pHVI versus the number of design evaluations for test problem 1 (cf. Eqn (A.1)) considering all parameter and objective functions as expensive.

These results are compelling, but there is one factor not considered in these figures: the computational "cost" of algorithmic overhead. P3GA requires the training of a SVDD in each iteration

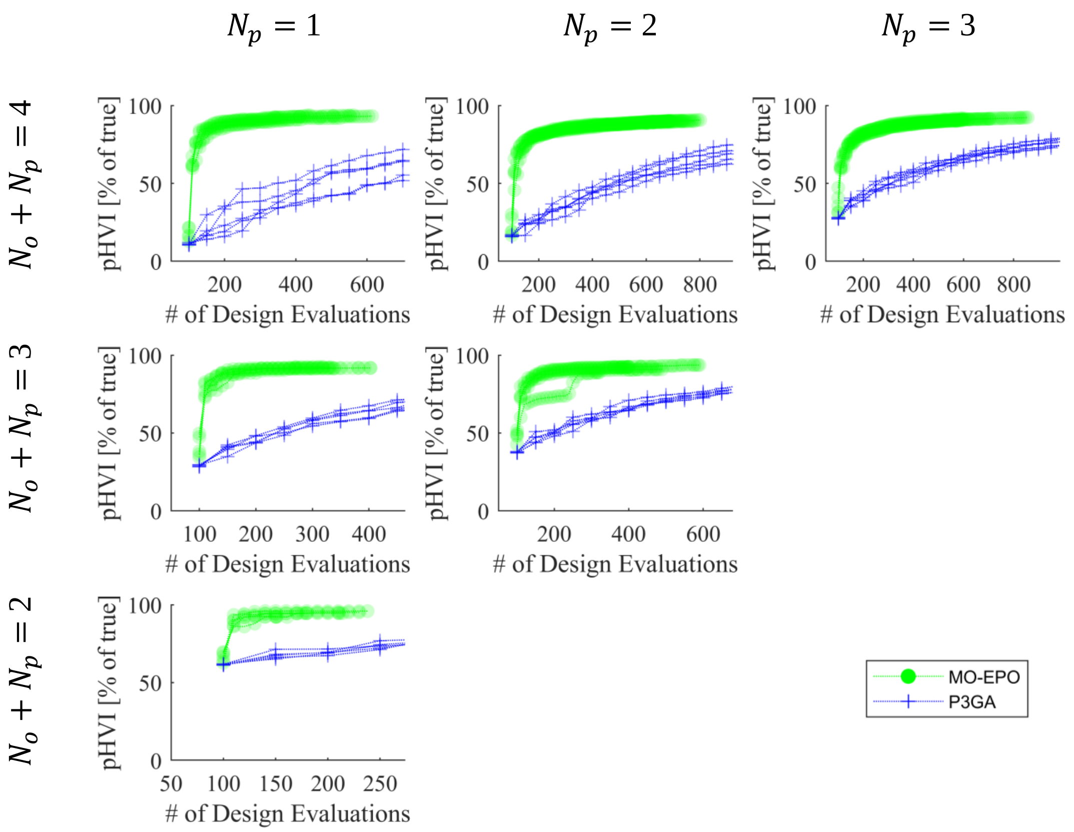


Figure 6.2: A close up view of the results presented in Fig. 6.1 to visualize the trends at low numbers of design evaluations.

whereas MO-EPO requires the training of surrogate models for each expensive function and the optimization of the expected improvement using these models in each iteration. To account for this overhead cost, Fig. 6.3 displays the pHVI value as a function of computational time (i.e., wall-clock time) assuming that each design evaluation requires 1 minute (serial computation). Results in Fig. 6.3 are very similar to that of Fig. 6.2. This shows that the overhead cost of MO-EPO does not diminish the results on this problem.

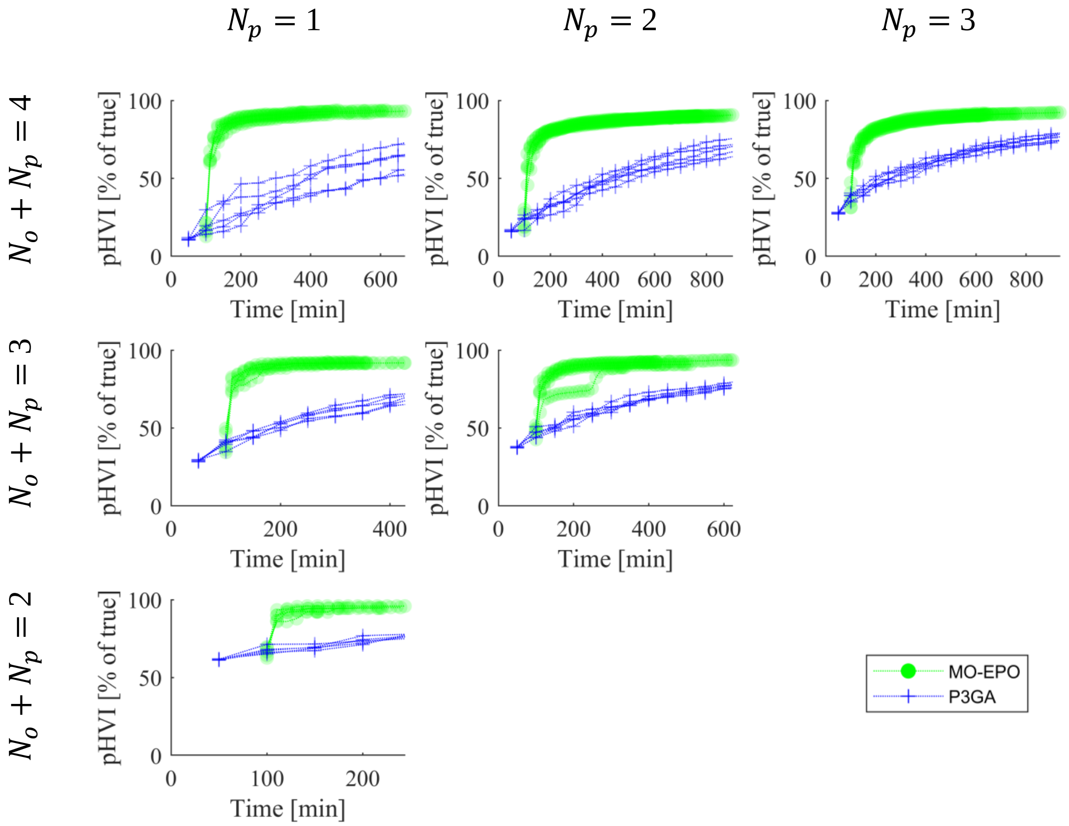


Figure 6.3: pHVI versus the wall-clock time assuming that each design evaluation requires 1 minute for test problem 1 (cf. Eqn (A.1)) considering all parameter and objective functions as expensive.

As discussed in Section 2.1, parameters can be dependent or independent. When parameters are defined by expensive functions, they are dependent. However, it is also possible for parameters to be independent and therefore cheap[2]. Therefore, the previous study is repeated for both P3GA and MO-EPO considering all parameters as independent and cheap. In this case MO-EPO allows 1,000 randomly chosen designs of interest (i.e., $EI(\boldsymbol{x}) > 0$) to be carried over from one iteration to the next. Figures 6.4 and 6.5 illustrate the superiority of MO-EPO over P3GA on this problem

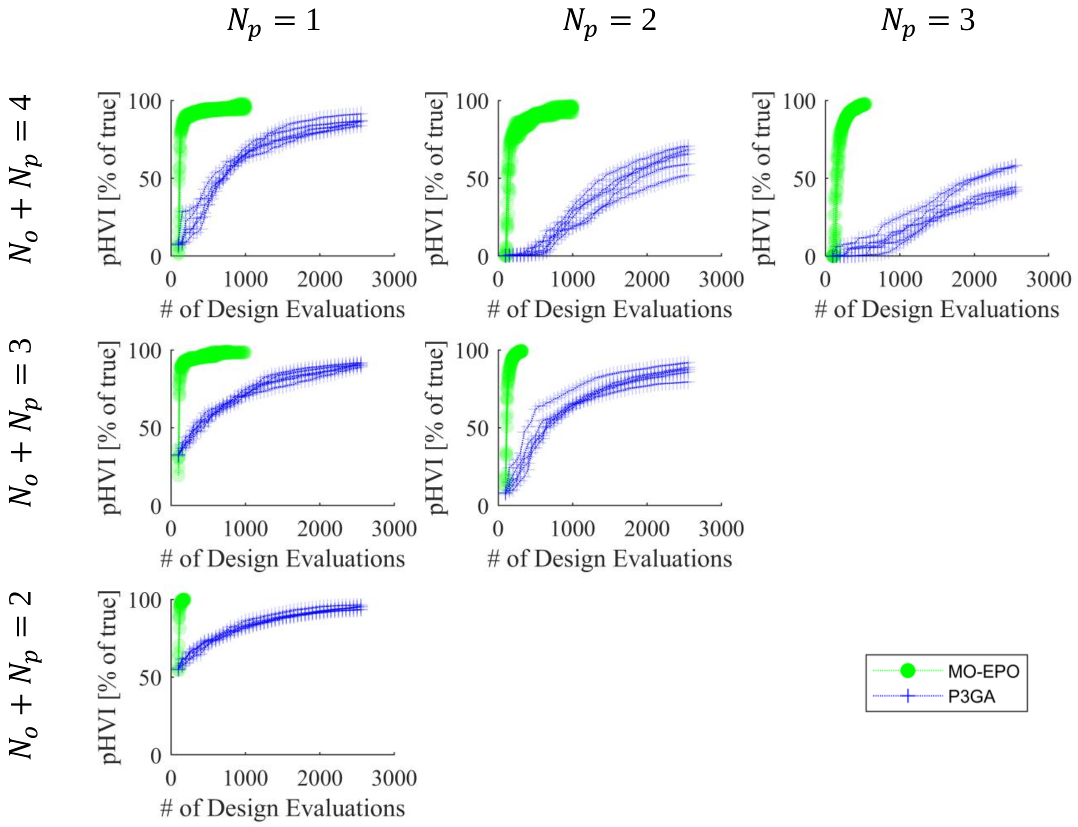


Figure 6.4: pHVI versus the number of design evaluations for test problem 1 (cf. Eqn (A.1)) considering all objective functions as expensive and all parameters as cheap and independent.

[2] Independent parameters are always cheap, but cheap parameters are not always independent. I.e., parameters can be both cheap and dependent.

with independent parameters. This is not surprising since P3GA relies on a naive search while MO-EPO is able to use cheap functions and models of expensive functions to predict optimal solutions more quickly. Of particular note here is that MO-EPO performs increasingly better as more parameters are considered (since these parameters are evaluated quickly) whereas P3GA pales in comparison.

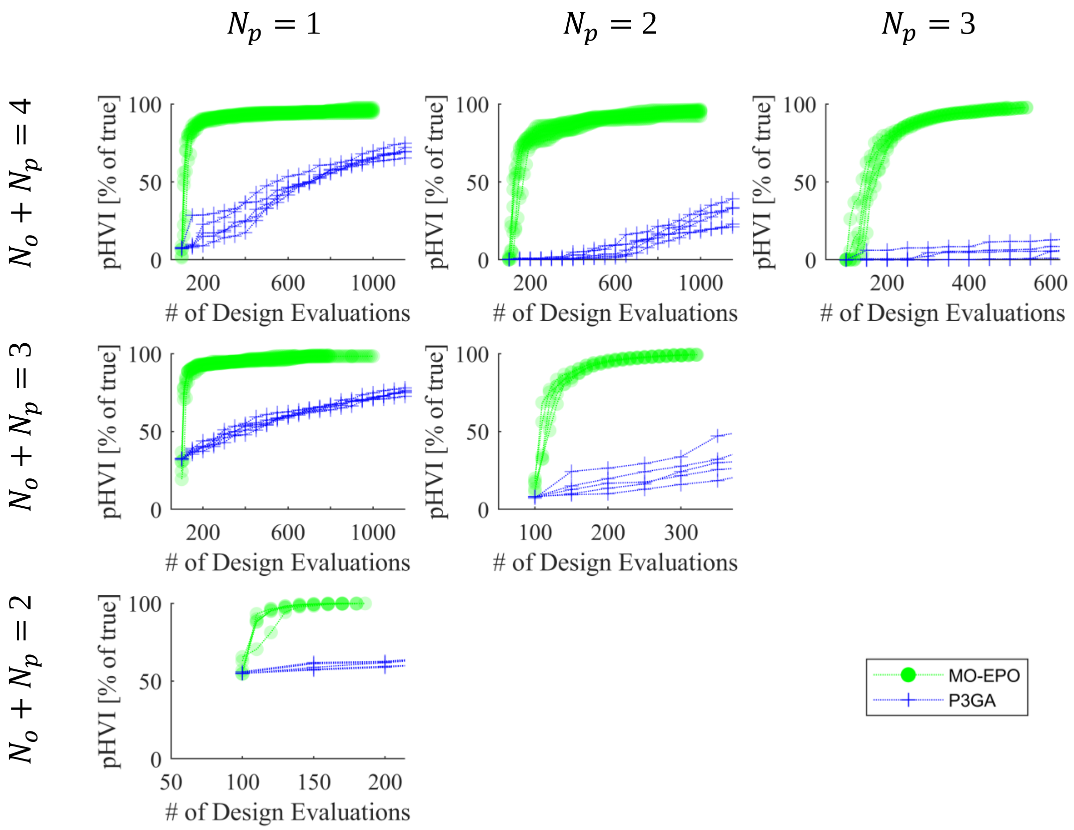


Figure 6.5: A close up view of the results presented in Fig. 6.4 to visualize the trends at low numbers of design evaluations.

6.3.2 Scalable Test Problem #2

The second test problem considered is given by Eqn. (A.2) with five optimization variables. This test problem proposed by Galvan *et al.* [5] is also characterized by a biased solution density away from the true solution. In contrast with the first test problem, this second problem exhibits a **variable** solution density on the parametric Pareto frontier making certain parts of the true solution more difficult to find than others. Finally, this problem exhibits high multimodality with many local optima. These characteristics provide a meaningful and different benchmarking example.

Similar to the previous test problem, all objectives and parameters are considered as expensive, and MO-EPO is employed to solve the problem. For both P3GA and MO-EPO, the reference space is bounded by 0 and 40 for the final objective. All other functions are bounded from 0 to 1. Additionally, both methods use a hypercone angle of 90°. For MO-EPO, 0 points of interest will carry over from one iteration to the next (cf. Section 6.2.3) and the algorithm will run until either: the maximum number of design evaluations has been met or no point with positive expected improvement can be found.

Figures 6.6 and 6.7 present the pHVI versus the number of design evaluations and computational time (assuming 1 minute per design evaluation), respectively. Similar to the performance of EPO on the single objective version of this problem (cf. Section 4.2.3), MO-EPO is unable to outperform the existing P3GA with respect to either computational time or the number of design evaluations. When comparing the characteristics of this test problem to the assumptions made within MO-EPO, these results are not surprising. In Section 6.2.1, the assumption that functions are “smooth and continuous” is made. Although this test problem is continuous, it is highly multimodal leading to non-smooth objective/parameter functions overall. To generate a trustworthy response surface model, much data is needed to capture the multimodality of this problem. Therefore, the solution improvement is relatively slow as a result of an untrustworthy response surface model. Despite this assumption violation and disadvantage, MO-EPO and P3GA performance are largely the same on this difficult problem for low numbers of design evaluations.

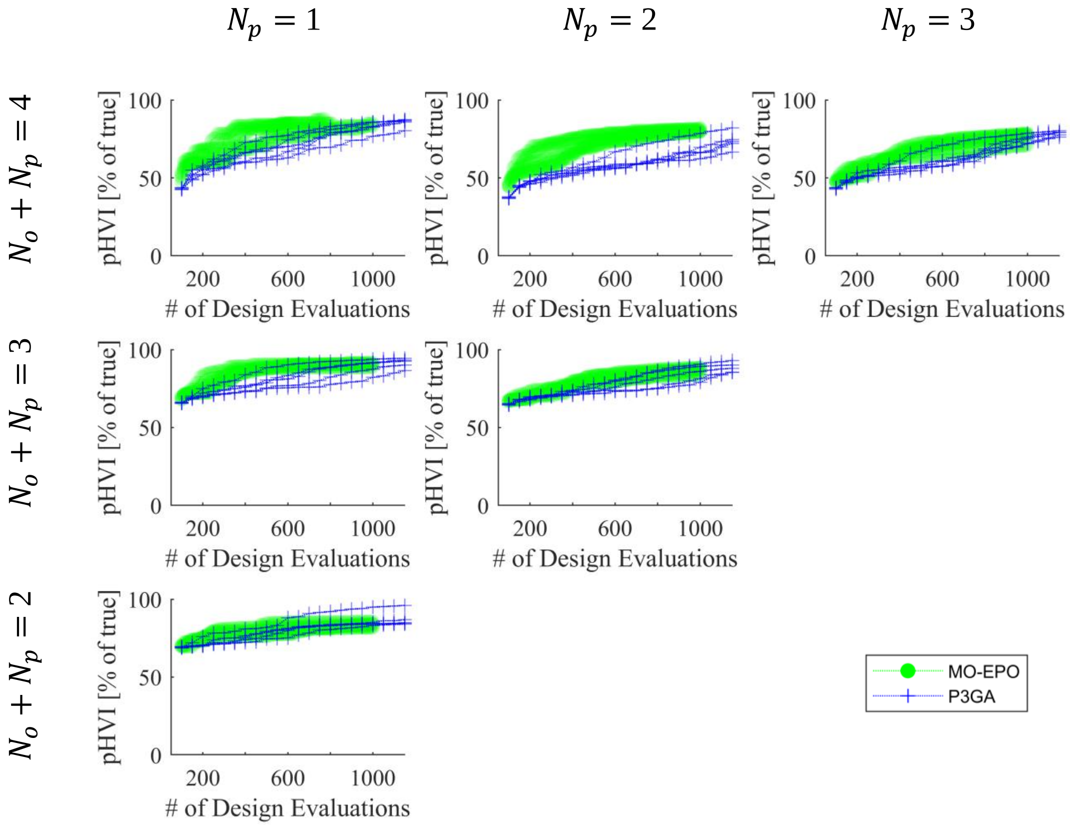


Figure 6.6: pHVI versus the number of design evaluations for test problem 2 (cf. Eqn (A.2)) considering all parameter and objective functions as expensive.

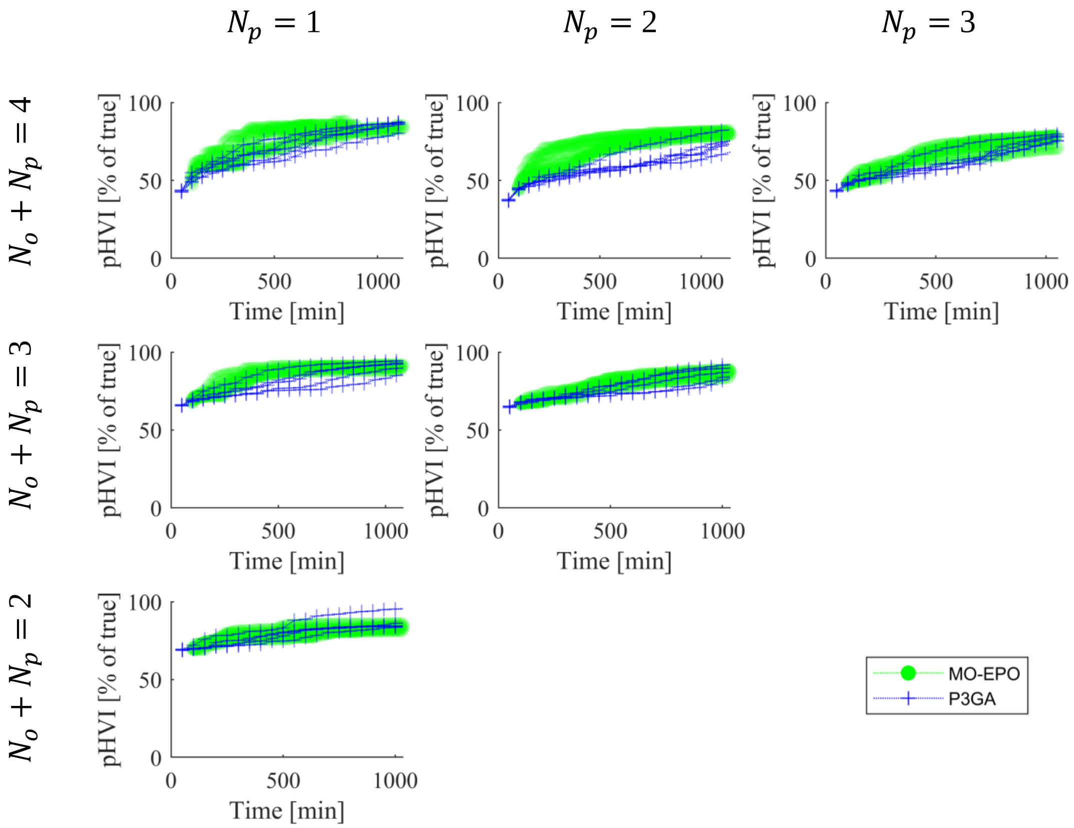


Figure 6.7: pHVI versus the wall-clock time assuming that each design evaluation requires 1 minute for test problem 2 (cf. Eqn (A.2)) considering all parameter and objective functions as expensive.

6.3.3 Scalable Test Problem #3

The third and final test problem considered is given by Eqn. (A.3) with five optimization variables. This test problem was proposed by Deb *et al.* as a purely multi-objective optimization problem. However, by treating some objective functions as parameter functions, a parametric test problem is formed that is used to test the hypercone angle. In the purely multi-objective form, the solution is a linear Pareto frontier. When parameters are considered, the solution becomes a piecewise linear parametric Pareto frontier. Part of this parametric Pareto frontier is characterized by a relatively steep slope that can pose a challenge for the hypercone heuristic.

For this problem, the reference space is bounded by 0 and 20 in every attribute and a hypercone angle ϕ_{hc} of 50° is employed for both MO-EPO and pHVI calculations. For MO-EPO, 100 points of interest will carry over from one iteration to the next (cf. Section 6.2.3) and the algorithm will run until either: the maximum number of design evaluations has been met or no point with positive expected improvement can be found.

Figures 6.8 and 6.9 present the pHVI versus the number of design evaluations and computational time (assuming 5 minutes per design evaluation), respectively. Due to the shape of the parameterized Pareto frontier, P3GA could have a slight advantage over MO-EPO. However, the results relative to both the number of design evaluations and the computational time show MO-EPO as the more efficient method. Comparing Figs. 6.8 and 6.9 reveals that the overhead time of MO-EPO is increased on this problem. This is a result of the smaller hypercone angle that, although necessary to correctly identify the true parametric Pareto frontier, increases the computational overhead. Section 5.4.2 briefly discusses two ways that the HCH-based pHVI method can be implemented efficiently by using dominance information from one iteration of an evolutionary algorithm to the next. In this case, a smaller hypercone angle results in less dominated data finally resulting in an increase in computational expense for computing the pHVI value used for expected improvement estimation. In this example, relatively cheap high-fidelity functions (<1 minute per evaluation) could be more easily optimized using something like P3GA instead of MO-EPO. As the design evaluations become more expensive, MO-EPO is most efficient.

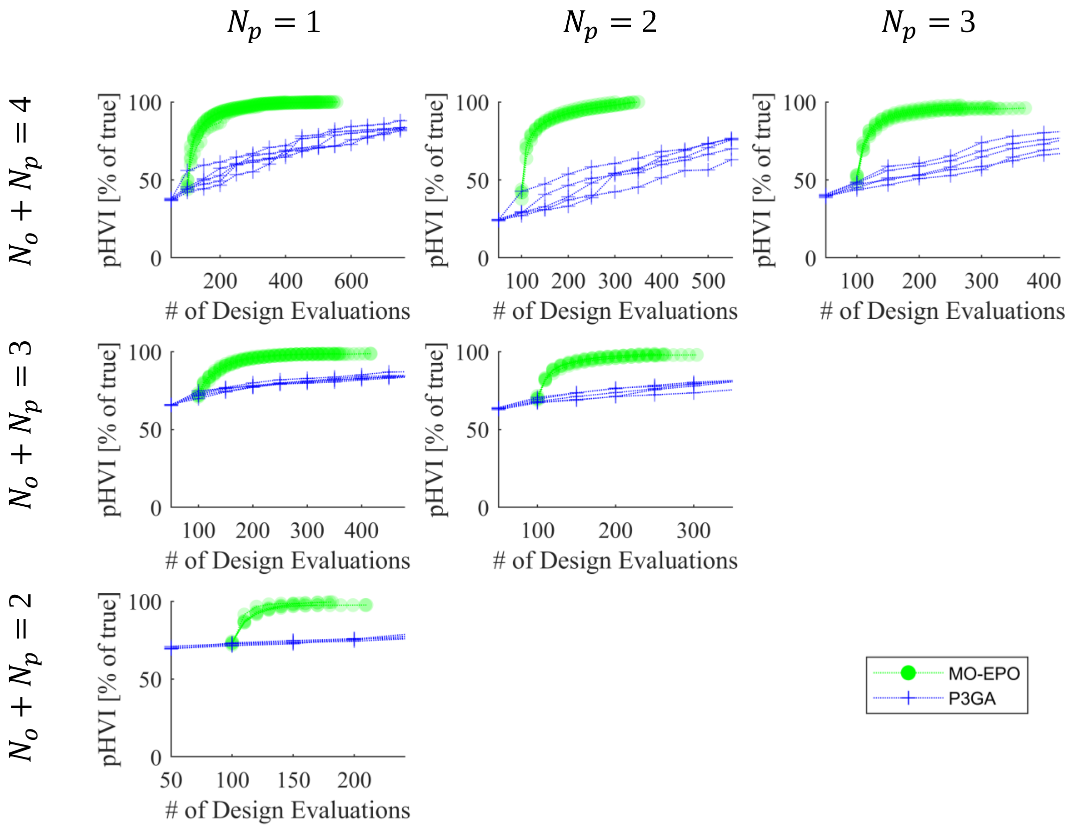


Figure 6.8: pHVI versus the number of design evaluations for test problem 3 (cf. Eqn (A.3)) considering all parameter and objective functions as expensive.

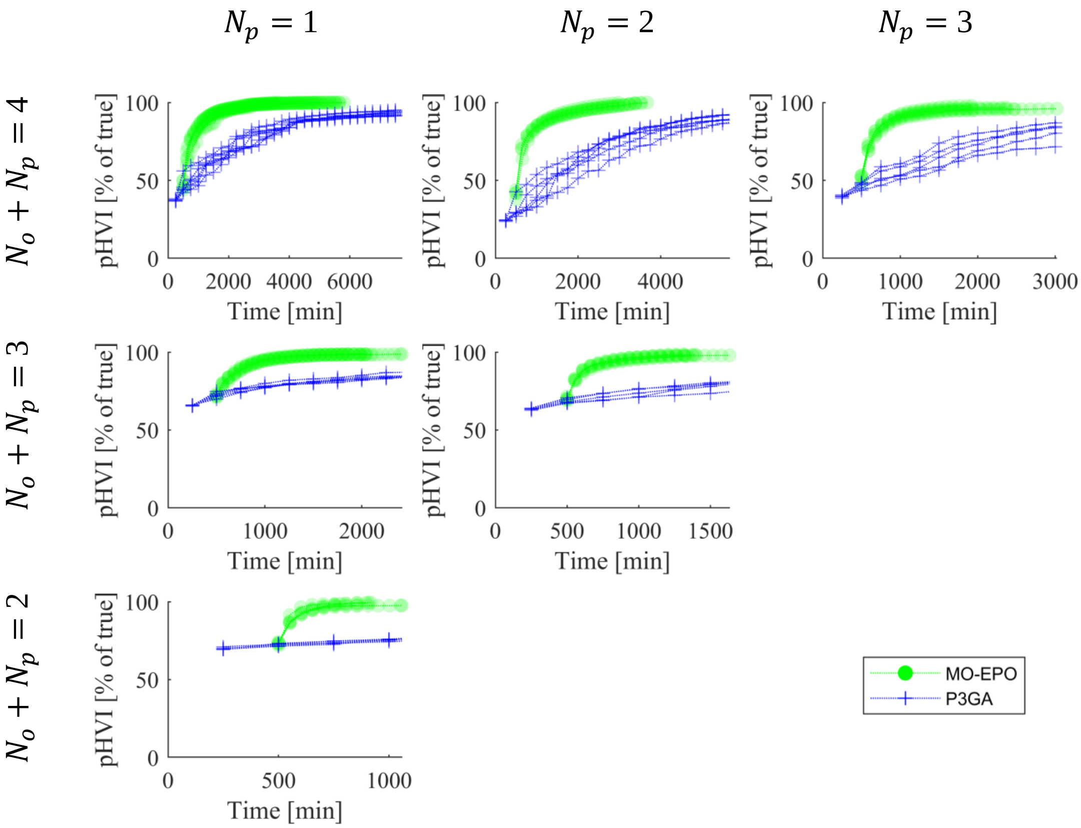


Figure 6.9: pHVI versus the wall-clock time assuming that each design evaluation requires 5 minutes for test problem 3 (cf. Eqn (A.3)) considering all parameter and objective functions as expensive.

7. PARAMETRIC OPTIMIZATION OF MORPHING AIRFOIL

This chapter demonstrates the use of the novel MO-EPO on the optimization of the morphing airfoil presented in Chapter 3. Results from MO-EPO are compared to that of P3GA considering solution quality (via pHVI), computational effort, and total number of expensive analyses. The computational effort considered is the combination of algorithmic overhead and time spent on expensive analyses. To allow for a side-by-side comparison with previous studies, the single objective problem is solved with MO-EPO and compared to the solution from P3GA in Section 7.1. However, since MO-EPO is fundamentally a *multi-objective* algorithm, an additional objective is considered in Section 7.2. This multi-objective problem is solved using both P3GA and MO-EPO.

Unlike the previous chapter, the pHVI of the true solution is not known for this engineering case study. Therefore, absolute pHVI values are presented instead of a fraction of a true value. Note that a pHVI value of exactly 1 is not possible; instead the pHVI value denotes the ratio of the reference space dominated by the data (cf. Section 4.2.1).

When comparing computational times in this chapter, please take note of the following. All reported times include both the time to perform analysis and the full algorithmic time for MO-EPO or P3GA. All expensive analyses via XFOIL are performed serially. In some cases, the combination of airfoil shape, airspeed, and angle of attack can produce an unstable CFD analysis. Therefore, the analysis code includes the capability to detect and terminate such unstable analyses. Such a capability is necessary, but it does not allow for parallel analysis of airfoil shapes. Optimization of the response surface model within MO-EPO (cf. line 11 of Algorithm 5), however, can be evaluated in parallel because CFD (or any expensive analysis) is not involved. All optimization and analyses are performed on the same machine[1] The analysis of a single airfoil shape via XFOIL requires 6+ seconds (in rare cases, analysis can take up to 2 minutes).

[1] Intel(R) Xeon(R) - W3570 @ 3.20 GHz CPU with 12 GB of RAM

7.1 Single Objective Formulation

The single objective problem is the same problem presented in Eqn. (3.1) of Chapter 3. The single objective is to maximize the lift-to-drag ratio of a camber morphing airfoil by varying the four shape coefficients $\mathbf{P}^c$. The three parameters considered are angle of attack α, airspeed V, and altitude H. The problem formulation is reprinted in Eqn. (7.1) for reader convenience.

$$\begin{gathered} J^*(\alpha, V, H) = \max_{\mathbf{P}^c} \; J(\mathbf{P}^c; \alpha, V, H) \\ \text{Subject to:} \\ \mathbf{P}^c \in \mathbb{P} \\ \begin{bmatrix} 0^\circ \\ 20 \text{ m/s} \\ 5000 \text{ ft} \end{bmatrix} \leq \begin{bmatrix} \alpha \\ V \\ H \end{bmatrix} \leq \begin{bmatrix} 12^\circ \\ 65 \text{ m/s} \\ 40000 \text{ ft} \end{bmatrix} \end{gathered} \tag{7.1}$$

In Chapter 3 this problem is solved using P3GA allowing for 400 generations of 150 population members. This yields a total of 60,000 possible analyses. To solve this problem with MO-EPO the options in Table 7.1 are used. The hypercone angle used for calculating the parametric hypervolume indicator (pHVI) and for assessing parametric Pareto dominance (PPD) is 90°. The maximum number of expensive analyses is set at 1,000 where up to 10 new designs are considered in each iteration of the algorithm. The expected improvement tolerance is set to zero to force the algorithm to continue until it either reaches the maximum number of expensive analyses or is unable to find any point with nonzero expected improvement. Finally, 100 points of nonzero expected improvement from one iteration (these are not evaluated with expensive analyses) are used to initialize the expected improvement maximization for the following iteration (cf. line 11 of Algorithm 5). The reference space is defined by Table 7.2.

To enable further comparison between P3GA and MO-EPO, the same 76 feasible population

Table 7.1: MO-EPO options to solve Eqn. (7.1). Refer to Algorithm 5.

ϕ_{hc}	N_{eval}^{max}	n	tol
90°	1,000	10	0

Table 7.2: Bounds of the reference space used to calculate the pHVI for solutions to Eqn. (7.1). Note that the reference space is comprised of all attributes (parameters and objectives).

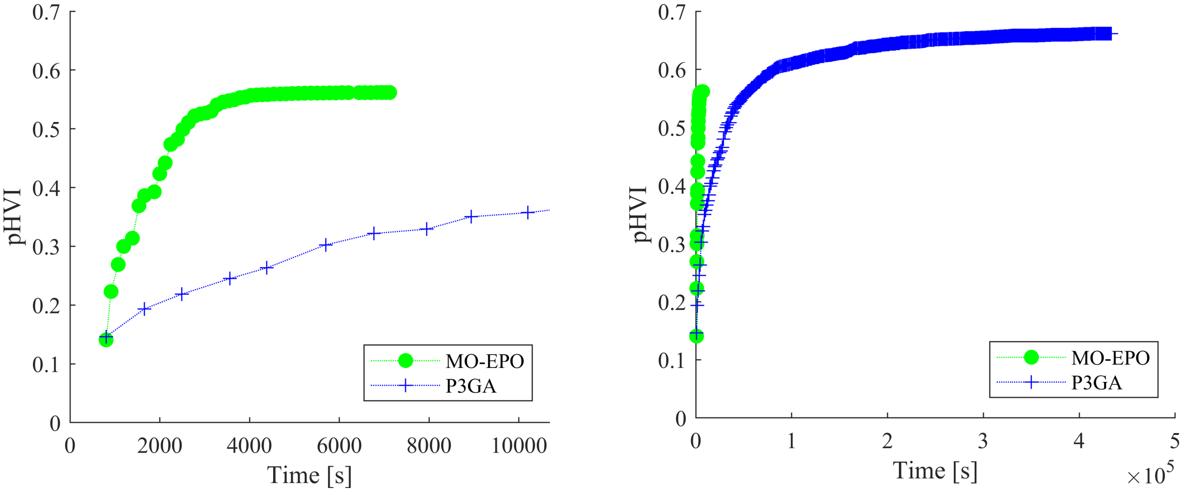
	α [°]	V [m/s]	H [ft]	J [-]
Lower	0	20	5,000	0
Upper	12	65	40,000	180

members[2] are used to initialize each method. Despite given a budget of 1,000 total analyses, MO-EPO converges after only 518 function evaluations in 47 iterations requiring about 2 hours of computational time. Figure 7.1a shows the pHVI as a function of time for both P3GA and MO-EPO on this multi-parametric, single objective problem. It is clear that MO-EPO is capable of finding a better solution than P3GA given a short time constraint. Figure 7.1b, however, shows what happens if P3GA is given more time for analysis. In this image, it becomes clear that P3GA is capable of finding a better solution without this time constraint.

In addition to considering pHVI as a function of time to take into account the computational cost of analysis and optimization, Fig. 7.2 illustrates the pHVI as a function of the number of design evaluations. The underlying data is identical for Figs. 7.1 and 7.2, but comparing the number of design evaluations can be beneficial to understand how MO-EPO may perform on a different case study.

It is encouraging to see MO-EPO outperform P3GA in this case considering both time and number of design evaluations, but the main limitation is that MO-EPO converged prematurely due to the response surface model. After 48 iterations of MO-EPO, the response surface model is confident in its predictions and no more points will have positive expected improvement.

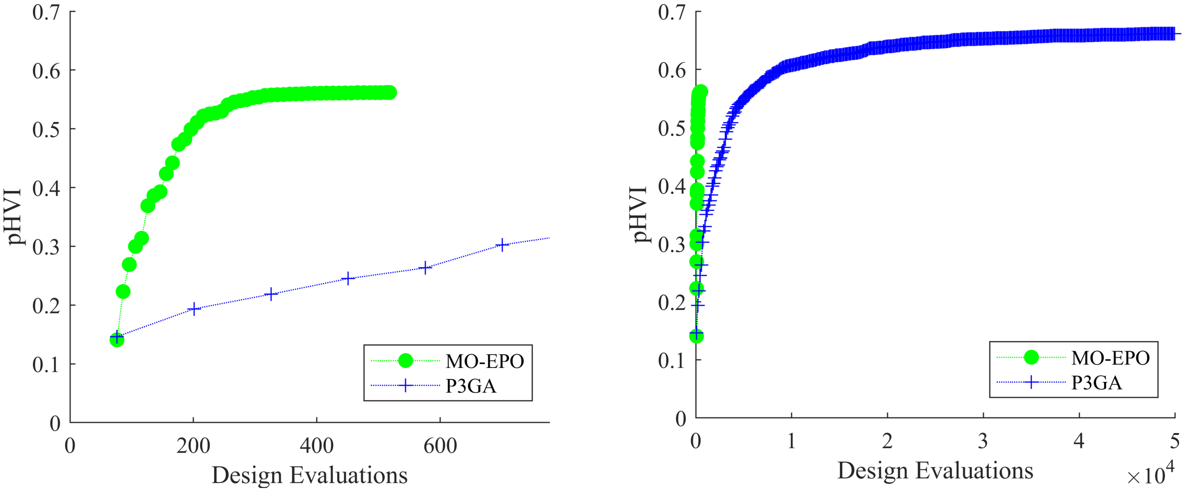
[2] Although P3GA has a larger population size, only the feasible members of this population are used to initialize MO-EPO.



(a)

(b)

Figure 7.1: pHVI value as a function of time for P3GA and MO-EPO for the single objective problem (Eqn. (7.1)) on the range of (a) 0-10,000 seconds and (b) 0-500,000 seconds.



(a)

(b)

Figure 7.2: pHVI value as a function of the number of designs evaluated for P3GA and MO-EPO for the single objective problem (Eqn. (7.1)) on the range of (a) 0-750 design evaluations and (b) 0-50,000 design evaluations.

Although the pHVI value for the MO-EPO result is about 15% less than that of P3GA, much of the solution from P3GA is found in the solution from MO-EPO. Plotting lower dimensional subspaces of the full, four-dimensional attribute space (one objective and three parameters) reveals a very similar overall trend. Figure 7.3 shows the parametrically nondominated designs from the full 50,000 designs evaluated by P3GA compared to all of the 518 designs evaluated by MO-EPO.It is clear that MO-EPO is able to gather much information about the solution to the problem defined in Eqn. (7.1) with only a fraction of the time and analysis required by P3GA.

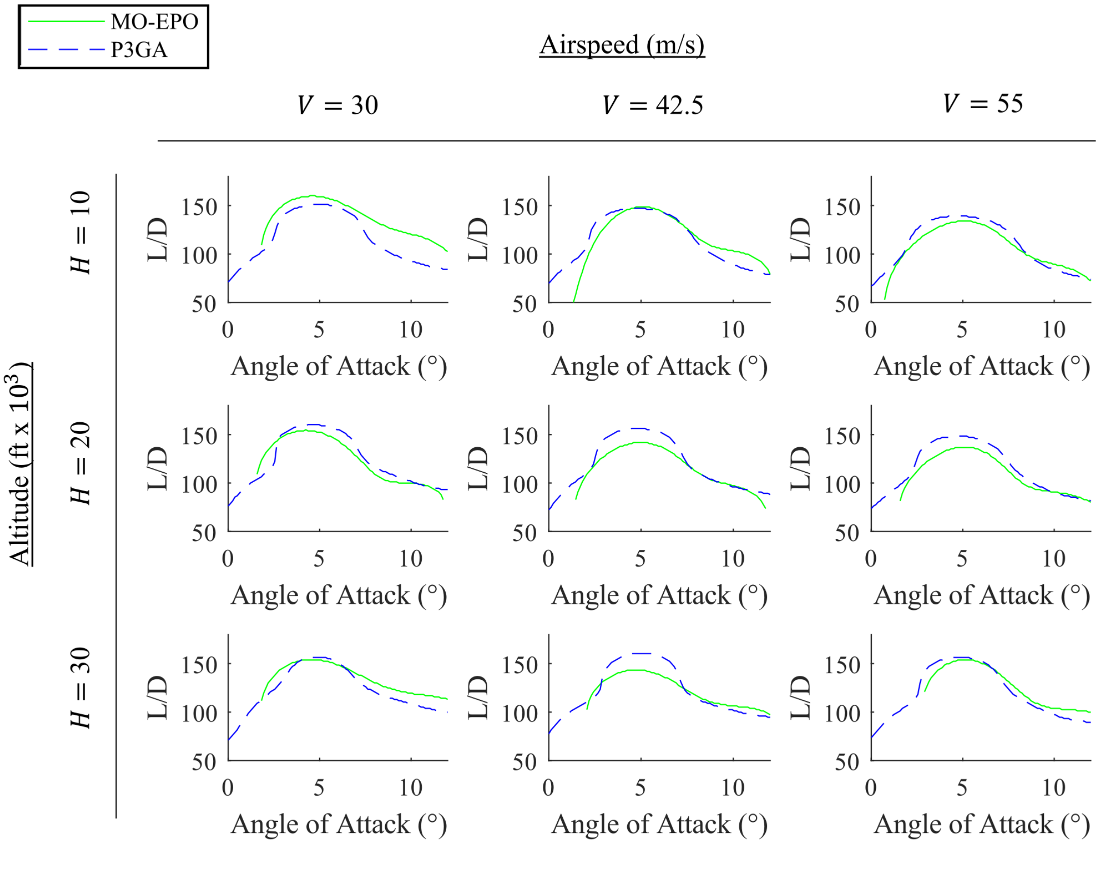


Figure 7.3: Parametric frontiers from P3GA requiring 50,000 design evaluations versus that of MO-EPO requiring 518 design evaluations for multiple 2D subspaces of the full 4D attribute space.

7.2 Multi-Objective Formulation

Consider the addition of a second objective. While maximization of the lift-to-drag ratio is a valid objective, one might also prefer to minimize the magnitude of the moment coefficient C_M. If this moment coefficient is close to zero the added load necessary to achieve force balance on the airfoil is reduced. Then the updated problem formulation is thus:

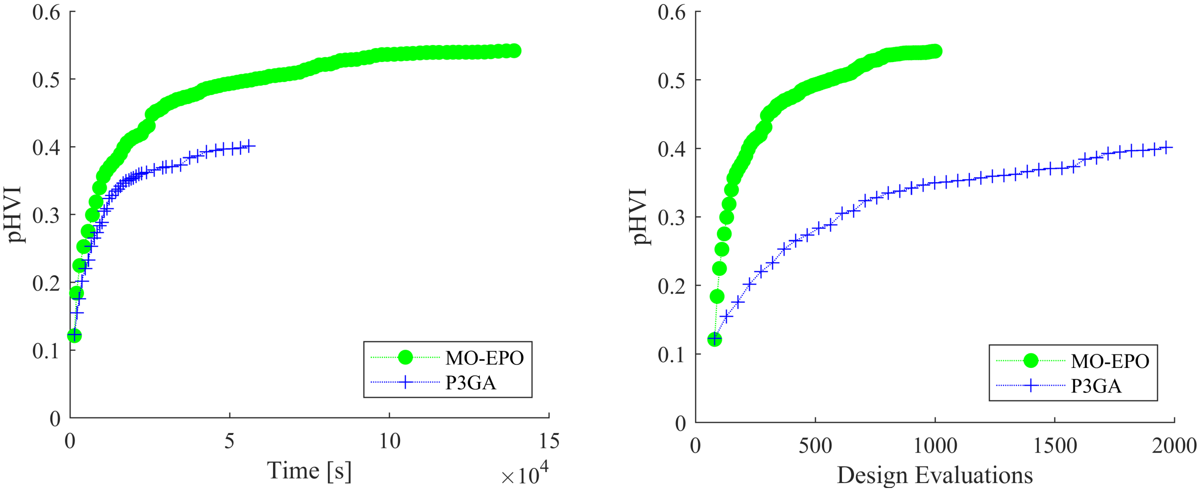
$$\begin{gathered}
\mathbf{J}^*(\alpha, V, H) = \left[\max_{\mathbf{P}^c} \; J_1(\mathbf{P}^c; \alpha, V, H), \; \min_{\mathbf{P}^c} \; J_2(\mathbf{P}^c; \alpha, V, H) \right], \\
\text{Subject to:} \\
\mathbf{P}^c \in \mathbb{P}, \\
\begin{bmatrix} 0^\circ \\ 20 \text{ m/s} \\ 5000 \text{ ft} \end{bmatrix} \leq \begin{bmatrix} \alpha \\ V \\ H \end{bmatrix} \leq \begin{bmatrix} 12^\circ \\ 65 \text{ m/s} \\ 40000 \text{ ft} \end{bmatrix},
\end{gathered} \tag{7.2}$$

where J_1 is the lift-to-drag ratio calculated via XFOIL to be maximized and J_2 is the magnitude of the coefficient of moment C_M to be minimized. Note that both of these objectives are expensive since XFOIL is involved.

To solve this problem with P3GA, 40 generations of 50 population members are considered. For MO-EPO, the same settings are used from Table 7.1. For both methods, the same 80 population members are used for initialization, and the pHVI is calculated considering the reference space defined in Table 7.3. Figure 7.4 presents the solution quality for both methods as a function of time and the number of design evaluations. These results show the same trend as that of the single-objective formulation. Whether looking at the computational time or the number of design evaluations, MO-EPO is able to find a higher quality solution faster than P3GA. In this image, the overhead time of MO-EPO can be seen. Despite MO-EPO considering half of the total design evaluations as P3GA, the computational time of MO-EPO is nearly three times that of P3GA. The results indicate, however, that this algorithmic overhead is worth the expense as MO-EPO is still

Table 7.3: Bounds of the reference space used to calculate the pHVI for solutions to Eqn. (7.2). Note that the reference space is comprised of all attributes (parameters and objectives).

	α [°]	V [m/s]	H [ft]	J_1 [-]	J_2 [-]
Lower	0	20	5,000	0	0
Upper	12	65	40,000	180	0.5



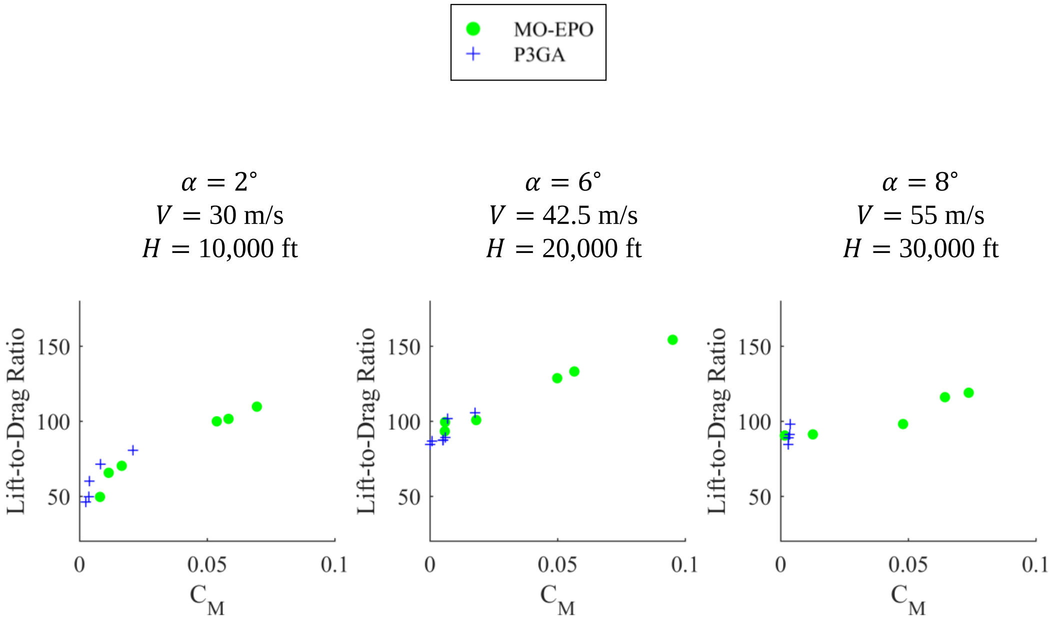
(a) (b)

Figure 7.4: pHVI value as a function of (a) time and (b) the number of designs evaluated for P3GA and MO-EPO for the multi-objective problem (Eqn. (7.2)).

able to outperform P3GA. Figure 7.5 illustrates three subspaces of the five-dimensional parameterized Pareto frontier (two objectives plus three parameters)[3]. In each of these parameter spaces, the Pareto frontier found by MO-EPO is able to find designs with larger lift-to-drag ratios compared to that of P3GA. In the short time that each algorithm was allowed to run, P3GA was only able to find designs with low moment coefficient whereas MO-EPO found a larger variety of designs to give a designer tradeoff information between maximizing lift-to-drag ratio and minimizing moment coefficient.

Figure 7.6 presents two airfoil shapes determined by MO-EPO for similar angle of attack and

[3] To plot scattered data in each subspace, data with $\alpha \pm 1^\circ$, $V \pm 10\,\text{m/s}$, and $H \pm 5,000\,\text{ft}$ of the printed value are projected into the given space.



(a)

(b)

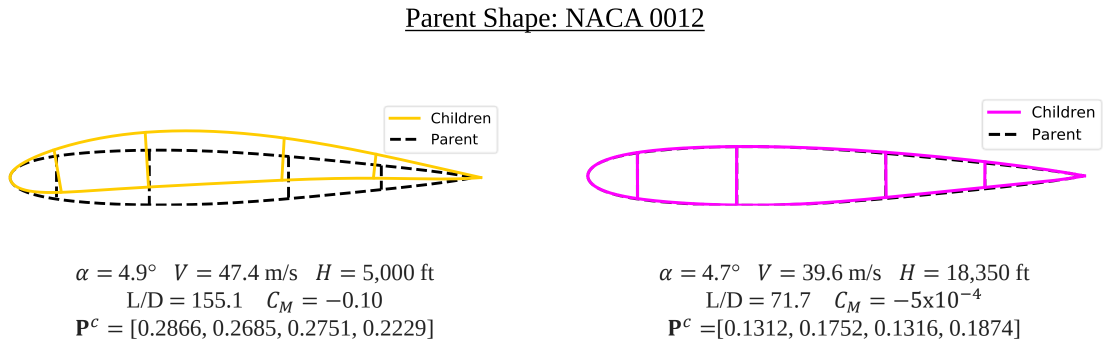
(c)

Figure 7.5: Parametrically nondominated designs from P3GA requiring 2,000 design evaluations versus all designs from MO-EPO requiring 1,000 design evaluations for multiple 2D subspaces of the full 5D attribute space. Lift-to-drag ratio should be maximized while moment coefficient should be minimized.

airspeed values. One design prioritizes maximizing lift-to-drag ratio (Fig. 7.6a) whereas the other design prioritizes minimizing the moment coefficient magnitude (Fig. 7.6b). The resulting shapes are not surprising as the high lift-to-drag airfoil (Fig. 7.6a) resembles those found in Chapter 3 while the low moment magnitude airfoil (Fig. 7.6b) resembles the parent NACA 0012 shape.

7.3 Conclusions

This chapter demonstrates the ability of MO-EPO to solve a real-world engineering problem involving a morphing airfoil with three parameters and one or two objectives. When only a limited number of function evaluations can be taken, MO-EPO is able to produce higher quality solutions compared to P3GA. Re-solving the problem formulation presented in Chapter 3 reveals that MO-EPO is able to gather similar information to what is gathered from P3GA with much less


Parent Shape: NACA 0012
Children
Parent
Children
Parent
$\alpha = 4.9°$ $V = 47.4$ m/s $H = 5{,}000$ ft
L/D = 155.1 $C_M = -0.10$
$\mathbf{P}^c$ = [0.2866, 0.2685, 0.2751, 0.2229]
$\alpha = 4.7°$ $V = 39.6$ m/s $H = 18{,}350$ ft
L/D = 71.7 $C_M = -5\text{x}10^{-4}$
$\mathbf{P}^c$ =[0.1312, 0.1752, 0.1316, 0.1874]

(a) (b)

Figure 7.6: Optimal morphed airfoil shapes found from MO-EPO prioritizing (a) lift-to-drag ratio and (b) moment coefficient magnitude.

computational time and less design evaluations. This capability is necessary for problems whose design evaluations become even more costly, and evaluating 50,000 designs is prohibitively expensive. The only limitation of MO-EPO identified with this engineering example is the premature convergence when the response surface model predicts that no additional data would be beneficial. Despite this premature convergence, the final solution is of high quality and can provide an engineer with much information. If time/resources permit, results from MO-EPO can be used to bootstrap another algorithm like P3GA in hopes of finding a better solution. However, if time and resources are limited, MO-EPO is the best option for finding the best possible solution.

8. PARAMETRIC OPTIMIZATION OF MORPHING SUPERSONIC CIVILIAN TRANSPORT*

8.1 Motivation

The ability of a supersonic aircraft to adapt its outer mold-line (OML) in response to local flight conditions offers the possibility for maintaining low en-route noise in circumstances not considered during the design process. Designing an adaptive OML provides the opportunity for combining established tools and design methods into a new framework which can efficiently explore the OML design space to maintain low en-route noise. The resulting computational framework must balance high fidelity analysis tools and constraints related to actuator design with computational efficiency to prevent unacceptable time penalties from slowing the overall development of the aircraft. OML adaptivity affords aircraft designers the ability to go beyond classical design philosophies that relied upon a single OML tailored to meet multiple performance objectives and open the design space to more efficiently meet mission goals and constraints.

The efforts by industry, NASA, and academia to design supersonic aircraft with fixed OMLs have been well documented [93, 94, 95, 96, 97, 98, 99, 100, 53, 101]. These design efforts used a wide range of aerodynamic analysis tools with varying fidelity, such as panel codes [102], Euler solvers [103], Reynolds-averaged Navier–Stokes (RANS) solvers [104], and space-marching approaches [105]. The equivalent area representation of the aerodynamics model from classical sonic boom theory has also been used during the conceptual design phase for fast sonic boom and wave drag predictions [53, 106]. In all of the design studies, the aerodynamic models were coupled with atmospheric wave propagation solvers to determine both the strength and the perceived level in decibels (PLdB) of the coalesced shock waves on the ground [107, 108, 109].

Methods for design optimization have been used to develop low-boom supersonic aircraft.

*Part of this chapter is reprinted with permission from “Computational Design Methodology of Adaptive Outer Mold Line for Low En-Route Noise of a Supersonic Aircraft” by Weaver-Rosen, J.M., Carpenter F.L., Cizmas P.G.A., Malak, R.J., Abraham, T.A., Hunsaker, D.F., and Lazzara D.S., 2021. *In Proceedings of the AIAA SciTech Forum*, AIAA 2021-0877 (16 pages), Copyright 2021 by Forrest L. Carpenter [92].

Specifically, adjoint methods combined with computational fluid dynamics (CFD) tools allow for design problems to be solved more quickly using constrained gradient-based optimization compared to finite-difference approaches [110, 111, 112, 113]. Additionally, response surface methods and evolutionary algorithms have benefited the design of supersonic aircraft [114], especially when taking into account multiple objectives and, more recently, varying operating parameters (e.g., Mach, angle-of-attack, and atmospheric profiles) [115]. Despite the advantages gained by optimization techniques, the design of supersonic aircraft is usually only focused on finding a single OML optimized for a single set of flight conditions. Since this is not representative of all possible flight conditions, it is more beneficial to explore an OML capable of adapting to external conditions [2]. Therefore, design optimization must take into account these variable operating conditions to assess an adaptive OML performance and return a set of viable OML configurations for various conditions.

The ground signatures of low-boom aircraft configurations are known to be sensitive to a number of flight conditions. These flight conditions include the atmospheric profiles, weather, and terrain covered by the flight path [116, 117]. The changes to local Mach number and angle-of-attack of the aircraft that are imposed by these conditions result in loudness variations over the course of a flight [118]. This work demonstrates the use of the novel MO-EPO to optimize the equivalent area representation of the aircraft considering local Mach number and angle-of-attack as exogenous parameters. The objectives considered are to minimize noise (PLdB) and to minimize the amount of required actuation.

8.2 Equivalent Area Description

Due to the many degrees of design freedom coupled with costly analysis, performing optimization of a full CFD simulation can quickly become time-consuming and intractable. Therefore, researchers propose that preliminary optimization be performed by directly modifying the equivalent area distribution of the aircraft as discussed by Abraham *et al.* [115] to focus the optimization of full CFD models. The equivalent area distribution of a supersonic aircraft is computed by incrementally adding components of the geometric area with a lift-derived area term. In this work, equivalent area distributions are determined from the nearfield pressure signature by [119]:

$$A_{E,r}(x) = 4\frac{\left(2R\sqrt{M_\infty^2 - 1}\right)^{\frac{1}{2}}}{\gamma M_\infty^2} \int\limits_0^x \left(\frac{\Delta p}{p_\infty}\right)(\xi - x_0)\sqrt{x - \xi}\, d\xi, \tag{8.1}$$

where x_0 is the x-location of the initial non-zero value of the nearfield signature. Equation (8.1) was solved at each location along the nearfield pressure signature to produce the full equivalent area distribution, and was used exclusively in this work to compute the equivalent area distribution from a given CFD solution.

Figure 8.1 shows the baseline equivalent area distribution for the NASA Configuration 25D aircraft computed using an inviscid nearfield solution and Eqn. (8.1) for the designed Mach number and angle-of-attack. This baseline equivalent area distribution can then be deformed by adding or subtracting area defined by two cubic splines. Figure 8.2 illustrates one such equivalent area deformation, where the deformation was defined by its longitudinal location, peak magnitude, and fore/aft lengths. Such a deformation description allows for both symmetric deformations, as utilized by Abraham *et al.* [115], and asymmetric deformations.

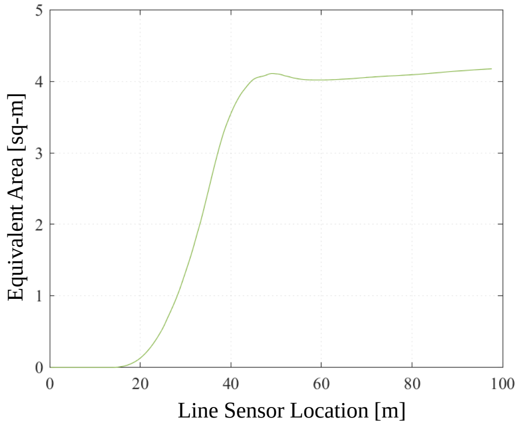


Figure 8.1: Baseline inviscid equivalent area distribution of the NASA Configuration 25D.

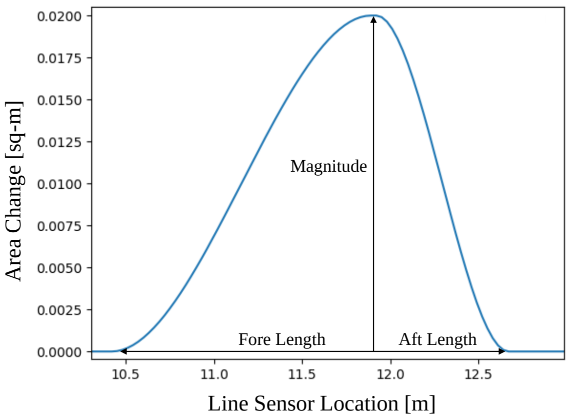


Figure 8.2: Spline-based equivalent area distribution modification schematic.

8.2.1 Equivalent Area Surrogate Model

One particular challenge to the parametric optimization herein is that the baseline equivalent area is not constant, but rather dependent upon the local Mach number and angle-of-attack as illustrated in Fig. 8.3. Typically, CFD must be performed for each variation in Mach number and angle-of-attack to produce the baseline equivalent area before any area changes can be applied for optimization purposes. Instead of overburdening CFD resources, a surrogate model can be trained to output the full equivalent area distribution given the local Mach number and angle-of-attack.

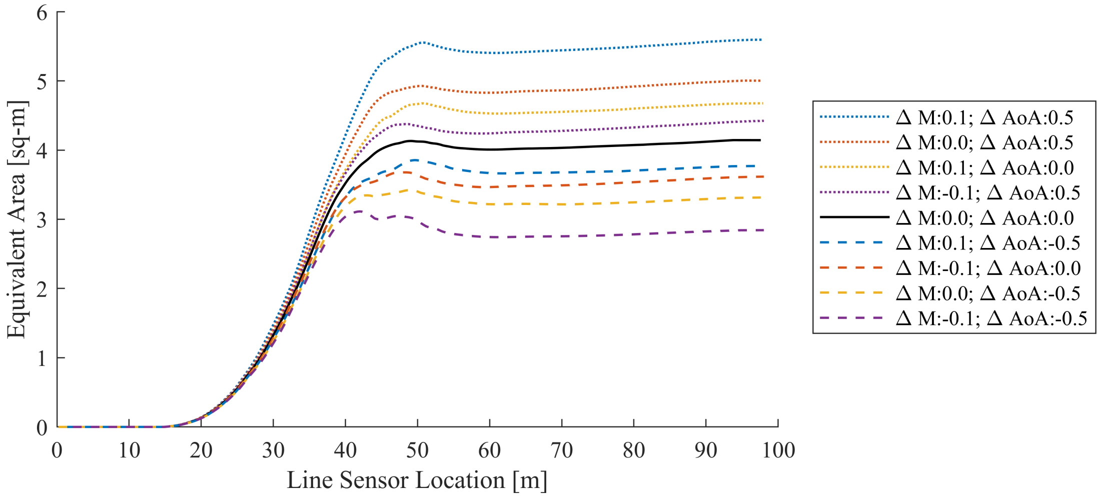


Figure 8.3: Variations of the equivalent area distribution for the NASA 25D aircraft for various local Mach number and angle of attack.

A total of 289 equivalent area distributions are determined via CFD for a gridded combination of 17 Mach variations between ± 0.1 and 17 angle-of-attack variations between ± 0.5°. Of these 289 equivalent area distributions, 240 are used for training and 49 (as a 7x7 internal grid) are used for validation purposes as illustrated by Fig. 8.4.

Before any model can be trained, the data must be preprocessed. First, each equivalent area distribution should be described by a uniform line sensor location spacing. Although we visualize the equivalent area distribution as a curve, the output from CFD tools is a set of line sensor loca-

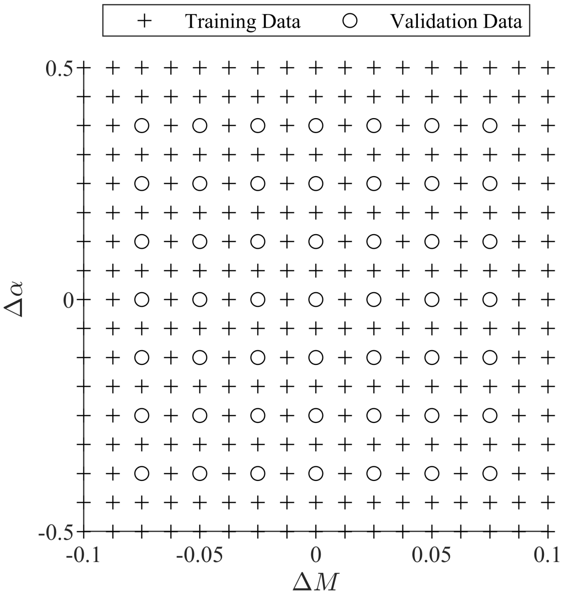
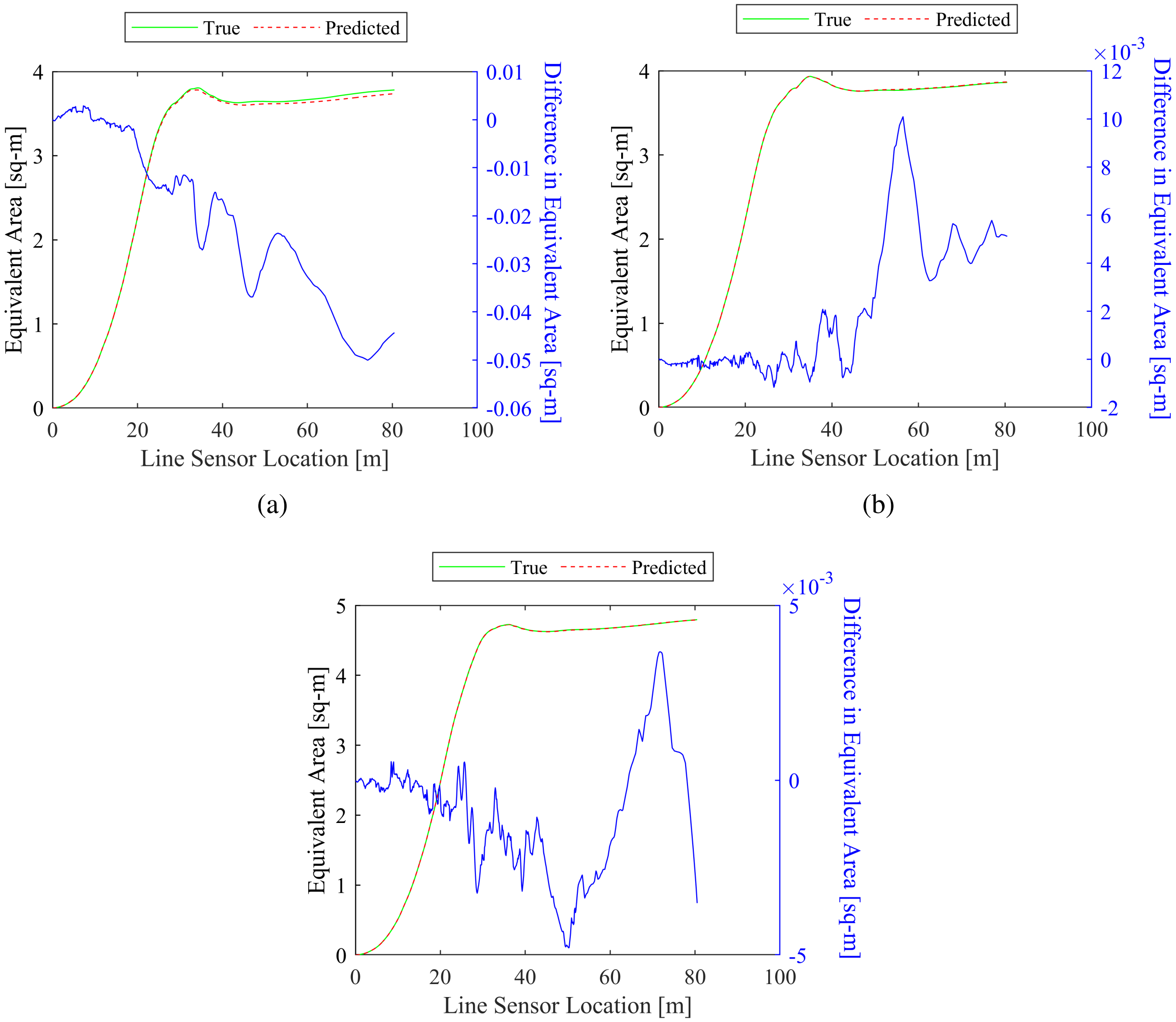


Figure 8.4: Mach and angle of attack combinations used for training and validation.

tion and equivalent area pairs. Although all CFD analysis is done in similarly for each parameter variation, these pairs are not consistently at the same line sensor locations. Therefore, a uniform spacing of 700 line sensor locations from 14.5 m to 95 m is defined and the corresponding equivalent area is linearly interpolated from the CFD data for each M-α combination. For all cases, the equivalent area at line sensor locations below 14.5 m is zero so these locations need not be considered in the model. Furthermore, not all training equivalent area distributions have values calculated from CFD beyond 95 m. To avoid any extrapolation, line sensor locations beyond 95 m are not considered. Finally, the Mach and angle-of-attack variations are normalized on a 0-1 scale for training purposes.

Once this preprocessing is complete, a Kriging model can be trained that accepts the normalized Mach and angle-of-attack variations as inputs to output 700 equivalent area values corresponding to the 700 line sensor locations. An exponential correlation function is used along with a second order polynomial regression function [63]. Figure 8.5 shows a few equivalent area distribution predictions from the resulting model. Figure 8.5a shows the least accurate predicted equivalent area distribution and Figs. 8.5b and 8.5c show some profiles that are predicted more accurately.



(a)

(b)

(c)

Figure 8.5: Equivalent area distribution predictions compared to the distribution calculated via CFD for (a) $\Delta M = -0.075$, $\Delta\alpha = 0.000$, (b) $\Delta M = 0.075$, $\Delta\alpha = -0.375$, and (c) $\Delta M = 0.000$, $\Delta\alpha = 0.375$.

For this application, testing the PLdB of the predicted equivalent area distributions is another important metric to assess model accuracy. Therefore, each of the predicted and actual equivalent area distributions are evaluated with the NASA codes sBOOM (governed by the augmented

Burgers' equation [107]) and LCASB [109] to calculate the PLdB for standard day atmospheric conditions. Figure 8.6 graphically presents the resulting PLdB error for each validation case along with a histogram of the errors. While there is room for model improvement, the accuracy is sufficient for optimization purposes since morphing of the aircraft will be represented by changes to the beginning of the distribution (from 0-35 m) and most of the model prediction error is located at the end of the signature (>35 m).

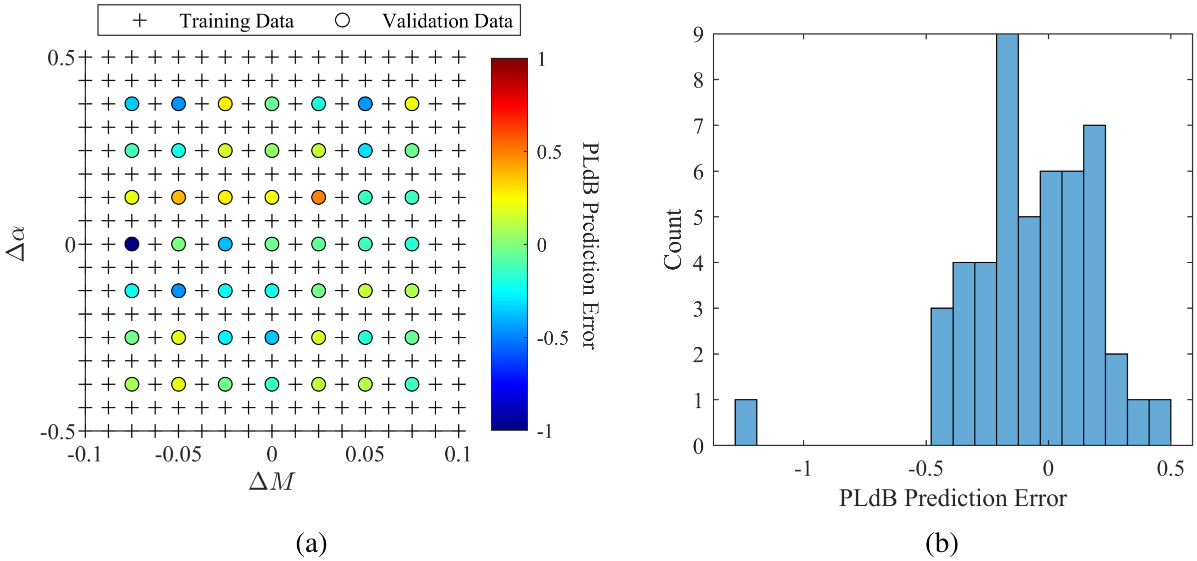


Figure 8.6: PLdB prediction error of validation data presented by (a) a heatmap and (b) histogram.

8.2.2 Problem Formulation

The parametric problem considered allows for three deformations to the baseline (i.e., undeformed) equivalent area via superposition of the description in Fig. 8.2. Therefore, 12 design variables are varied to minimize the PLdB and minimize change in the equivalent area considering all variations of Mach and angle of attack within a feasible range. The mathematical formulation is:

$$
\begin{aligned}
\mathbf{J}^*(\Delta M, \Delta\alpha) = \min_{L_d^i, h^i, l_f^i, l_a^i} \; & [J_1(L_d^i, h^i, l_f^i, l_a^i; \Delta M, \Delta\alpha), J_2(L_d^i, h^i, l_f^i, l_a^i; \Delta M, \Delta\alpha)] \\
\text{Subject to:} \\
& l_f^i + l_a^i \leq 3.5\,\mathrm{m} \\
& L_d^i - l_f^i \geq 0\,\mathrm{m} \\
& L_d^i + l_a^i \leq 35\,\mathrm{m} \\
& \begin{bmatrix} 5\,\mathrm{m} \\ -0.05\,\mathrm{m}^2 \\ 0.125\,\mathrm{m} \\ 0.125\,\mathrm{m} \end{bmatrix} \leq \begin{bmatrix} L_d^i \\ h^i \\ l_f^i \\ l_a^i \end{bmatrix} \leq \begin{bmatrix} 35\,\mathrm{m} \\ 0.05\,\mathrm{m}^2 \\ 3.375\,\mathrm{m} \\ 3.375\,\mathrm{m} \end{bmatrix} \\
& \forall\; i = [1, 2, 3] \\
& -0.075 \leq \Delta M \leq 0.075 \\
& -0.375^\circ \leq \Delta\alpha \leq 0.375^\circ \,,
\end{aligned}
\tag{8.2}
$$

where J_1 represents the PLdB calculated via sBOOM and LCASB and J_2 represents the total change in the equivalent area distribution (i.e., the equivalent "volume" change) via:

$$
J_2(L_d^i, h^i, l_f^i, l_a^i; \Delta M, \Delta\alpha) = \int_x |\Delta A_{E,r}(x, L_d^i, h^i, l_f^i, l_a^i; \Delta M, \Delta\alpha)| dx \;\forall\; i = [1, 2, 3]\,, \tag{8.3}
$$

where the notation $|\cdot|$ represents the absolute value of the argument. Furthermore, the optimization variables are those controlling the deformations illustrated in Fig. 8.2. L_d^i denotes the longitudinal location of the i^{th} deformation for each of the four total deformations. Similarly, h^i, l_f^i, and l_a^i represent the magnitude, fore length, and aft length of the i^{th} deformation. Four variables defining four total deformations adds up to 12 total optimization variables. Each deformation is constrained to be located in the region of the aircraft with longitudinal location L_d between 0.125 and 35 meters from the start of the nearfield line sensor (cf. Fig. 8.1). Each deformation is also constrained such that the absolute change in equivalent area by a single deformation does not exceed 0.05 m^2 and such that the total deformation length (including the fore length l_f and aft length l_a) is less than or equal to 3.5 m. The deformation location constraint is inspired by prior studies [115], and the deformation geometry is constrained via recommendations from an actuator expert [120].

8.3 Results and Discussion

Both MO-EPO and P3GA are used to solve the problem defined in Eqn. (8.2). MO-EPO is implemented with the user options listed in Table 8.1 and reference space defined in Table 8.3. Additionally, 300 points are carried from one iteration to the next when detecting points with high expected improvement (cf. Section 6.2.3). P3GA is implemented with the options listed in Table 8.2 using the same reference space defined in Table 8.3. All optimization is performed on the same machine[2].

Table 8.1: MO-EPO options to solve Eqn. (8.2). Refer to Algorithm 5.

ϕ_{hc}	N_{eval}^{max}	n	tol
90°	3,000	10	0

Table 8.2: P3GA options to solve Eqn. (8.2).

Population Size	Number of Generations	SVDD Width Parameter	Crossover Fraction	Mutation Fraction	Max. SVDD Population
400	300	4	0.30	0.15	4,000

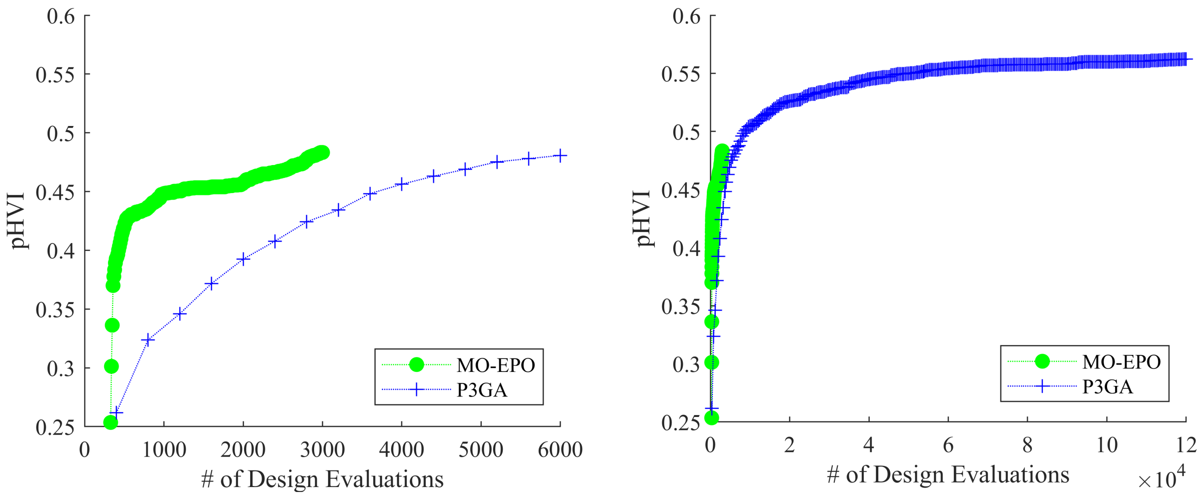
Similar to test problem 2 (cf. Eqn. (A.2)), this problem is multimodal even for a single deformation as illustrated in Ref. [115]. Despite this disadvantage, MO-EPO is capable of quickly exploring desirable regions of the design space in a low number of design evaluations as illustrated

[2] Intel(R) Core(TM) i9-10980XE @ 3.00 GHz CPU with 64 GB of RAM

Table 8.3: Bounds of the reference space used to calculate the pHVI for solutions to Eqn. (8.2). Note that the reference space is comprised of all attributes (parameters and objectives).

	ΔMach [-]	$\Delta\alpha$ [°]	J_1 [dB]	J_2 [m^3]
Lower	-0.075	-0.375	73	0
Upper	0.075	0.375	86	0.10

by Fig. 8.7a. For a total number of design evaluations less than 1,000, the pHVI improvement from MO-EPO outperforms that of P3GA. However, P3GA performs better when larger numbers of design evaluations are allowed (see Fig. 8.7b) due to the error of the response surface model of MO-EPO. For a problem like this, the best course of action is likely to use MO-EPO to quickly search for good designs. Then, when pHVI improvement begins to diminish, the resulting data can be used to initialize P3GA to run for the remainder of the budget.



(a)

(b)

Figure 8.7: pHVI value as a function of the number of designs evaluated for P3GA and MO-EPO for the problem defined in Eqn. (8.2) for (a) 0-6,000 design evaluations and (b) 0-120,000 design evaluations.

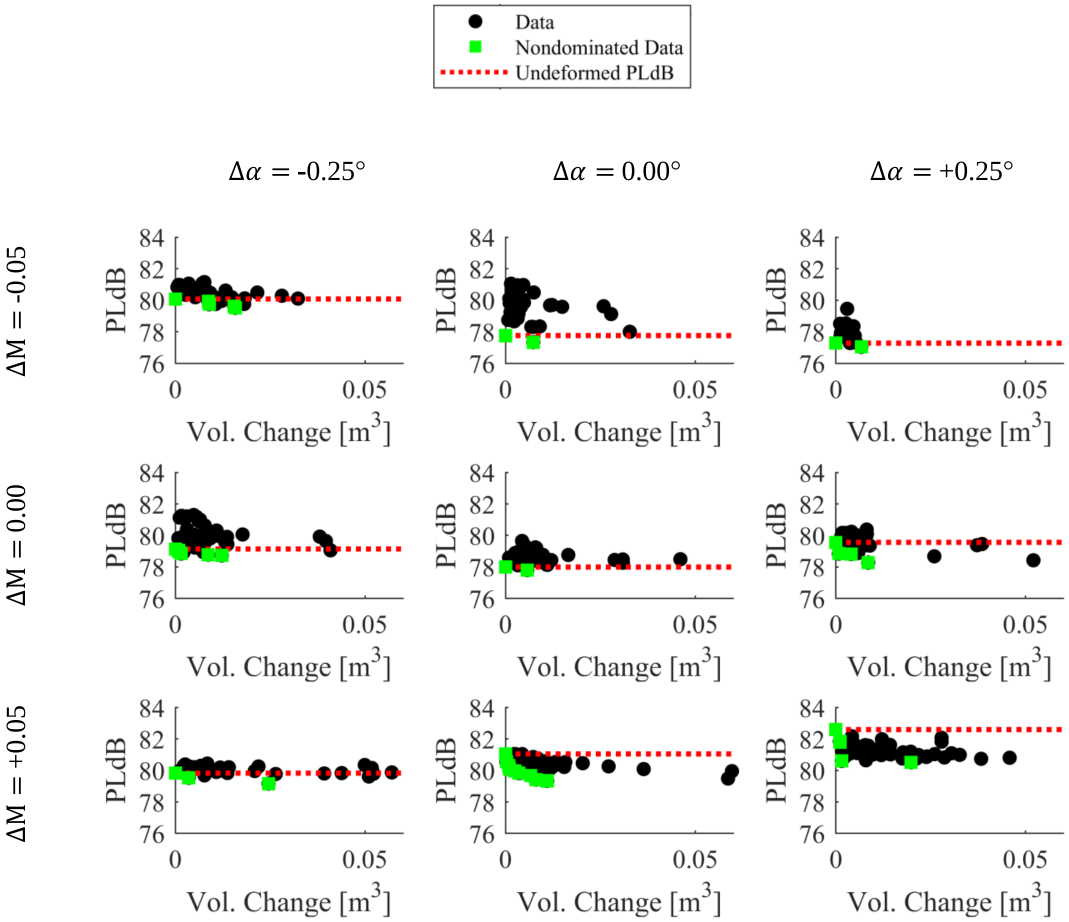


Figure 8.8: MO-EPO results plotted in nine subspaces of the full, four-dimensional parameterized Pareto frontier. Dotted line illustrates the resulting PLdB of the undeformed aircraft at each variation of Mach and angle of attack.

Figure 8.8 illustrates multiple two-dimensional Pareto frontiers pulled from the full four-dimensional parameterized Pareto frontier[3]. In each subplot, the baseline, undeformed PLdB of the aircraft is indicated for the given conditions. Although the optimization algorithm had no information of the baseline PLdB, at least one configuration quieter than the baseline is found within each subspace. In some of these spaces, only a small reduction of PLdB is found (0.1-0.5 dB). However,

[3] To plot scattered data in each subspace, data with ΔMach ±0.01 and $\Delta\alpha \pm 0.1^\circ$ of the printed value are projected into the given space.

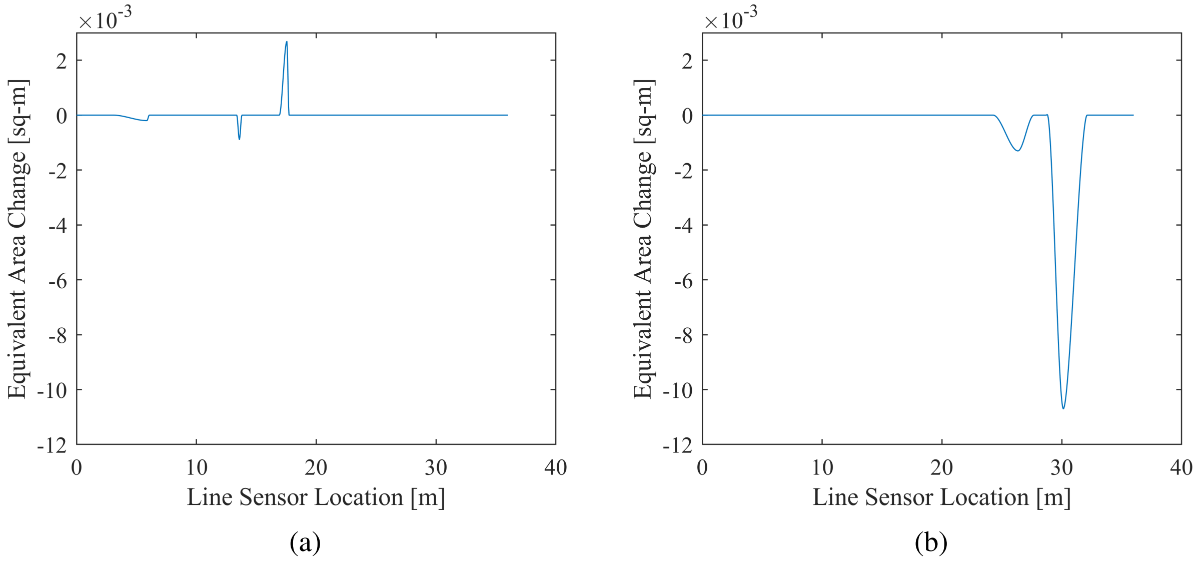


Figure 8.9: Deformations to the equivalent area distribution for PLdB reduction resulting in a perceived sonic boom of (a) 80.61 dB and (b) 80.52 dB.

in cases with increased angle of attack and increased Mach where baseline PLdB is most severe, designs with higher PLdB reduction are found (up to approximately 2 dB of reduction). Looking at the two quietest configurations from the subspace where ΔMach$= +0.05$ and $\Delta\alpha = +0.25°$ (cf. lower right subfigure of Fig. 8.8), an engineer can gain a sense of which part of the aircraft should be investigated for morphing. Figure 8.9 illustrates two separate changes to the equivalent area curve resulting in the PLdB reduction and volume change for this variation of Mach and angle of attack. These deformation locations line up with a previous design exploration study performed on the same aircraft identifying deformation locations of interest near 12 m and 30 m (for the designed Mach and angle of attack - i.e. $\Delta = 0$) [115]. It is encouraging to see deformations near these same locations previously identified even though the operating Mach and angle of attack is different here.

Figure 8.9b shows a configuration resulting in 80.52 dB perceived sonic boom. In contrast, Fig. 8.9a shows a much smaller change in equivalent area resulting in 80.61 dB perceived sonic boom. In general, an engineer might prefer the configuration resulting in a smaller volume change since the resulting PLdB values are so similar. However, consideration of the aircraft must be accounted for to determine where an engineer might prefer to include a morphing mechanism. Another consideration is the uncertainty of the model discussed in Section 8.2.1. When the deformations to the equivalent are very small as in Fig. 8.9a, one should look more at the desired equivalent area rather than the optimal change in equivalent area when solving the inverse problem to determine the 3D morphed shape as in Ref. [92].

9. CONCLUSIONS

9.1 Summary of Contributions

Parametric optimization has been shown to benefit engineering design in addition to other contexts. Specifically, parametric optimization is shown to be valuable for optimal morphing applications where an engineer seeks an adaptive system which remains optimal despite changing external conditions. This is demonstrated on a morphing airfoil concept whereby the optimal morphed shape is found for various flight conditions including angle of attack, airspeed, and altitude to maximize the lift-to-drag ratio.

For design problems with expensive analyses, efficient parametric optimization algorithms are shown to find better solutions with a limited analysis budget. The single objective efficient parametric optimization (EPO) algorithm is presented as a straight-forward extension of the non-parametric EGO algorithm. Despite its promising results, a parametric extension of EGO is limited since some valid assumptions for single-objective, non-parametric optimization do lend themselves to general parametric optimization. In some existing parametric optimization algorithms, parameters are treated similarly to objectives making the extension of a single-objective efficient algorithm flawed from the start.

Therefore, the multi-objective efficient parametric optimization (MO-EPO) algorithm is developed to be more general in handling both parameters and objectives. This method hinges upon the newly developed parametric hypervolume indicator (pHVI). Similar to its multi-objective counterpart, the pHVI measures the space dominated by a given set of data. The pHVI is capable of measuring the size of space dominated in an attribute space defined by both objectives with preference directions and parameters without preference directions. This measure is beneficial to all parametric optimization as it enables the comparison of different solutions to a parametric problem, it enables the in-situ detection of convergence during optimization, and it provides information during optimization that can be used to inform the search process. Although all of these benefits are important, the information gained during optimization is the driving force behind the novel MO-EPO algorithm.

MO-EPO follows the same overarching procedure as EPO and EGO, with some important internal differences. The overall procedure to train a response surface model of expensive functions, find and evaluate a design that is expected to improve the solution and/or model, and repeat is identical to all three of these algorithms. What MO-EPO does differently is finding a design with high expectation of improving the solution and/or model. By using the pHVI metric as the basis for “improvement”, MO-EPO can allow for any number of objectives and parameters. Therefore, MO-EPO seeks to find the parametric solution maximizing the pHVI value which has been proven to be the true solution to any optimization problem.

The novel MO-EPO algorithm is demonstrated on a number of analytical benchmarking problems and two distinct morphing applications. In all benchmarking problems, MO-EPO is shown to be as good as or better than the existing P3GA when the number of design evaluations is limited (based on computational resources, time, and/or monetary funds). This finding is promising for engineering design where analysis can become costly in terms of both money and time.

Furthermore, MO-EPO is implemented to solve a morphing airfoil problem with two formulations (considering one or two objectives) each with three parameters defining external flight conditions. For each formulation, MO-EPO outperforms the existing P3GA by finding a higher quality solution in a short time-frame with a low number of design evaluations. MO-EPO is also used to solve the multi-objective, multi-parametric problem of optimizing a supersonic aircraft via its equivalent area distribution. In this engineering case study, MO-EPO is able to identify regions and magnitudes of morphing interest more quickly than the existing P3GA despite the multimodality of the design space. Through these benchmarking and engineering case studies, the novel MO-EPO algorithm is shown to be a viable method for the efficient optimization of complex engineering systems.

My recommendation is that MO-EPO be used on problems where the analysis budget is limited especially when objective/parameter functions are well behaved. When analysis budgets are not limited (e.g., analysis requires a small fraction of a second and 100,000 design evaluations can be performed in a day), P3GA should likely be used. Alternatively when the analysis budget is limited and functions are not well behaved, a hybrid approach can be taken that first uses MO-EPO to exploit the adaptive sampling of the response surface model capturing large scale trends and then uses P3GA to expend the remaining analysis budget in hopes of finding a higher quality solution. As shown by the benchmarking problems, MO-EPO is capable of finding the true solution for some problems, but more multi-modal and difficult problems may also benefit from the genetic algorithm search of P3GA. A hybrid approach of both MO-EPO and P3GA is a combination worthy of future studies.

9.2 Limitations and Future Work

Although the results presented herein encourage the use of MO-EPO, the algorithm is not without its limitations. First of all, MO-EPO assumes that the the expensive objective/parameter functions are smooth and continuous. Although this assumption will hold for many engineering applications, a designer should be aware of this limitation when this assumption is not valid. The main crutch of the MO-EPO algorithm is the trust put into the response surface model (in terms of its predictions on both response and error). Although a good model will allow MO-EPO to thrive, an invalid model will guide MO-EPO towards poor solutions (like the expression “garbage in, garbage out”). As is the case with many meta-model based optimization algorithms, MO-EPO is limited by the ability to generate trustworthy response surface models of the expensive functions. This limitation is clear when MO-EPO is applied to the second benchmarking problem whereby only minor improvements are found (perhaps somewhat by chance) from one iteration to another at the same rate as P3GA. For the engineering case studies, the model can actually become too confident in its predictions after a number of evaluations. This results in a premature convergence despite the initial success when the number of design evaluations is very low. The remedy for this limitation is likely a more sophisticated response surface modeling technique though some problems may prove to be difficult for any modeling technique. Another solution could be to use MO-EPO to initialize another method such as P3GA.

Another limitation is the consideration of expensive constraint functions. The current implementation of MO-EPO calculates all constraints exactly under the assumption that these constraints are easy to evaluate. This assumption will certainly not always hold. In some cases, a user can implement expensive constraints into objectives through penalty functions, but a more robust method should be considered in future work for the direct consideration of expensive constraint functions.

Another limitation of MO-EPO is the ability to train response surface models in the presence of discrete optimization variables. Response surface models rely on the assumption that variables are continuous. Although the model could be artificially made to be continuous by convert artificial continuous variables to physical discrete variables (e.g., rounding), this can be problematic as fitting a smooth description to a model with discontinuities is not ideal. In Section 4.2.3, the single objective EPO algorithm still outperforms P3GA even though one variable is discrete. However, for problems with many discrete variables, an algorithm such as P3GA might be a better option.

Despite its benefits, the hypercone heuristic (HCH) for calculating the pHVI value is also limited. As previously discussed, the decision of the hypercone angle and reference space are paramount when predicting parametric Pareto dominance (PPD). Although an engineer can gain an intuition for how to set these algorithmic parameters, it is difficult to know what the "best" values are a priori. Although this work presents some illustrations of the effects of these algorithmic parameters to provide a user with some intuition, a more general rule of thumb would be beneficial as an area of future work.

REFERENCES

[1] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Springer*, vol. 3, no. 2, pp. 95–99, 1988.

[2] J. M. Weaver-Rosen, P. B. C. Leal, D. J. Hartl, and R. J. Malak, "Parametric optimization for morphing structures design: application to morphing wings adapting to changing flight conditions," *Structural and Multidisciplinary Optimization*, July 2020.

[3] P. B. Leal, M. A. Savi, and D. J. Hartl, "Aero-structural optimization of shape memory alloy-based wing morphing via a class/shape transformation approach," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 232, pp. 2745–2759, Dec. 2018.

[4] P. Gamboa, J. Vale, F. J. P. Lau, and A. Suleman, "Optimization of a Morphing Wing Based on Coupled Aerodynamic and Structural Constraints," *AIAA Journal*, vol. 47, pp. 2087–2104, Sept. 2009.

[5] E. Galvan, R. J. Malak, D. J. Hartl, and J. W. Baur, "Performance assessment of a multi-objective parametric optimization algorithm with application to a multi-physical engineering system," *Structural and Multidisciplinary Optimization*, vol. 58, pp. 489–509, Aug. 2018.

[6] E. N. Pistikopoulos, V. Dua, N. A. Bozinis, A. Bemporad, and M. Morari, "On-line optimization via off-line parametric optimization tools," *Computers & Chemical Engineering*, vol. 24, no. 2-7, pp. 183–188, 2004.

[7] D. J. Hartl, E. Galvan, R. J. Malak, and J. W. Baur, "Parameterized Design Optimization of a Magnetohydrodynamic Liquid Metal Active Cooling Concept," *Journal of Mechanical Design*, vol. 138, p. 031402, Mar. 2016.

[8] Y.-K. Tsai and R. J. Malak, "A Methodology for Designing a Nonlinear Feedback Controller via Parametric Optimization: State-Parameterized Nonlinear Programming Control," in *ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (Accepted), 2021.

[9] R. J. Malak and C. J. J. Paredis, "Using Parameterized Pareto Sets to Model Design Concepts," *Journal of Mechanical Design*, vol. 132, no. 4, p. 041007, 2010.

[10] E. Galvan and R. J. Malak, "P3ga: An algorithm for technology characterization," *Journal of Mechanical Design*, vol. 137, no. 1, p. 011401, 2015.

[11] J. M. Weaver-Rosen and R. J. Malak, "Efficient Parametric Optimization for Expensive Single Objective Problems," *Journal of Mechanical Design*, vol. 143, p. 031711, Mar. 2021.

[12] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 421–439, Aug. 2006.

[13] A. V. Fiacco, "Sensitivity analysis for nonlinear programming using penalty methods," *Mathematical Programming*, vol. 10, pp. 287–311, Dec. 1976.

[14] E. N. Pistikopoulos and C. for Process Systems Engineering, eds., *Multi-parametric programming: theory, algorithms, and applications*. No. Centre for Process Systems Engineering. Ed. by Efstratios N. Pistikopoulos ... ; Vol. 1 in Process systems engineering, Weinheim: Wiley-VCH, 2007. OCLC: 180943251.

[15] C. Filippi, "An Algorithm for Approximate Multiparametric Linear Programming," *Journal of Optimization Theory and Applications*, vol. 120, pp. 73–95, Jan. 2004.

[16] T. Johansen, "On multi-parametric nonlinear programming and explicit nonlinear model predictive control," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 3, (Las Vegas, NV, USA), pp. 2768–2773, IEEE, 2002.

[17] A. J. Keane and I. I. Voutchkov, "Robust design optimization using surrogate models," *Journal of Computational Design and Engineering*, vol. 7, pp. 44–55, Feb. 2020.

[18] H.-G. Beyer and B. Sendhoff, “Robust optimization – A comprehensive survey,” *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 3190–3218, July 2007.

[19] M. Li and Z. Wang, “Confidence-Driven Design Optimization Using Gaussian Process Metamodeling With Insufficient Data,” *Journal of Mechanical Design*, vol. 140, p. 121405, Dec. 2018.

[20] J. Acevedo and M. Salgueiro, “An Efficient Algorithm for Convex Multiparametric Nonlinear Programming Problems,” *Industrial & Engineering Chemistry Research*, vol. 42, pp. 5883–5890, Nov. 2003.

[21] Y. Wang, H. Seki, S. Ohyama, K. Akamatsu, M. Ogawa, and M. Ohshima, “Optimal grade transition control for polymerization reactors,” *Computers & Chemical Engineering*, vol. 24, pp. 1555–1561, July 2000.

[22] I. A. Fotiou, P. Rostalski, P. A. Parrilo, and M. Morari, “Parametric optimization and optimal control using algebraic geometry methods,” *International Journal of Control*, vol. 79, pp. 1340–1358, Nov. 2006.

[23] S. Avraamidou and E. N. Pistikopoulos, “B-POP: Bi-level parametric optimization toolbox,” *Computers & Chemical Engineering*, vol. 122, pp. 193–202, Mar. 2019.

[24] M. Spiegel and A. Subrahmanyam, “Informed Speculation and Hedging in a Noncompetitive Securities Market,” *The Review of Financial Studies*, vol. 5, no. 2, pp. 307–329, 1992.

[25] M. Kojima and R. Hirabayashi, “Continuous deformation of nonlinear programs,” in *Sensitivity, Stability and Parametric Analysis* (R. W. Cottle, L. C. W. Dixon, B. Korte, M. J. Todd, E. L. Allgower, W. H. Cunningham, J. E. Dennis, B. C. Eaves, R. Fletcher, D. Goldfarb, J.-B. Hiriart-Urruty, M. Iri, R. G. Jeroslow, D. S. Johnson, C. Lemarechal, L. Lovasz, L. McLinden, M. J. D. Powell, W. R. Pulleyblank, A. H. G. Rinnooy Kan, K. Ritter, R. W. H. Sargent, D. F. Shanno, L. E. Trotter, H. Tuy, R. J. B. Wets, E. M. L. Beale, G. B. Dantzig, L. V. Kantorovich, T. C. Koopmans, A. W. Tucker, P. Wolfe, and A. V. Fiacco, eds.), vol. 21, pp. 150–198, Berlin, Heidelberg: Springer Berlin Heidelberg, 1984.

[26] J. Guddat, F. Guerra Vazquez, and H. T. Jongen, *Parametric optimization: singularities, pathfollowing, and jumps*. Stuttgart : Chichester, England ; New York: B.G. Teubner ; Wiley, 1990.

[27] B. N. Lundberg and A. B. Poore, “Numerical Continuation and Singularity Detection Methods for Parametric Nonlinear Programming,” *SIAM Journal on Optimization*, vol. 3, pp. 134–154, Feb. 1993.

[28] R. Oberdieck, N. A. Diangelakis, M. M. Papathanasiou, I. Nascu, and E. N. Pistikopoulos, “POP – Parametric Optimization Toolbox,” *Industrial & Engineering Chemistry Research*, vol. 55, pp. 8979–8991, Aug. 2016.

[29] B. N. Lundberg and A. B. Poore, “Variable Order Adams-Bashforth Predictors with an Error-Stepsize Control for Continuation Methods,” *SIAM Journal on Scientific and Statistical Computing*, vol. 12, pp. 695–723, May 1991.

[30] E. T. Hale, *Numerical Methods for d-Parametric Nonlinear Programming with Chemical Process Control and Optimization Applications*. PhD Thesis, Chemical Engineering Department, University of Texas, Austin, TX, 2005.

[31] M. Brodzik, “The computation of simplicial approximations of implicitly defined p-dimensional manifolds,” *Computers & Mathematics with Applications*, vol. 36, pp. 93–113, Sept. 1998.

[32] M. E. Henderson, “Multiple Parameter Continuation: Computing Implicitly Defined K-Manifolds,” *International Journal of Bifurcation and Chaos*, vol. 12, pp. 451–476, Mar. 2002.

[33] M. E. Henderson, “Higher-Dimensional Continuation,” in *Numerical Continuation Methods for Dynamical Systems: Path following and boundary value problems* (B. Krauskopf, H. M. Osinga, and J. Galán-Vioque, eds.), pp. 77–115, Dordrecht: Springer Netherlands, 2007.

[34] T. A. Johansen, “Approximate explicit receding horizon control of constrained nonlinear systems,” *Automatica*, vol. 40, pp. 293–300, Feb. 2004.

[35] V. Dua, K. Papalexandri, and E. Pistikopoulos, “Global Optimization Issues in Multiparametric Continuous and Mixed-Integer Optimization Problems,” *Journal of Global Optimization*, vol. 30, pp. 59–89, Sept. 2004.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, Apr. 2002.

[37] MATLAB, “{MATLAB} {User} {Manual} {R}2017b.”

[38] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[39] H. M. Gutmann, “A Radial Basis Function Method for Global Optimization,” *Journal of Global Optimization*, vol. 19, pp. 201–227, 2001.

[40] D. Wang, Z. Wu, Y. Fei, and W. Zhang, “Structural design employing a sequential approximation optimization approach,” *Computers & Structures*, vol. 134, pp. 75–87, Apr. 2014.

[41] J. Knowles, “ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 50–66, Feb. 2006.

[42] A. J. Keane, “Statistical Improvement Criteria for Use in Multiobjective Design Optimization,” *AIAA Journal*, vol. 44, pp. 879–891, Apr. 2006.

[43] D. Hernández-Lobato, J. M. Hernández-Lobato, A. Shah, and R. P. Adams, “Predictive Entropy Search for Multi-objective Bayesian Optimization,” p. 10, 2014.

[44] G. Sun, Y. Tian, R. Wang, J. Fang, and Q. Li, “Parallelized multiobjective efficient global optimization algorithm and its applications,” *Structural and Multidisciplinary Optimization*, vol. 61, pp. 763–786, Feb. 2020.

[45] D. Khatamsaz, L. Peddareddygari, S. Friedman, and D. L. Allaire, "Efficient Multi-Information Source Multiobjective Bayesian Optimization," in *AIAA Scitech 2020 Forum*, (Orlando, FL), American Institute of Aeronautics and Astronautics, Jan. 2020.

[46] M. T. M. Emmerich, A. H. Deutz, and J. W. Klinkenberg, "Hypervolume-based expected improvement: Monotonicity properties and exact computation," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, (New Orleans, LA, USA), pp. 2147–2154, IEEE, June 2011.

[47] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms — A comparative case study," in *Parallel Problem Solving from Nature — PPSN V* (G. Goos, J. Hartmanis, J. van Leeuwen, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, eds.), vol. 1498, pp. 292–301, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. Series Title: Lecture Notes in Computer Science.

[48] D. A. Van Velhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Dissertation, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 1999.

[49] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Dissertation, Swiss Federal Institute of Technology Zurich, 1999.

[50] M. Laumanns, G. Rudolph, and H.-P. Schwefel, "Approximating the Pareto Set: Concepts, Diversity Issues, and Performance Assessment," p. 13, 1999.

[51] M. Fleischer, "The Measure of Pareto Optima: Applications to Multi-Objective Metaheuristics," Technical Report ISR TR 2002-32, 2003.

[52] E. Zitzler, D. Brockhoff, and L. Thiele, "The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration," in *Evolutionary Multi-Criterion Optimization* (S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, eds.), vol. 4403, pp. 862–876, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[53] D. J. Maglieri, P. J. Bobbitt, K. J. Plotkin, K. P. Shepherd, P. G. Coen, and D. M. Richwine, "Sonic boom: Six decades of research," Tech. Rep. NASA SP-2014-622, NASA Langley Research Center, Hampton, Virginia, United States, 2014.

[54] C. Nam, A. Chattopadhyay, and Y. Kim, "Application of shape memory alloy (SMA) spars for aircraft maneuver enhancement," in *Smart structures and materials 2002: smart structures and integrated systems*, vol. 4701, pp. 226–236, International Society for Optics and Photonics, 2002.

[55] E. Tandis and E. Assareh, "Inverse design of airfoils via an intelligent hybrid optimization technique," *Engineering with Computers*, vol. 33, pp. 361–374, July 2017.

[56] K. Lane and D. Marshall, "A Surface Parameterization Method for Airfoil Optimization and High Lift 2D Geometries Utilizing the CST Methodology," in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, (Orlando, Florida), American Institute of Aeronautics and Astronautics, Jan. 2009.

[57] R. P. Liem, J. R. Martins, and G. K. Kenway, "Expected drag minimization for aerodynamic design optimization based on aircraft operational data," *Aerospace Science and Technology*, vol. 63, pp. 344–362, Apr. 2017.

[58] A. M. Morris, C. B. Allen, and T. C. S. Rendall, "High-fidelity aerodynamic shape optimization of modern transport wing using efficient hierarchical parameterization," *International Journal for Numerical Methods in Fluids*, vol. 63, pp. 297–312, May 2010.

[59] Y. Liu, C. Yang, and X. Song, "An airfoil parameterization method for the representation and optimization of wind turbine special airfoil," *Journal of Thermal Science*, vol. 24, pp. 99–108, Apr. 2015.

[60] R. M. Hicks and P. A. Henne, "Wing Design by Numerical Optimization," *Journal of Aircraft*, vol. 15, pp. 407–412, July 1978.

[61] P. B. C. Leal and D. J. Hartl, "Structurally Consistent Class/Shape Transformation Equations for Morphing Airfoils," *Journal of Aircraft*, vol. 56, pp. 505–516, Mar. 2019.

[62] W. Van Beers and J. Kleijnen, “Kriging Interpolation in Simulation: A Survey,” in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, vol. 1, (Washington, D.C.), pp. 107–115, IEEE, 2004.

[63] S. N. Lophaven, H. B. Nielsen, and J. Sondergaard, “Dace - a Matlab Kriging Toolox, Version 2.0,” 2002.

[64] P. B. Leal, R. Petterson, and D. J. Hartl, “Design optimization toward a shape memory alloy-based bio-inspired morphing wing,” in *25th AIAA/AHS Adaptive Structures Conference*, (Grapevine, Texas), American Institute of Aeronautics and Astronautics, Jan. 2017.

[65] B. Kulfan, “Recent Extensions and Applications of the "CST" Universal Parametric Geometry Representation Method,” in *7th AIAA ATIO Conf, 2nd CEIAT Int’l Conf on Innov and Integr in Aero Sciences,17th LTA Systems Tech Conf; followed by 2nd TEOS Forum*, (Belfast, Northern Ireland), American Institute of Aeronautics and Astronautics, Sept. 2007.

[66] M. Drela, *XFOIL 6.9 User Primers*. MIT AERO and Astro, Harold Youngren, Aerocraft, Inc., 2001.

[67] M. S. Selig, ed., *Summary of low speed airfoil data*. Virginia Beach, Va: SoarTech Publications, 1995.

[68] R. J. Malak and C. J. J. Paredis, “Using Support Vector Machines to Formalize the Valid Input Domain of Predictive Models in Systems Design Problems,” *Journal of Mechanical Design*, vol. 132, no. 10, p. 101001, 2010.

[69] D. M. Tax and R. P. Duin, “Support vector domain description,” *Pattern Recognition Letters*, vol. 20, pp. 1191–1199, Nov. 1999.

[70] D. Arthur and S. Vassilvitskii, “k-means++: The Advantages of Careful Seeding,” p. 11.

[71] I. H. Abbott and A. E. VonDoenhoff, *Theory of wing sections: including a summery of airfoil data*. New York, NY: Dover Publ, unabr. and corr. republ., [nachdr.] ed., 2010. OCLC: 837654605.

[72] C. Tropea, H. Bleckmann, and D. Forschungsgemeinschaft, eds., *Nature-inspired fluid mechanics: results of the DFG priority programme 1207 "nature-inspired fluid mechanics" 2006 - 2012*. No. 119 in Notes on numerical fluid mechanics and multidisciplinary design, Heidelberg: Springer, 2012. OCLC: 845004823.

[73] E. N. Jacobs and R. M. Pinkerton, “Tests in the Variable-Density Wind Tunnel of Related Airfoils Having the Maximum Camber Unusually Far Forward,” Tech. Rep. NACA-TR-537, U.S. Environmental Protection Agency available through the National Technical Information Service, Springfield, Va., Langley Field, VA, United States, 1936.

[74] A. M. Pankonien, L. Gamble, C. Faria, and D. J. Inman, “Synergistic Smart Morphing AIleron: Capabilites Identification,” in *24th AIAA/AHS Adaptive Structures Conference*, (San Diego, California, USA), American Institute of Aeronautics and Astronautics, Jan. 2016.

[75] H. Liu, Y.-S. Ong, and J. Cai, “A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design,” *Structural and Multidisciplinary Optimization*, vol. 57, pp. 393–416, Jan. 2018.

[76] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable Test Problems for Evolutionary Multiobjective Optimization,” in *Evolutionary Multiobjective Optimization* (A. Abraham, L. Jain, and R. Goldberg, eds.), pp. 105–145, London: Springer-Verlag, 2005.

[77] L. While, L. Bradstreet, and L. Barone, “A Fast Way of Calculating Exact Hypervolumes,” *IEEE Transactions on Evolutionary Computation*, vol. 16, pp. 86–95, Feb. 2012.

[78] A. Jaszkiewicz, “Improved quick hypervolume algorithm,” *Computers & Operations Research*, vol. 90, pp. 72–83, Feb. 2018.

[79] J. Bader, K. Deb, and E. Zitzler, “Faster Hypervolume-Based Search Using Monte Carlo Sampling,” in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems* (M. Ehrgott, B. Naujoks, T. Stewart, and J. Wallenius, eds.), vol. 634 of *Lecture*

Notes in Economics and Mathematical Systems, pp. 313–326, Berlin, Heidelberg: Springer, 2010.

[80] S. Timoshenko and J. M. Gere, "Elastic Buckling of Bars and Frames," in *Theory of elastic stability*, pp. 46–162, Mineola, N.Y: Dover Publications, 2nd ed., dover ed ed., 2009. OCLC: ocn294885242.

[81] S. Gao, W. Zhao, H. Lin, F. Yang, and X. Chen, "Feature suppression based CAD mesh model simplification," *Computer-Aided Design*, vol. 42, pp. 1178–1188, Dec. 2010.

[82] J. Knowles and D. Corne, "On metrics for comparing nondominated sets," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 1, (Honolulu, HI, USA), pp. 711–716, IEEE, 2002.

[83] J. Kristensen, Y. Ling, I. Asher, and L. Wang, "Expected-Improvement-Based Methods for Adaptive Sampling in Multi-Objective Optimization Problems," in *Volume 2B: 42nd Design Automation Conference*, (Charlotte, North Carolina, USA), p. V02BT03A024, American Society of Mechanical Engineers, Aug. 2016.

[84] G. Zhao, R. Arroyave, and X. Qian, "Fast Exact Computation of Expected HyperVolume Improvement," *arXiv:1812.07692 [cs, stat]*, Jan. 2019. arXiv: 1812.07692.

[85] I. Couckuyt, D. Deschrijver, and T. Dhaene, "Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization," *Journal of Global Optimization*, vol. 60, pp. 575–594, Nov. 2014.

[86] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, pp. 1653–1669, Sept. 2007.

[87] Y. Wang, S. Limmer, M. Olhofer, M. Emmerich, and T. Baeck, "Automatic Preference Based Multi-objective Evolutionary Algorithm on Vehicle Fleet Maintenance Scheduling Optimization," *arXiv:2101.09556 [cs]*, Jan. 2021. arXiv: 2101.09556.

[88] M. Emmerich, A. Deutz, J. Kruisselbrink, and P. K. Shukla, “Cone-Based Hypervolume Indicators: Construction, Properties, and Efficient Computation,” in *Evolutionary Multi-Criterion Optimization* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, eds.), vol. 7811, pp. 111–127, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Series Title: Lecture Notes in Computer Science.

[89] G. H. Golub and J. H. Welsch, “Calculation of Gauss Quadrature Rules,” *Mathematics of computation*, vol. 23, no. 106, pp. 221–230, 1969.

[90] A. Ralston and P. Rabinowitz, *A first course in numerical analysis*. Courier Corporation, 2001.

[91] G. Von Winckel, “Legendre-gauss quadrature weights and nodes,” *Matlab function lgwt. URL: http://www. mathworks. com/matlabcentral/fileexchange/4540 Wessel P, Smith WHF (1998), New, improved version of the Generic Mapping Tools released, EOS Trans. AGU*, vol. 79, p. 579, 2004.

[92] J. M. Weaver-Rosen, F. L. Carpenter, P. G. Cizmas, R. J. Malak, T. A. Abraham, D. F. Hunsaker, and D. S. Lazzara, “Computational Design Methodology of Adaptive Outer Mold Line for Robust Low En-Route Noise of a Supersonic Aircraft,” in *AIAA Scitech 2021 Forum*, (VIRTUAL EVENT), American Institute of Aeronautics and Astronautics, Jan. 2021.

[93] D. G. Baize, “1995 NASA High-Speed Research Program Sonic Boom Workshop–Volume I,” NASA CP-3335, July 1996.

[94] D. G. Baize, “1995 NASA High-Speed Research Program Sonic Boom Workshop–Volume II–Configuration Design, Analysis and Testing,” NASA CP-1999-209520, Dec. 1999.

[95] T. E. Magee, P. A. Wilcox, S. R. Fugal, K. E. Acheson, E. E. Adamson, A. L. Bidwell, and S. G. Shaw, “System-Level Experimental Validations for Supersonic Commercial Transport

Aircraft Entering Service in the 2018-2020 Time Period: Phase I Final Report," Technical Report NASA/CR-2013-217797, NASA Langley Research Center, 2013.

[96] T. E. Magee, S. R. Fugal, L. E. Fink, E. E. Adamson, and S. G. Shaw, "System-Level Experimental Validations for Supersonic Commercial Transport Aircraft Entering Service in the 2018-2020 Time Period: Phase II Final Report," Technical Report NASA/CR-2015-218983, NASA Langley Research Center, 2015.

[97] T. E. Magee, S. R. Fugal, L. Fink, and S. G. Shaw, "Boeing N+2 Supersonic Experimental Validation Phase II program," AIAA 2014-2137, June 2014.

[98] T. E. Magee, S. G. Shaw, and S. R. Fugal, "Experimental Validations of a Low-Boom Aircraft Design," AIAA 2013-0646, Jan. 2013.

[99] J. Banke, "NASA's Experimental Supersonic Aircraft Now Known as X-59 QueSST," NASA, July 2018.

[100] J. Pawlowski, D. Graham, C. Boccadoro, P. Coen, and D. Maglieri, "Origins and Overview of the Shaped Sonic Boom Demonstration Program," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Jan. 2005.

[101] Y. Makino and I. Kroo, "Robust Objective Functions for Sonic-Boom Minimization," vol. 43, Journal of Aircraft, Oct. 2006. Issue: 5.

[102] R. Carmichael and L. Erickson, "PAN AIR–A higher order panel method for predicting subsonic or supersonic linear potential flows about arbitrary configurations," in *14th Fluid and Plasma Dynamics Conference*, (Palo Alto, California), June 1981.

[103] M. Nemec and M. Aftosmis, "Parallel Adjoint Framework for Aerodynamic Shape Optimization of Component-Based Geometry," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Jan. 2011.

[104] M. A. Park, "Low Boom Configuration Analysis with FUN3D Adjoint Simulation Framework," in *29th AIAA Applied Aerodynamics Conference*, (Honolulu, Hawaii), AIAA 2011-3337, June 2011.

[105] H. Shen and D. S. Lazzara, "A Space Marching Method for Sonic Boom Near Field Predictions," in *54th AIAA Aerospace Sciences Meeting*, (San Diego, California), AIAA 2016-2037, Jan. 2016.

[106] R. Seebass, "Sonic Boom Theory," vol. 6, Journal of Aircraft, 1969. Issue: 3.

[107] S. K. Rallabhandi, "Advanced Sonic Boom Prediction Using the Augmented Burgers Equation," *Journal of Aircraft*, vol. 48, pp. 1245–1253, July 2011.

[108] S. S. Stevens, "Perceived Level of Noise by Mark VII and Decibels (E)," *The Journal of the Acoustical Society of America*, vol. 51, pp. 575–601, Feb. 1972.

[109] K. P. Shepard and B. M. Sullivan, "A Loudness Calculation Procedure Applied to Shaped Sonic Booms," Technical Paper 3141, NASA, Nov. 1991.

[110] M. Aftosmis, M. Nemec, and S. Cliff, "Adjoint-based low-boom design with Cart3D (Invited)," in *29th AIAA Applied Aerodynamics Conference*, (Honolulu, Hawaii), American Institute of Aeronautics and Astronautics, June 2011.

[111] M. Wintzer, M. Nemec, and M. Aftosmis, "Adjoint-Based Adaptive Mesh Refinement for Sonic Boom Prediction," in *26th AIAA Applied Aerodynamics Conference*, (Honolulu, Hawaii), American Institute of Aeronautics and Astronautics, Aug. 2008.

[112] S. K. Rallabhandi, "Uncertainty Analysis and Robust Design of Low-Boom Concepts using Atmospheric Adjoints," in *33rd AIAA Applied Aerodynamics Conference*, (Dallas, TX), American Institute of Aeronautics and Astronautics, June 2015.

[113] F. Palacios, J. Alonso, M. Colonno, J. Hicken, and T. Lukaczyk, "Adjoint-based method for supersonic aircraft design using equivalent area distribution," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, (Nashville, Tennessee), American Institute of Aeronautics and Astronautics, Jan. 2012.

[114] T. Lukaczyk, F. Palacios, and J. Alonso, “Response Surface Methodologies for Low-Boom Supersonic Aircraft Design Using Equivalent Area Distributions,” in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Indianapolis, Indiana), American Institute of Aeronautics and Astronautics, Sept. 2012.

[115] T. Abraham, D. F. Hunsaker, J. M. Weaver-Rosen, and R. J. Malak, “Identifying Optimal Equivalent Area Changes to Reduce Sonic Boom Loudness,” in *AIAA Scitech 2020 Forum*, (Orlando, FL), American Institute of Aeronautics and Astronautics, Jan. 2020.

[116] I. E. Garrick and D. J. Maglieri, “A summary of results on sonic-boom pressure-signature variations associated with atmospheric conditions,” NASA TN D-4588, 1968.

[117] L. D. Robinson, “Sonic Boom Propagation through an inhomogeneous, windy atmosphere,” Ph. D Thesis, The University of Texas at Austin, 1991.

[118] D. S. Lazzara, T. Magee, H. Shen, and J. H. Mabe, “Off-Design Sonic Boom Performance for Low-Boom Aircraft,” in *AIAA Scitech 2019 Forum*, (San Diego, California), American Institute of Aeronautics and Astronautics, Jan. 2019.

[119] W. Li and S. Rallabhandi, “Inverse Design of Low-Boom Supersonic Concepts Using Reversed Equivalent-Area Targets,” *Journal of Aircraft*, vol. 51, pp. 29–36, Jan. 2014.

[120] J. Mabe. 2020. Private Communication.

[121] P. B. Leal, H. Stroud, E. Sheahan, M. Cabral, and D. J. Hartl, “Skin-based camber morphing utilizing shape memory alloy composite actuators in a wind tunnel environment,” in *2018 AIAA/AHS Adaptive Structures Conference*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Jan. 2018.

APPENDIX A

TEST PROBLEM DEFINITIONS

A.1 Test Problem #1

The variant of Test Problem C from Galvan et al. [5] used in this work is given by Eqn. (A.1) where N_{var}, N_o, and N_p denote the number of optimization variables, objectives, and parameters, respectively.

$$
\begin{aligned}
&\mathbf{J}^*(\boldsymbol{\theta}) = \min_{\boldsymbol{x}} \mathbf{J}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) \;\forall\; \boldsymbol{\theta} \in [\boldsymbol{\theta}_{lb}, \boldsymbol{\theta}_{ub}] \\
&\text{subject to:} \\
&\quad 0 \leq x_m \leq 1 \,\forall\, m = 1, 2, ..., N_{var} \\
&\text{where:} \\
&\mathbf{J}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = [J_1(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})), ..., J_{N_o}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x}))] \\
&\boldsymbol{\theta}(\boldsymbol{x}) = [\theta_1(\boldsymbol{x}), ..., \theta_{N_p}(\boldsymbol{x})] \\
&\theta_i(\boldsymbol{x}) = \frac{1}{2}(1 + \sin(20 x_i)) \forall i = 1, 2, ..., N_p \\
&J_j(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = \frac{1}{2}(1 + \sin(20 x_{(N_p + j)})) \forall j = 1, 2, ..., N_o - 1 \\
&J_{N_o}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = \left(1 + \frac{9}{6 - N_o - N_p} \sum_{k = N_o + N_p}^{N_{var}} x_k\right) \\
&\qquad\qquad \times \left(3(N_o + N_p) - \sum_{i=1}^{N_p} (2\theta_i + \sin(3\pi\theta_i)) \right. \\
&\qquad\qquad\quad \left. - \sum_{j=1}^{N_o - 1} (2J_j + \sin(3\pi J_j))\right)
\end{aligned}
\tag{A.1}
$$

A.2 Test Problem #2

Similarly, Eqn. (A.2) gives the variant of Test Problem B [5] used in this work.

$$
\begin{aligned}
&\mathbf{J}^*(\boldsymbol{\theta}) = \min_{\boldsymbol{x}} \ \mathbf{J}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) \ \forall \ \boldsymbol{\theta} \in [\boldsymbol{\theta}_{lb}, \boldsymbol{\theta}_{ub}] \\
&\text{subject to:} \\
&\quad 0 \leq x_m \leq 1 \, \forall \, m = 1, 2, ..., N_{var} \\
&\text{where:} \\
&\mathbf{J}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = [J_1(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})), ..., J_{N_o}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x}))] \\
&\boldsymbol{\theta}(\boldsymbol{x}) = [\theta_1(\boldsymbol{x}), ..., \theta_{N_p}(\boldsymbol{x})] \\
&\theta_i(\boldsymbol{x}) = \frac{1}{2}(1 + \sin(2\pi \cos(5\pi x_i))) \forall i = 1, 2, ..., N_p \\
&J_j(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = \frac{1}{2}(1 + \sin(2\pi \cos(5\pi x_{(N_p+j)}))) \forall j = 1, 2, ..., N_o - 1 \\
&J_{N_o}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = \left(1 + N_o + N_p + \sum_{k=0}^{N_{var}-N_o-N_p} \left(\cos((k+1)\pi x_{k+N_o+N_p}^{1/50})\right)\right) \\
&\qquad\qquad \times \left(N_o + N_p + \sum_{i=1}^{N_p} \left(\theta_i^2 + \cos((1-\theta_i)\pi)^2\right)\right. \\
&\qquad\qquad\quad \left. + \sum_{j=1}^{N_o-1} \left(J_j^2 + \cos((1-J_j)\pi)^2\right)\right)
\end{aligned}
\tag{A.2}
$$

A.3 Test Problem #3

Equation (A.3) gives the parametric variant of DTLZ1 [76] used in this work.

$$
\begin{aligned}
&\mathbf{J}^*(\boldsymbol{\theta}) = \min_{\boldsymbol{x}} \; \mathbf{J}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) \;\; \forall \;\; \boldsymbol{\theta} \in [\boldsymbol{\theta}_{lb}, \boldsymbol{\theta}_{ub}] \\
&\text{subject to:} \\
&\quad 0 \leq x_m \leq 1 \, \forall \, m = 1, 2, ..., N_{var} \\
&\text{where:} \\
&\mathbf{J}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = [J_1(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})), ..., J_{N_o}(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x}))] \\
&\boldsymbol{\theta}(\boldsymbol{x}) = [\theta_1(\boldsymbol{x}), ..., \theta_{N_p}(\boldsymbol{x})] \\
&\theta_1(\boldsymbol{x}) = \frac{1}{2} \left(\prod_{k=1}^{N_o+N_p-1} x_k \right) (1 + 5f(\boldsymbol{x})) \\
&\theta_i(\boldsymbol{x}) = \frac{1}{2} \left(\prod_{k=1}^{N_o+N_p-i} x_k \right) \left(1 - x_{N_o+N_p-i+1}\right) (1 + 5f(\boldsymbol{x})) \forall i = 2, 3, ..., N_p \\
&J_j(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x})) = \frac{1}{2} \left(\prod_{k=1}^{N_o-j} x_k \right) (1 - x_{N_o-j+1}) (1 + 5f(\boldsymbol{x})) \forall j = 1, 2, ..., N_o \\
&f(\boldsymbol{x}) = N_{var} + \sum_{m=1}^{N_{var}} \left((x_m - 0.5)^2 + \cos\left(\pi(x_m - 0.5)\right) \right)
\end{aligned}
\tag{A.3}
$$

APPENDIX B

CAMBER MORPHING WING DESCRIPTION*

The CST equations are bijective, hence two equations are necessary to fully represent the upper and lower surfaces of the airfoil. The two surfaces are denoted herein as active (lower) and passive (upper). While the passive surface is restricted to have constant length but it is free to bend, the active surface is free to both constrict and bend. This represents the case where the lower surface contains active material. The CST equations are formulated in the non-dimensional $\psi - \zeta$ domain that is related to the physically meaningful domain $x - z$ via the airfoil chord c ($\psi = x/c$ and $\zeta = y/c$) depicted in Fig. 3.1. The CST equations for subsonic airfoils ($N_1 = 0.5$ and $N_2 = 1$) with trailing edge thickness $\Delta\zeta_{TE}$ are as follows:

$$\zeta_A = \psi^{0.5}(1-\psi)\sum_{i=1}^{n} A_i \frac{n!}{i!(n-i)!}\psi^i(1-\psi)^{n-i} + \psi\frac{\Delta\zeta_{TE}}{2}\,, \tag{B.1}$$

$$\zeta_P = \psi^{0.5}(\psi-1)\sum_{i=1}^{n} P_i \frac{n!}{i!(n-i)!}\psi^i(1-\psi)^{n-i} - \psi\frac{\Delta\zeta_{TE}}{2}\,, \tag{B.2}$$

where ζ_A and ζ_P are the non-dimensional distances perpendicular to the chord to the active and passive surfaces, ψ is the non-dimensional distance along the chord ($\psi \in [0, 1]$), A_i and P_i are the shape function coefficients, and n is the order of the Bernstein polynomials.

Leal and Hartl [61] modified the CST equations so that children airfoils (i.e., morphed configurations) can be generated from any parent airfoil (i.e., original configuration) while considering kinematic constraints related to the internal structure; thus, all of the explored designs herein are known to be physically feasible for low strains ($\leq 4\%$), even without recourse to full structural analysis. In the model it is assumed that the overall morphing structure has or maintains the following: rigid body spars, constant leading edge radius, constant passive surface length, and constant

*This section is reprinted with permission from “Parametric optimization for morphing structures design: application to morphing wings adapting to changing flight conditions” by Weaver-Rosen, J.M., Leal, P.B.C., Hartl, D.J., and Malak, R.J., 2020. *Structural and Multidisciplinary Optimization*, 62, 2995-3007 Copyright 2020 by Springer Nature Switzerland AG [2].

angles between spars and passive surface. The equations above can be used to describe the children and parent configurations (denoted by a superscript c and p, respectively) by varying the values of the shape coefficients. Using an owl airfoil as an example parent airfoil [121] and an internal structure consisting of five spars, a different OML can be generated at minimal computational cost (i.e., indirect calculation) that preserves the spar dimensions as shown in Fig. 3.1.

A system of equations is formulated by applying Eqn. (B.1) for n spar intersections with the child active surface, located at coordinates $(\psi^c_{A,j}, \zeta^c_A(\psi^c_{A,j}))\, \forall\, j \in [1, n]$, as follows:

$$f_j = \sum_{i=0}^{n} F_{ji} A^c_i \quad \text{or} \quad \mathbf{f} = \mathbf{F}\,\mathbf{A}\,, \tag{B.3}$$

where $\mathbf{A}$ is the active shape coefficient vector (n-dimens-ional), $\mathbf{f}$ is the forwards restriction vector (n-dimensional)

$$f_j = \frac{2\;\zeta^c_A(\psi^c_{A,j}) + \psi^c_{A,j}\Delta\zeta^c_{TE}}{2\sqrt{\psi^c_{A,j}}(\psi^c_{A,j}-1)} - A_0(1-\psi^c_{A,j})^n\,, \tag{B.4}$$

and $\mathbf{F}$ is the forwards function matrix ($n \times n$ dimensional), given as

$$F_{ji} = \frac{n!}{i!(n-i)!}(\psi^c_{A,j})^i(1-\psi^c_{A,j})^{n-i}\,. \tag{B.5}$$

If all coordinates $(\psi^c_{A,j}, \zeta^c_A(\psi^c_{A,j}))$ can be calculated, $\mathbf{F}$ is a square tensor and invertible, and the values of the active children shape coefficients A^c_i are determined by solving the linear system of equations (Eqn. (B.3)). Therefore the only free degrees of freedom for the children configurations are the passive shape coefficients $\mathbf{P}^c$.

As detailed in [61], a set of kinematic constraints is utilized to guarantee structural feasibility of the morphed configurations and to establish coordinates $(\psi^c_{A,j},\, \zeta^c_A(\psi^c_{A,j}))$ as a function of $\mathbf{A}^p$, $\mathbf{P}^p$, $\mathbf{P}^c$, and the locations and heights of spars. In the context of real aircraft, the parent airfoil and much of its internal structure are fixed and known. Therefore, the children airfoils are only a function of $\mathbf{P}^c$.

Since the framework i) guarantees kinematically acceptable deformation states, ii) only consists of linear system of kinematic equations, iii) assumes that enough force is generated to obtain the desired strains, and iv) strictly limits the strains generated anywhere in the morphing body to be within material limits, structural analysis is not necessary and the generation of new children configurations is greatly expedited. The framework is independent of actuator technology, but previous work has shown that shape memory alloy actuators, for example, can generate the necessary strains for a camber morphing wing [121].