Pareto Gamuts: Exploring Optimal Designs Across Varying Contexts

by

Liane Makatura

A.B., Dartmouth College (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2020

Author . . . .
Department of Electrical Engineering and Computer Science
August 28, 2020

Certified by . . . .
Wojciech Matusik
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Pareto Gamuts: Exploring Optimal Designs Across Varying Contexts

by

Liane Makatura

Submitted to the Department of Electrical Engineering and Computer Science
on August 28, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

Manufactured parts are meticulously engineered to perform well with respect to several conflicting metrics, like weight, stress, and cost. The best achievable trade-offs reside on the *Pareto front*, which can be discovered via performance-driven optimization. Objective functions used to define the Pareto front often incorporate assumptions about the *context* in which a part will be used, including loading conditions, environmental influences, material properties, or regions that must be preserved to interface with a surrounding assembly. Existing multi-objective optimization tools are only equipped to study one context at a time, so engineers must run independent optimizations for each context of interest. However, engineered parts frequently appear in many contexts: wind turbines must perform well in many wind speeds, and a bracket might be optimized several times with its bolt-holes fixed in different locations on each run. In this paper, we formulate a framework for variable-context multi-objective optimization. We introduce the *Pareto gamut*, which captures Pareto fronts over a range of contexts. We develop a global-local optimization algorithm to discover the Pareto gamut directly, rather than discovering a single fixed-context "slice" at a time. To validate our method, we adapt existing multi-objective optimization benchmarks to contextual scenarios. We also demonstrate the practical utility of Pareto gamut exploration for several engineering design problems.

Thesis Supervisor: Wojciech Matusik
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank Professors Wojciech Matusik, Justin Solomon, and Adriana Schulz for serving as my advisors and mentors throughout this project. Their advice, enthusiasm, and support has been invaluable, both professionally and personally.

I would also like to thank my colleagues in the Computational Design and Fabrication Group (CDFG) and the Geometric Data Processing (GDP) Group, for many fruitful discussions. I would like to offer special thanks to the following lab mates: Harrison Wang, for his early contributions to this project; Tao Du, for his initial explanations and encouragement; Paul Zhang, David Palmer, and Edward Chien, for their mathematical insights; Minghao Guo, for help with examples and comparisons to previous work; and Tim Erps, Mike Foshey, Andy Spielberg, and James Minor for discussions about potential use cases.

Finally, I would like to thank my family and friends, whose unconditional love and support have made this journey possible. To my partner, Hannes – you've supported me and this project in more ways than I can count, with everything from jokes to JavaScript. I am so grateful to have you by my side. And of course, to my parents: thank you for always believing in me, and making sure I had the right tools and mindset to succeed. I would not be who I am without you!

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1122374.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Performance-driven optimization is a cornerstone for effective engineering design, because it allows engineers to fine-tune parts based on their predicted performance in the real world. This is critical because most engineering design processes are driven by a set of high-level performance specifications. For example, favorable wrench designs must be durable, lightweight, and capable of transforming a small input force into a large torque. Similarly, wind turbines should be compact with a high power output; bicycle components should be lightweight and strong; and a robot's locomotion should be fast and energy-efficient. Strategic optimization is necessary because the space of possible designs is typically very large, and the performance of any given design is non-obvious and expensive to evaluate. Thus, an unguided design process would require a laborious trial-and-error approach that expends significant effort on infeasible or undesirable candidates.

Multi-objective optimization schemes are particularly relevant for engineering design, because the desirable performance metrics typically *conflict* with one another (as in the examples above). In this case, there is no single optimal solution. Instead, the engineer must consider the set of *Pareto-optimal* designs, which exhibit the best achievable performance trade-offs between the given metrics. More precisely, a design is *Pareto-optimal* if it is impossible to improve its performance on all metrics at the same time; improving any metric will necessarily worsen at least one other metric. Fig. 1-1 illustrates this concept for a wind turbine. The most notable trade-off oc-

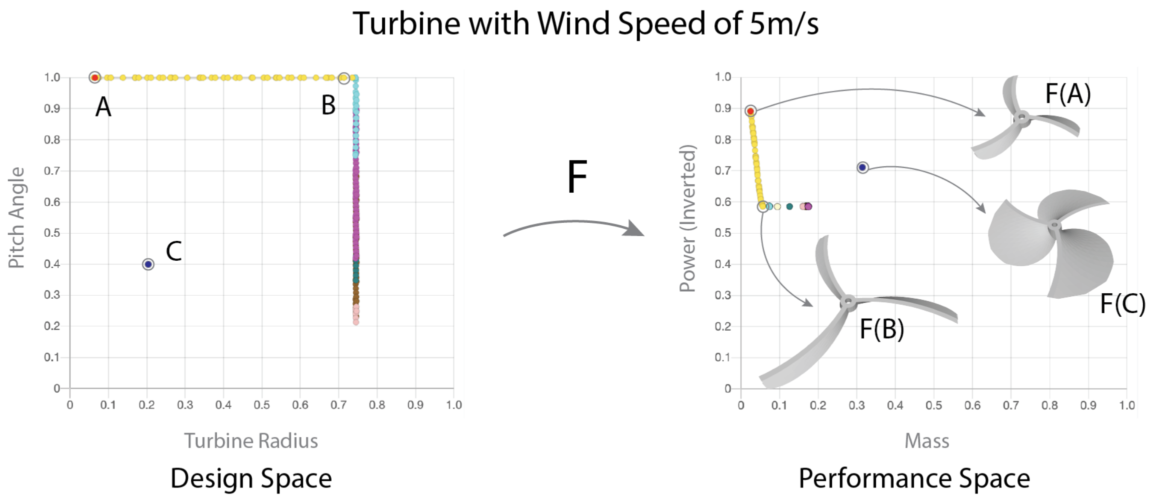

Figure 1-1: Trade-off between two metrics for a wind turbine; both objectives are formulated for minimization (i.e., smaller performance values are better). Designs A and B are Pareto optimal: improving the design's performance on either metric will worsen its performance w.r.t. the other metric. However, design C is **not** Pareto optimal, because it is *dominated* by other designs (like B) that exhibit better performance on both metrics.

curs as the blade radius grows: the power output improves but the turbine becomes heavier. Without additional information, there is no obvious ranking between (A) a lightweight, low-power turbine and (B) a heavy, high-power turbine. However, it is clear that there is never a reason to select a turbine design (C) that is both heavy *and* low-powered, because it is possible to do better on at least one of these metrics without sacrificing performance on the other. Thus, it is reasonable to consider only the Pareto-optimal designs like (A) and (B), which exhibit the strongest possible trade-offs. This restricted exploration is enabled by multi-objective optimization schemes, which directly discover such a set of Pareto-optimal designs. This permits better design decisions by automatically culling sub-optimal regions of the design space and focusing the user's attention on the most promising solutions.

To use a multi-objective optimization scheme, engineers must first parametrize the given part with a set of *design variables*. Design variables specify aspects of the design (e.g. handle length, blade radius) that are allowed to vary throughout the optimization process. To measure the performance of each design, engineers must also define the set of relevant *performance metrics* (e.g. mass, power, or stress) and any external constraints (e.g. fixed positions) or assumptions (e.g. loading conditions, temperature, fluid density) derived from the part's intended *context*. Context information is critical to ensure that the final design is properly tailored to its intended purpose. Thus, the context parameters must be carefully chosen, and their values must remain fixed during the optimization process.

Nevertheless, it is often desirable to examine the Pareto fronts for several distinct contexts before committing to a final design. Existing optimization tools are only able to provide insight into a single context at a time, which renders thorough contextual exploration both tedious and time-consuming. We address this challenge by developing a novel framework for variable-context optimization.

The remainder of this chapter presents several motivating scenarios for contextual exploration (Section 1.1), then provides an overview of our proposed approach (Section 1.2 and 1.3).

1.1 Variable-Context Optimization: Motivation

Standard engineering practices have shown that existing fixed-context optimization procedures are insufficient. This is because manufactured parts are often used in several contexts, each of which presents a unique set of external constraints and assumptions. For example, common "building blocks" like lugs and brackets are adapted to many specific bending angles or bolt hole locations (Fig. 1-2, (a)). Engineers maintain a library of optimal designs for common contexts and compute the solutions for each non-standard request independently. For fast adaptation, it would be helpful to know the Pareto-optimal designs for *any* specific context, but this is prohibitively expensive with existing tools.

In other cases, parts must be designed for dynamic contexts (Fig. 1-2, (b)). For instance, a turbine's power output depends on environmental factors like wind speed and air density, which can take on a *range* of conditions [Johnson, 2006]. Similarly, mounting brackets must withstand a range of loading conditions. With current optimization tools, engineers often consider dynamic contexts by computing the optimal

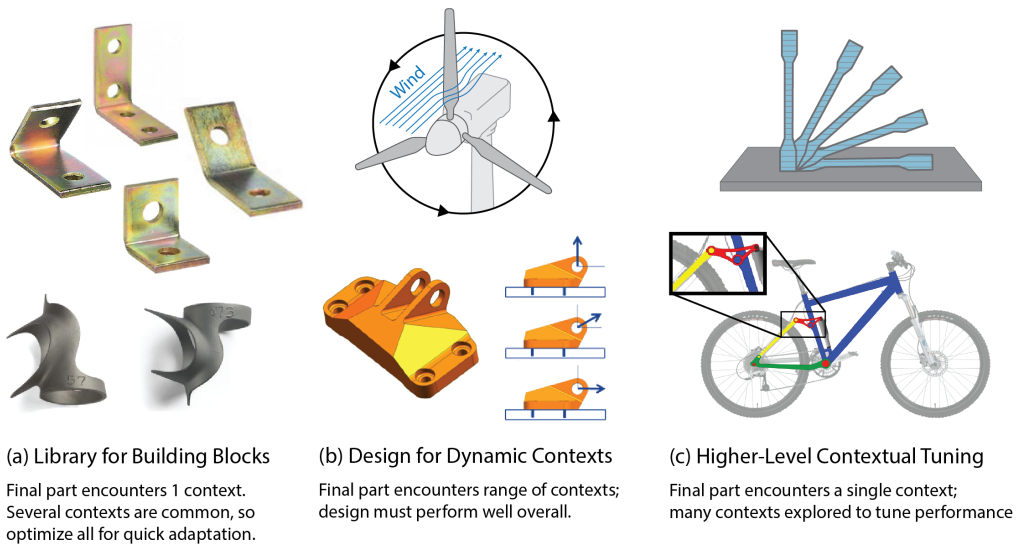

Figure 1-2: Multi-context design examples; each column illustrates a distinct problem class (context in italic). **(a)** Manufacturers maintain libraries of basic components to suit various contexts. We show angle brackets (top) and bicycle lugs (bottom) for different *angle constraints*. **(b)** Some parts encounter many contexts: turbines (top) must perform well under many *wind speeds* and *temperatures*; mounting brackets (bottom) must withstand a range of loading conditions. **(c)** Contexts might be explored to gauge higher-level trade-offs. For 3D-printing (top), *orientation* or *layer height* affect the properties of each individual part, but engineers may also set these print parameters to optimize the run overall (capacity for other parts, print time, etc.). Assemblies like bicycles (bottom) present hierarchical design decisions, because the *location of functional interfaces* (e.g., pivot between red rocker and yellow stay) affects the achievable performance of the individual parts and the assembly overall.

designs for some “representative” context. In many cases, this representative is either an average over some distribution of expected contexts, or a carefully constructed “worst-case” scenario [Panetta et al., 2017, Schumacher et al., 2018, Ulu et al., 2017]. The resulting designs are often sub-optimal with respect to the individual contexts, but the engineer has no efficient way to compare the performance that is sacrificed due to the generalized context assumptions. Alternatively, engineers may sparsely sample the desired context range and evaluate their designs only with respect to these selected contexts [Montazersadgh and Fatemi, 2007, GE, 2013]. Some industrial design programs like SolidWorks and Altair Inspire include tools to manage this ad hoc contextual exploration, but the process remains inefficient. According to a SolidWorks director of product management, “you never hit run once [for any optimization]. You look at the results and run it again with different assumptions and inputs to evaluate those inputs. Then you run it again. It’s an iterative process” [Wasserman, 2017]. This process can be time-consuming and relies on expert knowledge to select representative contexts.

Expert knowledge is also a cornerstone of the third class of contextual problems: hierarchical design (Fig. 1-2, (c)). For example, high-performance bicycles include dozens of sub-assembly components, and hundreds of tuneable design choices [Seven Cycles, 2020]. Each sub-assembly has its own set of performance metrics (e.g., minimal mass and compliance), so it is prohibitively expensive to compute a global optimization over the entire system. To make the problem tractable, engineers begin by blocking out the rough assembly, including the length and adjoining angles of the frame tubes and the location of pivot points within the rear suspension system [Covill et al., 2014, Spratt, 2019]. After these assembly-level constraints are set (thus fixing the context of each part), the individual elements are optimized for performance [Bogomonly, 2019, Spratt, 2019]. It would be beneficial for engineers to explore the designs resulting from many assembly-level configurations, because even slight variations can have a drastic effect on the global performance of the bike [PHeller, 2014, Covill et al., 2014]. However, each assembly-level change (e.g., pivot position) imposes a new context on the sub-assembly components, causing a change in the Pareto set for each component. Thus, to understand the performance of the new assembly, many individual parts must be re-optimized under their new context. This limits potential exploration, as each change would require repeated optimization of the bike's components (tubes, lugs, yokes, etc.) under the new constraints.

1.2 Our Approach

In this thesis, we formulate a framework for variable-context multi-objective optimization. Beyond demonstrating how to incorporate context variables into the multi-objective optimization formalism, we provide an algorithm that seeks the Pareto front at *every* context in a desired range, rather than discovering one fixed-context "slice" at a time. This collection of Pareto fronts forms a new object, which we call the *Pareto gamut* (see Fig. 1-3).

We present a simple global/local algorithm for discovering the Pareto gamut, which alternates between two steps. In the first step, we use global optimization to uncover a few points on the Pareto gamut. The context value for each sample is chosen randomly within our range. This allows us to explore multiple contexts simultaneously, which encourages even coverage across the Pareto gamut. In the second step, we perform local exploration to uncover Pareto-optimal designs in the neighborhood of each point computed in the first step. The local exploration uses a first-order approximation that takes a single Pareto-optimal point in *some* context and identifies directions in the joint design-context space along which we can walk without leaving the Pareto gamut. This novel formulation allows us uncover entire patches of the Pareto gamut at once, which provides dense coverage of the contextual range without incurring the high cost of computing each fixed-context Pareto front independently.

Our resulting Pareto gamut addresses several classes of contextual design problems. It improves contextual resolution for "building block" libraries by construction, as the Pareto gamut provides rapid access to all fixed-context Pareto sets within a given range. The Pareto gamut also provides additional insight for dynamic con-

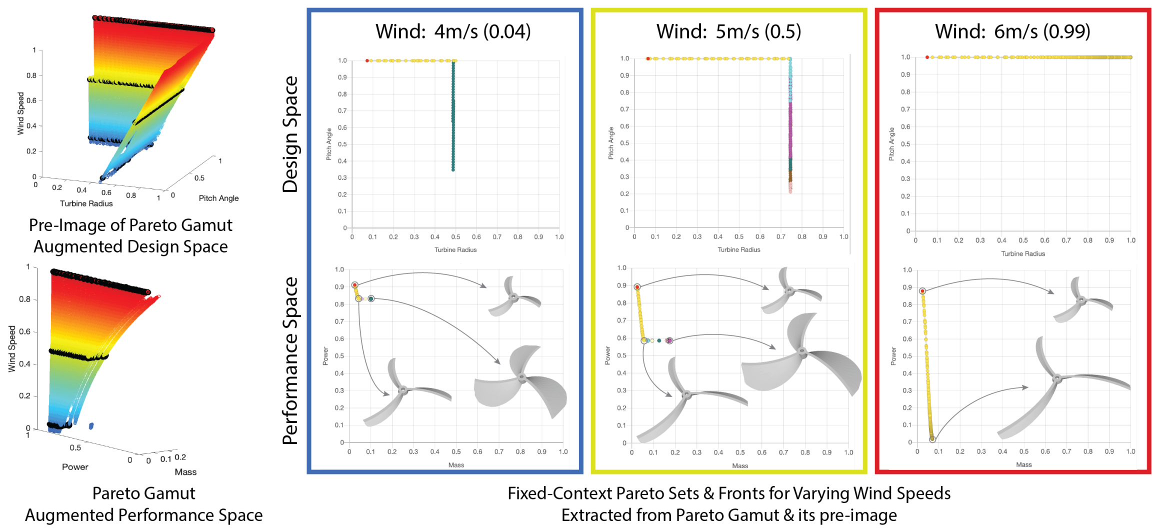

Figure 1-3: Manufactured parts are frequently optimized for performance in many different contexts: here, turbines are optimized for mass and power (performance) under various wind speeds (context). Each context yields a distinct *Pareto set*, the collection of designs with optimal performance trade-offs. Existing tools require a separate optimization for each context of interest (right, border colored by context value). As an alternative, we augment the standard multi-objective optimization framework by simultaneously considering design *and* context variables. This allows us to find the *Pareto gamut* (left), which captures the Pareto-optimal designs over a range of contexts (colored by context value). From this gamut, we can extract any fixed-context Pareto set (right, top) and its image in performance space (right, bottom).

text problems, by allowing engineers to explore a particular design relative to the achievable Pareto trade-offs in any context. We also explore the Pareto gamut as an initial step toward hierarchical optimization for largely-separable structures such as the bicycle suspension system shown in Fig. 1-2.

1.3 Thesis Contributions

At a high level, our contributions include:

- codifying the concepts and use cases for contextual parameters and Pareto gamuts,
- developing a mathematical framework for variable-context optimization,
- devising a global/local discovery algorithm for the full Pareto gamut, and
- prototyping visualization tools for interactive Pareto gamut exploration.

These contributions address an important under-explored problem in engineering design: engineers frequently explore multiple contexts before selecting a final design, but existing tools must discover each desired Pareto front from scratch. Our solution mitigates this problem by proposing a new structure (the Pareto gamut) that (1) codifies the way in which the Pareto set/front changes as a function of context, and (2) permits us to propagate information about optimal designs across a neighborhood of contexts during discovery. This amortizes the cost of optimization over the entire context range, and enables the elusive task of simple, thorough contextual exploration.

Chapter 2

Related Work

This thesis draws inspiration from several disciplines, including multi-objective optimization, computer graphics, and engineering design.

2.1 Multi-Objective Optimization

Multi-objective optimization has received considerable research attention over the last several decades, as it is a powerful tool with widespread applications in fields such as engineering and economics. We present a brief overview of multi-objective optimization methods to situate our problem. We then focus on one particular class of methods, *continuation methods*, that are most closely related to our work. For an in-depth survey of multi-objective optimization methods, we refer the reader to the works by Cho et al. [2017] and Marler and Arora [2004].

2.1.1 General Overview

There are three classes of multi-objective optimization algorithms, based on the timing of user guidance relative to optimization: before (*a priori*), during (interactive), or after (*a posteriori*) [Van Veldhuizen and Lamont, 2000]. The first two classes allow users to guide the optimization toward "desirable" regions. Interactive methods present intermediate solutions to the user, and require feedback about which search directions to explore next [Ruiz et al., 2019, Meignan et al., 2015, Koyama et al., 2020]. For *a priori* methods, users generally encode their desired trade-off in a single utility function (e.g., weighted sum of metrics) that is solved via single-objective optimization methods [Marler and Arora, 2004]. These approaches require relatively few samples, but the limited user-guided exploration is biased toward known solutions, which reinforces user expectations and reduces novel insights. By contrast, *a posteriori* methods solicit user preferences only *after* the Pareto front has been discovered. Popular *a posteriori* methods include Normalized Boundary Intersection methods [Das and Dennis, 1998], Normalized Normal Constraint methods [Messac et al., 2003], and evolutionary algorithms [Zhang and Xing, 2017, Deb et al., 2002, Deb and Jain, 2014]. By exposing the set of Pareto-optimal solutions before making decisions, *a posteriori* algorithms increase understanding of the attainable trade-offs while mitigating the influence of known solution biases, at the cost of additional optimization effort.

We develop an *a posteriori* approach over the joint design-context space. This enables exploration of fixed-context Pareto fronts, while highlighting how the fronts vary when the context changes.

2.1.2 Continuation Methods

Algorithms in the previous work mentioned above suffer from a common pitfall: each point on the front must be found independently. This discrete approximation is inefficient, as it neglects the fact that Pareto-optimal points are typically clustered together [Hillermeier, 2001a]. We capitalize on this insight by developing a continuation scheme that uncovers large regions of the Pareto gamut at a time.

Via an extension of the classical KKT conditions [Karush, 1939, Kuhn and Tucker, 1951] (see Section 3.1), solutions of multi-objective optimization problems satisfy a constrained system of nonlinear equations with locally continuous solution manifolds [Hillermeier, 2001b,a]. This observation enables path-following methods, or *continuation schemes*, which explore the neighborhoods around Pareto-optimal points to discover connected, continuous regions of the front. Rakowska et al. [1991] first applied this approach to bi-objective problems, revealing interpretable design families and insight about the cause of discrete jumps within the Pareto set for specific problems. These results were later generalized for k objectives [Hillermeier, 2001a, Martín and Schütze, 2018], but these papers only consider local expansion from a single known solution. This can be addressed by embedding local continuation within a global/local discovery algorithm, as in the interactive method presented by Schütze et al. [2019] and the *a posteriori* approach of Schulz et al. [2018].

Our global/local discovery algorithm is most closely related to that of Schulz et al. [2018], whose fixed-context optimization routine provides the basis for our algorithm. We *generalize* this algorithm and its supporting theory to include contextual exploration during the discovery phase. This requires a new data structure and revisions to the global sampling, optimization, and local exploration steps. On the theoretical side, we generalize the perturbative step in Schulz et al. [2018] and the predictor step in Martín and Schütze [2018] by deriving the effect of changing context on the shape of the Pareto front. While previous works uncover a single fixed-context front, our algorithm is unique in its ability to discover the Pareto fronts over *many* contexts simultaneously.

2.2 Performance-Driven Design

Optimization is essential for performance-driven design, to identify feasible designs that realize some high-level functional specification. Myriad functional goals have been explored, including balance [Prévost et al., 2013, Wang and Whiting, 2016]; light reflection [Schwartzburg et al., 2014]; acoustic properties [Bharaj et al., 2015, Li et al., 2016]; and deformation [Chen et al., 2014]. While many of these methods are automatic, others incorporate user feedback in the design loop, coupled with optimizations that guide the user toward viable designs. This approach has been applied to several practical domains, including the design of functional furniture [Umetani et al., 2012, Yao et al., 2017] and stable flight [Umetani et al., 2014, Du et al., 2016]. Most of these works are restricted to single-objective optimization, and none consider contextual parameters in their most general form.

There is, however, a growing body of work that considers optimal structural design under a parametrized set of loading conditions. For example, Zhou et al. [2013] optimize the structural integrity of 3D printed structures based on geometry and material alone, without making any specific assumptions about the incident load. Panetta et al. [2017] develop methods to minimize the stress concentration of periodic microstructures under all unit loads. Ulu et al. [2017] optimize structures to be as lightweight as possible, while ensuring their ability to withstand any incident force configuration within a pre-determined force-magnitude budget. Similarly, Schumacher et al. [2018] optimize the strength and balance of large-scale 3D-printed objects under unknown external loads. In general, these methods handle input uncertainty or variability by computing a *worst-case load*, which tries to account for all of the most problematic loading conditions at once. This is a useful approach, particularly if the engineer has very little insight about which loading conditions to expect. However, the conservative, aggregate nature of this approach can result in designs which are over-engineered (and thus, sub-optimal) with respect to any particular context they may encounter. By contrast, our Pareto gamut exposes the Pareto fronts for a wide range of plausible contexts, each of which may present a unique subset of the "worst-case" aggregate. This provides complementary information that allows the engineer to understand their chosen design's optimality (or sacrifice thereof) relative to the achievable trade-offs in particular loading conditions that are likely to occur.

2.3 Design Exploration

Our approach is also related to methods that *explore* a design space. Recent works on material design seek the gamut of achievable material properties by sampling the space of material performance achievable by a fabrication device [Bickel et al., 2010, Dong et al., 2010, Zhu et al., 2017]. These methods probe the design space strategically to achieve a functional goal. Other methods traverse the entire design space, so users can efficiently focus on any region of interest [Shugrina et al., 2015, Schulz et al., 2017]. Schulz et al. [2017] also discusses a special case of contextual parameters, by interpolating e.g. stress distributions on a mesh subject to distinct loading conditions. However, each load must be selected and precomputed individually on the adaptive grid samples.

We draw inspiration from all of these approaches, as we seek a hybrid goal. Within any particular fixed-context, we only care to explore Pareto-optimal designs. More broadly, however, we wish to explore the Pareto-optimal designs over the breadth of contextual configurations. Our efficient, thorough contextual exploration enables insights which were previously infeasible, like measuring a particular design's performance with respect to the achievable optima in many different contexts.

Chapter 3

Mathematical Formulation

In this chapter, we extend the formal notion of Pareto optimality to incorporate contextual information. Section 3.1 reviews classical theory for fixed-context optimization. No novel ideas are contained in Section 3.1, but we augment the standard notation with a fixed context in anticipation of our variable-context formulation, as explored in Section 3.2. This latter section introduces our novel object, the Pareto gamut, with both intuitive and formal mathematical treatments. We also discuss the first order approximation of the Pareto gamut, which serves as the crux of our discovery algorithm in Section 4.

3.1 Fixed-Context Pareto Optimality

Consider a problem parameterized by a set of D design parameters and C contextual parameters. Any instantiation of this model can be represented by a design configuration $\mathbf{x} \in \mathbb{R}^D$ and a contextual configuration $\mathbf{z} \in \mathbb{R}^C$. For example, a wind turbine might be prescribed by three design variables (radius, height, and pitch angle; $D = 3$) and one environmental context parameter (wind speed; $C = 1$). We control the feasible set of values for each variable using constraint functions $g_k(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \to \mathbb{R}$ encoding the constraint $g_k(\mathbf{x}, \mathbf{z}) \leq 0$. For the wind turbine, constraints may include an upper bound on the blade radius or a minimal stress rating. We denote our constraints $g_k(\mathbf{x}, \mathbf{z})$ as *active* if $g_k(\mathbf{x}, \mathbf{z}) = 0$ and *inactive* if $g_k(\mathbf{x}, \mathbf{z}) < 0$.

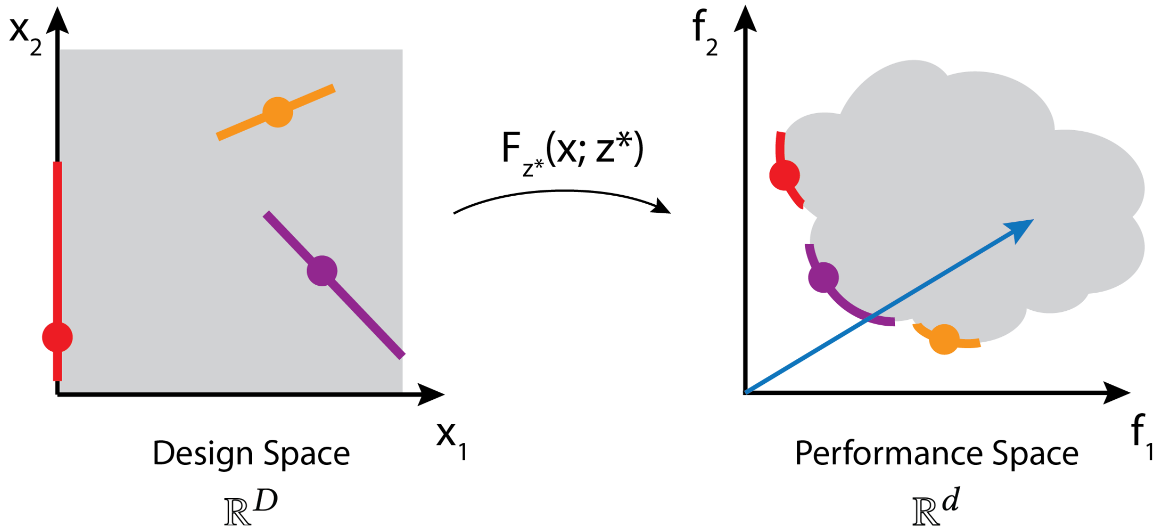

Figure 3-1: Design space $\mathcal{X}$ and performance space $F(\mathcal{X})$ shown in gray, for fixed context $\mathbf{z} = \mathbf{z}^*$. The fixed-context Pareto set $\mathcal{P}_{\mathbf{z}^*}$ (red, orange and purple) represents the design points with optimal performance trade-offs; these points map onto the Pareto front $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$. The Pareto set colors represent contiguous solution manifolds; the color is preserved when the points are mapped into performance space. Any ray from the origin (blue line) intersects the Pareto front at most once.

The classical fixed-context definition of Pareto optimality assumes that the contextual variables $\mathbf{z} \in \mathbb{R}^C$ are fixed to particular values $\mathbf{z} = \mathbf{z}^*$ *a priori*. Since $\mathbf{z}^*$ is effectively a constant, it is typically excluded from the notation; we preserve this dependence to lay the groundwork for upcoming sections. We begin by formalizing the fixed-context design space, as illustrated in Fig. 3-1:

Definition 3.1 (Design Space). *For fixed context $\mathbf{z}^* \in \mathbb{R}^C$, the feasible design space is defined as*

$$\mathcal{X}_{\mathbf{z}^*} := \{\mathbf{x} = (x_1, \ldots, x_D) \in \mathbb{R}^D \;\mid\; g_k(\mathbf{x}; \mathbf{z}^*) \leq 0, \forall k \in \{0, \ldots, K\}\} \;.$$

Intuitively, the design space represents the set of valid, manufacturable designs corresponding to the given context $\mathbf{z}^*$.

Each design in this space can be evaluated with respect to a set of d performance metrics $f_{\mathbf{z}^*,i}(\mathbf{x}) := f_i(\mathbf{x}; \mathbf{z}^*) : \mathbb{R}^D \to \mathbb{R}$. For most engineering problems, $d \ll D$. We let $F_{\mathbf{z}^*}(\mathbf{x}; \mathbf{z}^*) : \mathbb{R}^D \to \mathbb{R}^d$ be the concatenation of all performance metrics:

$$F_{\mathbf{z}^*}(\mathbf{x}) := (f_1(\mathbf{x}; \mathbf{z}^*), \ldots, f_d(\mathbf{x}; \mathbf{z}^*)).$$

The image of the feasible set $\mathcal{X}_{\mathbf{z}}^*$ under the performance metrics yields the *performance space*:

Definition 3.2 (Performance Space). *The fixed-context performance space is the image of the design space $\mathcal{X}_{\mathbf{z}^*}$ under the map $F_{\mathbf{z}^*}$:*

$$F_{\mathbf{z}^*}(\mathcal{X}_{\mathbf{z}^*}) \subseteq \mathbb{R}^d \ .$$

Without loss of generality, we assume that smaller values of f_i are desirable. Then, our multi-objective optimization problem can be notated

$$\min_{\mathbf{x}} \ \{f_i(\mathbf{x}; \mathbf{z}^*)\} \text{ subject to } \mathbf{x} \in \mathcal{X}_{\mathbf{z}^*} \tag{3.1}$$

Solutions of Eqn. 3.1 are *Pareto-optimal* in the following sense:

Definition 3.3 (Fixed-Context Dominance). *Let $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}_{\mathbf{z}^*}$ be feasible design points. We say $\hat{\mathbf{x}}$ dominates $\mathbf{x}$ in context $\mathbf{z}^*$ if $\hat{\mathbf{x}}$ performs at least as well as $\mathbf{x}$ on all metrics:*

$$f_i(\hat{\mathbf{x}}; \mathbf{z}^*) \leq f_i(\mathbf{x}; \mathbf{z}^*), \forall i \in \{1, \ldots, d\},$$

and $\hat{\mathbf{x}}$ is strictly better than $\mathbf{x}$ on at least one metric:

$$\exists i \in \{1, \ldots, d\} \ s.t. \ f_i(\hat{\mathbf{x}}; \mathbf{z}^*) < f_i(\mathbf{x}; \mathbf{z}^*).$$

Definition 3.4 (Fixed-context Pareto Optimality). *A point $\mathbf{x} \in \mathcal{X}_{\mathbf{z}^*}$ is Pareto-optimal if there does not exist any $\hat{\mathbf{x}} \in \mathcal{X}_{\mathbf{z}^*}$ that dominates $\mathbf{x}$. The set of Pareto-optimal points is the fixed-context* Pareto set *$\mathcal{P}_{\mathbf{z}^*} \subseteq \mathbb{R}^D$; the image $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*}) \subseteq \mathbb{R}^d$ is the fixed-context* Pareto front.

To build intuition, assume that our performance metric evaluations are nonnegative and consider drawing a ray from the origin in any direction $\alpha \in \mathbb{R}^d_+$. As illustrated in Fig. 3-1, there exists at most one $t \geq 0$ such that $t\alpha \in F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$. Thus, for our minimization problem, the best designs are roughly those that map closest to the origin along different directions.

Generically, the Pareto front and set are composed of collections of $(d-1)$-dimensional manifolds [Hillermeier, 2001a], implying that almost any Pareto-optimal design sits within some Pareto-optimal neighborhood. This insight is exposed via the *dual* formulation of Eqn. 3.1, as expressed by the KKT conditions [Hillermeier, 2001b]:

Proposition 3.1 (KKT conditions)**.** *Let f_i and g_k be continuously differentiable, and assume the vectors*

$$\{\nabla_x g_{k'} \mid k' \textit{ is an index of an active constraint}\}$$

are linearly independent. Then, for any $\mathbf{x}^* \in \mathcal{P}_{\mathbf{z}^*}$, there exist dual variables $\boldsymbol{\alpha}^* \in \mathbb{R}^d, \boldsymbol{\beta}^* \in \mathbb{R}^K$ such that:

$$\mathbf{x}^* \in \mathcal{X}_{\mathbf{z}^*}, \tag{3.2}$$

$$\alpha_i \geq 0 \quad \forall i \in \{1, \ldots, d\}, \tag{3.3}$$

$$\beta_k \geq 0 \quad \forall k \in \{1, \ldots, K\}, \tag{3.4}$$

$$\sum_{i=1}^{d} \alpha_i = 1, \tag{3.5}$$

$$\beta_k g_k(\mathbf{x}^*; \mathbf{z}^*) = 0 \quad \forall k \in \{1, \ldots, K\}, \textit{ and} \tag{3.6}$$

$$\left[\sum_{i=1}^{d} \alpha_i \nabla_{\mathbf{x}} f_i(\mathbf{x}^*; \mathbf{z}^*)\right] + \left[\sum_{k=1}^{K} \beta_k \nabla_{\mathbf{x}} g_k(\mathbf{x}^*; \mathbf{z}^*)\right] = 0. \tag{3.7}$$

The KKT conditions are necessary (but not sufficient) for Pareto-optimality. Differentiating them with respect to $\mathbf{x}$ yields a first-order approximation of the Pareto front about $(\mathbf{x}^*; \mathbf{z}^*)$, revealing $(d-1)$ directions in *design space* that locally preserve

Pareto optimality. This observation is the crux of continuation schemes for fixed-context optimization [Hillermeier, 2001a, Martín and Schütze, 2018, Schulz et al., 2018] and is also the basis of our local exploration step (Section 3.2.4).

3.2 Pareto Gamuts

The previous section defines various aspects of Pareto optimality with respect to a fixed context. An engineer might use this formalism to compute design trade-offs for a *particular* context of the wind turbine with wind speed V, for example. To select a final design that performs well at the site overall, however, engineers must explore the optimal designs over a *range* of contexts. To facilitate this process, we introduce the *Pareto gamut*, which captures the Pareto fronts at all contexts in a given range.

Intuitively, one can think of the Pareto gamut as a collection of fixed-context Pareto fronts that have been stacked along an additional axis according to their context value. Similarly, the pre-image of the Pareto gamut can be seen as a stack of all the fixed-context Pareto sets within the desired context range. This concept is illustrated in Fig. 3-2.

The remainder of this section posits the formal definition of the Pareto gamut and explores several of its important properties, including its first order approximation (Section 3.2.4, 3.2.5). Much like the fixed-context scenario, our approximation yields a set of directions that locally preserve Pareto optimality. However, our directions are a generalization of the previous work, as they additionally encode the way in which the Pareto set/front change as a function of context. As mentioned, this approximation allows us to propagate information about optimal designs across independent contexts and, thus, avoid the need to discover each fixed-context front from scratch.

3.2.1 Setup and Definition

Given a bounded range of interest $[z_i^{\min}, z_i^{\max}]$ for each variable, the set of contextual configurations is as follows:

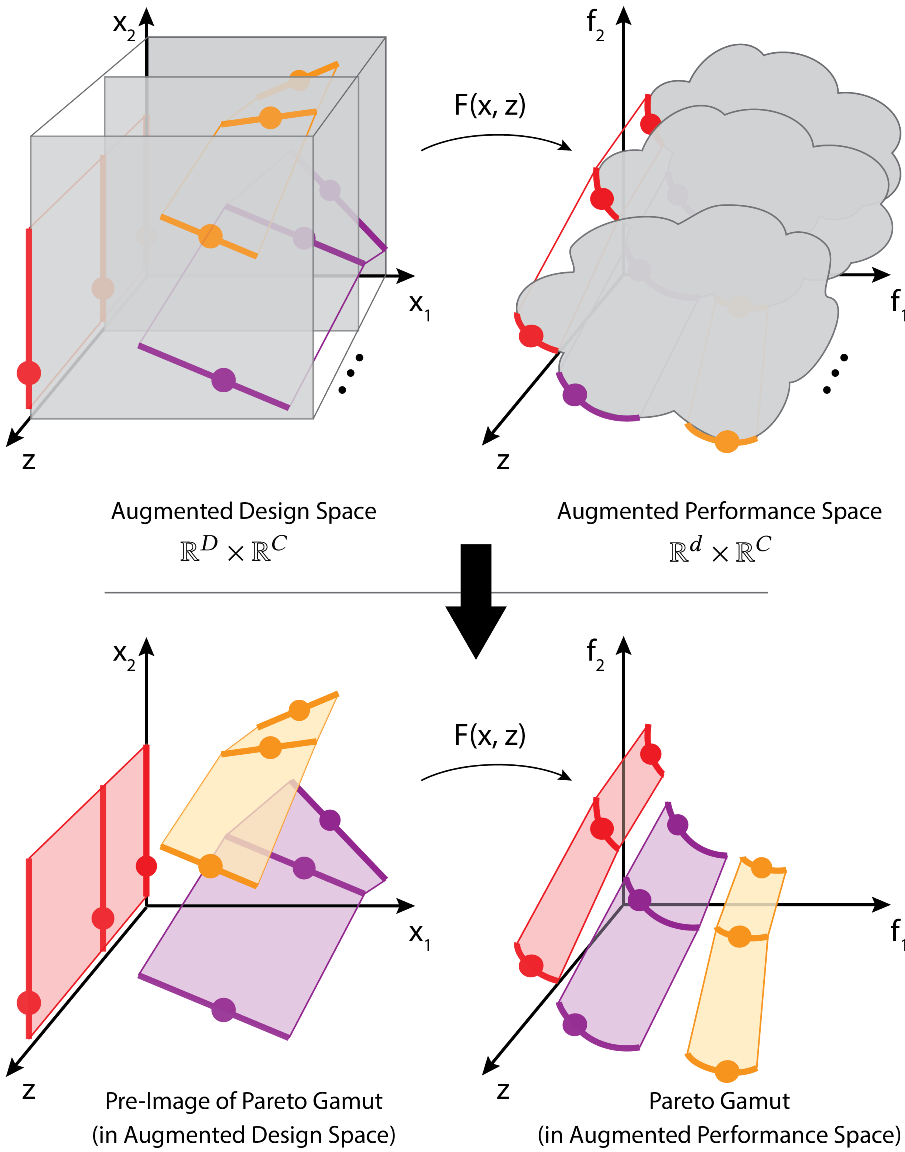

Figure 3-2: (**Top**) Augmented design (performance) space is the union of infinitely many fixed-context design (performance) spaces, drawn in gray. Each fixed-context admits a corresponding Pareto set (front), with contiguous families shown in orange, purple and red. **(Bottom)** The Pareto gamut $F(\mathcal{P})$ is the union of all Pareto fronts in augmented performance space; the pre-image of the Pareto gamut $\mathcal{P}$ is the union of all Pareto sets in augmented design space.

Definition 3.5 (Context Space)**.** *The context space $\mathcal{Z}$ is given by*

$$\mathcal{Z} := \{\mathbf{z} = (z_1, \dots, z_C) \in \mathbb{R}^C \;\mid\; z_i^{min} \leq z_i \leq z_i^{max}, \;\; \forall i \in \{1 \dots C\}\} \; .$$

Each $\mathbf{z}^* \in \mathcal{Z}$ yields a distinct Pareto set and front. Thus, to keep track of the context for a given Pareto-optimal point, we define the following *augmented spaces*:

Definition 3.6 (Augmented Spaces)**.** *The augmented design space is*

$$\mathcal{X} := \{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^D \times \mathbb{R}^C \;\mid\; \mathbf{z} \in \mathcal{Z} \textit{ and } g_k(\mathbf{x}, \mathbf{z}) \leq 0 \forall k \in \{0, \dots, K\}\} \; .$$

The augmented performance space is the image of the augmented design space under the map $F(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \to \mathbb{R}^d \times \mathbb{R}^C$, where $F(\mathbf{x}, \mathbf{z}) := (F_{\mathbf{z}}(\mathbf{x}; \mathbf{z}), \mathbf{z})$.

When $C = 1$, the augmented design (performance) space can be visualized by stacking fixed-context design (performance) spaces along a contextual axis (Fig. 3-2 (top, in gray)). Each fixed-context design (performance) space contains a Pareto set (front), as shown by the manifolds in Fig. 3-2 (top). Then, the *collection* of these Pareto sets (fronts) is our object of study (Fig. 3-2, bottom). We define this object as follows:

Definition 3.7 (Pareto Gamut)**.** *The* Pareto gamut $F(\mathcal{P})$ *is the union of all fixed-context Pareto fronts in augmented performance space:*

$$F(\mathcal{P}) = \bigcup_{\mathbf{z}^* \in \mathcal{Z}} F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*}) \quad \subseteq \mathbb{R}^d \times \mathbb{R}^C.$$

The pre-image of the Pareto gamut is denoted $\mathcal{P} \subseteq \mathbb{R}^D \times \mathbb{R}^C$.

The Pareto gamut characterizes the full set of trade-offs at any particular contextual configuration, while also encoding how these trade-offs change as a function of the context.

3.2.2 Context Parameters: A Novel Entity

As discussed in the previous section, the context must be used to augment both the design and performance spaces. However, context parameters are neither design variables nor performance metrics; they are fundamentally different from the existing machinery in multi-objective optimization.

To verify this notion, we consider two brief thought experiments to the contrary. If we treated the context as a performance metric, we would unjustly penalize large context values and bias our solutions toward high-performing designs with small context values. Alternatively, if we were to treat context parameters as design variables, we would only be able to identify the *joint* context/design configurations with optimal trade-offs. This yields a subset of the Pareto gamut containing the best-performing designs from *any* context in the permissible range; we examine this subset more closely in Section 3.2.3. Critically, neither of these context-parameter treatments yield the full Pareto gamut, which is designed to capture the entire set of trade-offs under any particular context. Thus, the context parameters must be handled as a novel entity.

This nuance comes from the fact that we would like to understand the effect of smoothly varying context parameters, but the notion of optimality only makes sense within a fixed context. In particular, any point $(\mathbf{x}^*, \mathbf{z}^*)$ on the Pareto gamut is (by construction) only required to be Pareto optimal with respect to its fixed-context, $\mathbf{z}^*$. This implies that the context values must be treated as a constant during all optimization steps, as in previous approaches. However, as we will show in Section 3.2.4, we are able to relax this constraint during non-optimization steps to explore the impact of variable contexts. To the best of our knowledge, we are the first to formalize this concept and develop the mathematics required to uncover the full Pareto gamut.

3.2.3 Gamut Lower Envelope

In this section, we describe the *lower envelope* of the Pareto gamut, which contains the best achievable trade-offs over *all* contexts (Fig. 3-3). This lower envelope offers important intuition, because it is precisely the solution we would get if we were to treat

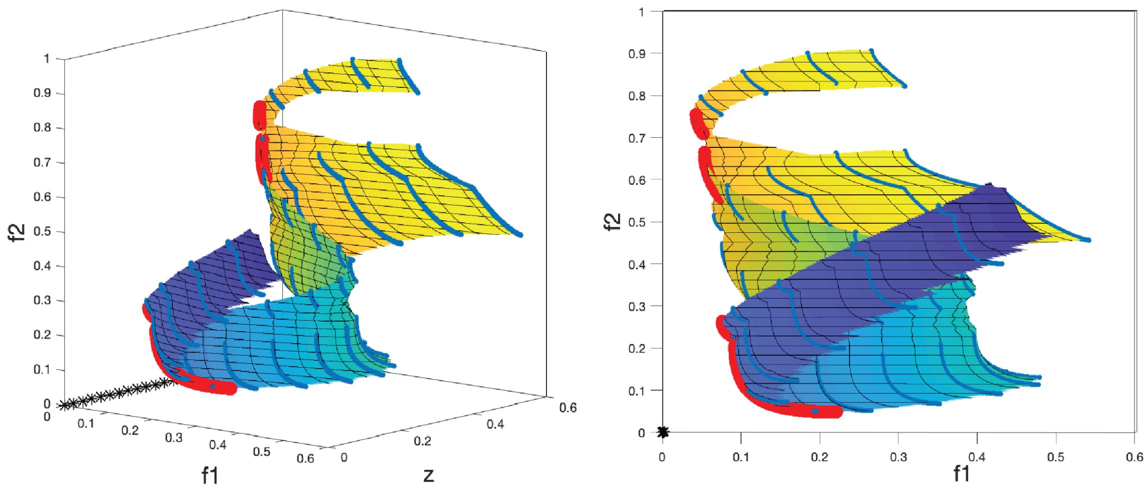

Figure 3-3: The Pareto gamut's *lower envelope* (red curve, left) contains the best achievable trade-offs over *all* contexts. By projecting away the contextual axes, we obtain the *flattened lower envelope* (red curve, right). The Pareto gamut (mesh) is for a sum of sines benchmark, colored by contextual (z) value. Blue curves indicate fixed-context Pareto fronts, and black asterisks mark the utopian corner for each fixed-context optimization ($f_1(\mathbf{x};\mathbf{z}) = f_2(\mathbf{x};\mathbf{z}) = 0$). Note that the lower envelope often spans many contexts.

the context parameters as design variables during optimization. The lower envelope also provides a direct correlation between our Pareto gamuts and a related Pareto front that can be obtained via existing multi-objective optimization benchmarks. We use this property to validate our results in Section 5.

Definition 3.8 (Lower Envelope)**.** *The lower envelope $F(\mathcal{L})$ of a Pareto gamut $F(\mathcal{P})$ is the set of gamut points that are not dominated by points in another context:*

$$F(\mathcal{L}) := \{F(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^d \times \mathbb{R}^C \ \mid \ \nexists (\hat{\mathbf{x}}, \hat{\mathbf{z}}) \in \mathcal{P} \text{ that dominates } (\mathbf{x}, \mathbf{z})\} \ . \tag{3.8}$$

The flattened lower envelope $F^{\perp}(\mathcal{L}) \subseteq \mathbb{R}^d$ is obtained by projecting away the contextual dimensions of all points in $F(\mathcal{L})$.

We illustrate the lower envelopes in Fig. 3-3. The flattened lower envelope is particularly useful, as it is linked to the Pareto front of a modified problem, where context parameters are treated as if they were design variables:

Definition 3.9 (Free-Context Problem)**.** *The free-context problem $H_F(\boldsymbol{y}) : \mathbb{R}^{D+C} \rightarrow$*

$\mathbb{R}^d$ *is a modified version of* F *that treats all contextual parameters of* F *as additional design variables that can be optimized. Namely, the design variables are* $\boldsymbol{y} := (x_1, \dots, x_D, z_1, \dots, z_C) \in \mathcal{X}$*, and each metric is* $h_i(\boldsymbol{y}) = f_i(\mathbf{x}, \mathbf{z})$*.*

The definition of Pareto optimality for H_F in the free-context problem is precisely Eqn. 3.8 in Definition 3.8. Hence, we have the following proposition:

Proposition 3.2 (Lower Envelope Optimality)**.** *For variable-context objective function* $F(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \to \mathbb{R}^d \times \mathbb{R}^C$*, we have* $H_F(\mathcal{P}) = F^{\perp}(\mathcal{L})$*, where* $H_F(\mathcal{P})$ *is the Pareto front of the free-context problem* H_F *and* $F^{\perp}(\mathcal{L})$ *is the flattened lower envelope defined above.*

The flattened lower envelope will be used in Section 5.1 to validate our approach against fixed-context multi-objective optimization benchmarks.

3.2.4 First-Order Approximation

To discover the Pareto gamut efficiently, we must be able to propagate information about Pareto-optimal points across contexts. At a high level, we seek a set of exploration directions $(\mathbf{x}', \mathbf{z}')$ that perturb the known-optimal design point $(\mathbf{x}^*, \mathbf{z}^*)$ while preserving the KKT conditions (Fig. 3-4). We recall that any point $(\mathbf{x}^*, \mathbf{z}^*)$ on the Pareto gamut is (by construction) only required to be Pareto optimal with respect to its fixed-context, $\mathbf{z}^*$. Thus, to traverse the Pareto gamut about $(\mathbf{x}^*, \mathbf{z}^*)$, we must seek a set of exploration directions $(\mathbf{x}', \mathbf{z}')_i$ such that $(\hat{\mathbf{x}}, \hat{\mathbf{z}}) := (\mathbf{x}^*, \mathbf{z}^*) + \sum_i \varepsilon_i (\mathbf{x}', \mathbf{z}')_i$ satisfies the KKT conditions for the fixed-context $\hat{\mathbf{z}}$.

To identify these directions, we draw inspiration from the fixed-context continuation schemes in Section 3.1, which use a first-order approximation of the Pareto front to exploit known solutions for local exploration. We generalize this approximation to allow exploration *across* contexts as well, as detailed in the following proposition and proof.

Proposition 3.3 (Augmented KKT Perturbation)**.** *Suppose* $(\mathbf{x}(t), \mathbf{z}(t)) : (-\varepsilon, \varepsilon) \to \mathbb{R}^D \times \mathbb{R}^C$ *is composed of Pareto-optimal points, that is,* $(\mathbf{x}(t), \mathbf{z}(t)) \in \mathcal{P}_{\mathbf{z}(t)}$ *for all*

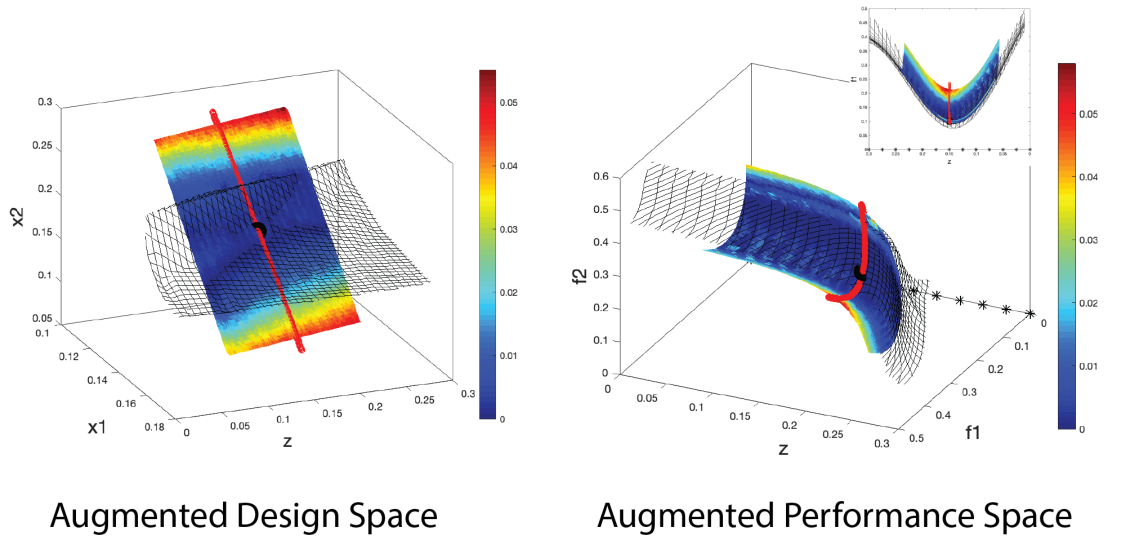

Figure 3-4: Existing continuation schemes identify a $(d-1)$-dimensional approximation (red curve) within a single fixed context of the Pareto gamut. Proposition 3.3 describes a $(d-1+C)$-dimensional approximation (colored mesh) that *contains* the previous work as a subspace, while also encoding the way in which the Pareto set/front change as a function of the context parameters. Our approximation is colored by its error relative to the ground truth wireframe mesh.

$t \in (-\varepsilon, \varepsilon)$. *Let* f_i *and* g_k *be continuously differentiable for* $1 \le i \le d$ *and* $1 \le k \le K'$, *where* $K' \le K$ *is the number of active constraints at* $(\mathbf{x}(0), \mathbf{z}(0))$. *Let* $\boldsymbol{\alpha}(t)$ *and* $\boldsymbol{\beta}(t)$ *be the KKT dual variables at* $(\mathbf{x}(t), \mathbf{z}(t))$ *(see Proposition 3.1), and define* $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}(0)$ *and* $\boldsymbol{\beta}^* := \boldsymbol{\beta}(0)$. *Then,*

$$\underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_{\mathbf{x}}G & D_{\mathbf{z}}G \\ D_x F^T & D_x G^T & Hx & Hz \end{bmatrix}}_{M \in \mathbb{R}^{(1+K'+D)\times(d+K'+D+C)}} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}'(0) \\ \boldsymbol{\beta}'(0) \\ \mathbf{x}'(0) \\ \mathbf{z}'(0) \end{bmatrix}}_{v \in \mathbb{R}^{(d+K'+D+C)}} = \mathbf{0}\ , \tag{3.9}$$

where

$$Hx = \left[\sum_{i=1}^{d} \alpha_i^* H_{xx} f_i(\mathbf{x}(0), \mathbf{z}(0)) + \sum_{k=1}^{K'} \beta_k^* H_{xx} g_k(\mathbf{x}(0), \mathbf{z}(0)) \right], \text{ and}$$
$$Hz = \left[\sum_{i=1}^{d} \alpha_i^* H_{xz} f_i(\mathbf{x}(0), \mathbf{z}(0)) + \sum_{k=1}^{K'} \beta_k^* H_{xz} g_k(\mathbf{x}(0), \mathbf{z}(0)) \right].$$

For a function $X(u, v)$, the expression $D_u X$ denotes the Jacobian of X with respect to the variables u; the Jacobians in Eqn. 3.9 are evaluated at $(\mathbf{x}(0), \mathbf{z}(0))$. Similarly, $H_{uv}X$ denotes the matrix of second derivatives obtained by taking the partials of X with respect to u and then v. The result is similar to a Hessian matrix, except that $H_{uv}X$ is asymmetric if $u \not\equiv v$. We prove Proposition 3.3 in Section 3.2.5.

Remark 1. *Eqn. 3.9 is a generalization of the results given by Schulz et al. [2018, Proposition 5.1] and Martín and Schütze [2018, Theorem 3.1]. If $\mathbf{z}'(t) \equiv \mathbf{0}$ (i.e., if we do not allow the context to change), we recover the expressions from previous work.*

Similar to the discussion in Section 3.1, Proposition 3.3 yields a set of directions $v := [\mathbf{x}', \mathbf{z}']$ along which we can walk in the augmented design space without leaving the Pareto gamut. To gain some intuition about our result, we consider a brief dimensionality analysis. As shown in Eqn. 3.9, the exploration directions v are given by the null space of M. The rank of M is at most $\min(\#\text{rows}, \#\text{cols}) = \min(1 + K' + D, d + K' + D + C) = D + 1 + K'$. Then, the Rank Nullity theorem bounds the number of independent directions v:

$$\begin{aligned}
\text{nullity}(M) &= \#\text{columns(M)} - \text{rank}(M) \\
&\geq (d + K' + D + C) - (1 + K' + D) \\
&= d - 1 + C\ .
\end{aligned}$$

Thus, by computing the null space of M, we will generically obtain a set of at least $d - 1 + C$ directions along which we can walk in augmented design space. The exact case ($\text{nullity}(M) = d - 1 + C$) is the expected dimensionality for our variable-context Pareto manifold. For intuition, recall that the Pareto manifold at any given context has dimension $d - 1$ [Hillermeier, 2001a]. Then, the gamut can be thought of as an extrusion of this fixed-context front through C-dimensional contextual space. This interpretation also provides geometric insight about the relationship between our exploration directions and those obtained by previous works: the latter form a $(d - 1)$-dimensional subspace of the space spanned by our exploration directions, illustrated in Fig. 3-4.

It can also happen that nullity$(M) > d-1+C$ (e.g. there are more than $d-1+C$ directions to choose from). This scenario typically occurs when there are design variables that have no effect on the performance metrics locally. In this case, it is sufficient to select any $d-1+C$ of the directions at random, and discard the rest.

3.2.5 Proof of First-Order Approximation

Suppose we construct a curve $\gamma(t) := (\mathbf{x}(t), \mathbf{z}(t))$, constrained to the Pareto gamut for $t \in (-\varepsilon, \varepsilon)$ as in Proposition 3.3. At a given point $(\mathbf{x}^*, \mathbf{z}^*)$ along the curve, the number of active constraints g_k is $K' \leq K$. Without loss of generality, we can permute the constraints g_k such that the first $K' \leq K$ are the active ones. Because our objectives are continuous, all (in)active constraints remain (in)active within our (potentially restricted) ε-ball of interest. Thus, inactive constraints g_k have no impact on Eqn. 3.7 (KKT stationarity), because Eqn. 3.6 (KKT complementary slackness) ensures that $\beta_k = 0$ for all relevant ε. We reformulate our KKT conditions from Proposition 3.1, in order to (1) explicitly ignore the inactive constraints (which have no effect), (2) reflect our parametrization on t, and (3) require that the contextual configuration $\mathbf{z}$ remains within its predefined feasible range:

$$(\mathbf{x}^*, \mathbf{z}^*) \in \mathcal{X} \tag{3.10}$$

$$\alpha_i(t) \geq 0 \quad \forall i \in \{1, \ldots, d\}, t \in (-\epsilon, \epsilon) \tag{3.11}$$

$$\beta_k(t) \geq 0 \quad \forall k \in \{1, \ldots, K'\}, t \in (-\epsilon, \epsilon) \tag{3.12}$$

$$\sum_{i=1}^{d} \alpha_i(t) = 1 \quad \forall t \in (-\epsilon, \epsilon) \tag{3.13}$$

$$\beta_k(t) g_k(\mathbf{x}(t), \mathbf{z}(t)) = 0 \quad \forall k \in \{1, \ldots, K'\}, t \in (-\epsilon, \epsilon) \tag{3.14}$$

$$\left[\sum_{i=1}^{d} \alpha_i(t) \nabla_{\mathbf{x}} f_i(\mathbf{x}(t), \mathbf{z}(t))\right] + \left[\sum_{k=1}^{K'} \beta_k(t) \nabla_{\mathbf{x}} g_k(\mathbf{x}(t), \mathbf{z}(t))\right] = 0 \quad \forall t \in (-\epsilon, \epsilon) \tag{3.15}$$

Denote with $h(t)$ the last line of the modified KKT conditions (Eqn. 3.15). We recall that at $t = 0$, we have $(\mathbf{x}(0), \mathbf{z}(0)) = (\mathbf{x}^*, \mathbf{z}^*)$, which is Pareto optimal by construction. Thus, there generically exist KKT dual variables $\boldsymbol{\alpha}(0) = \boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}(0) = \boldsymbol{\beta}^*$ such that $h(0) = 0$. We wish to find a first order approximation about $t = 0$ that preserves this condition locally, i.e. $h(t) = 0$ for $t \in (-\epsilon, \epsilon)$. Thus, we want to enforce $h'(t) = 0$ for $t \in (-\epsilon, \epsilon)$. Differentiating $h(t)$ and evaluating at $t = 0$ yields the following relation:

$$
\begin{aligned}
0 \equiv h'(0) &= D_x F^T(\mathbf{x}^*, \mathbf{z}^*)\boldsymbol{\alpha}'(0) + D_x G^T(\mathbf{x}^*, \mathbf{z}^*)\boldsymbol{\beta}'(0) \\
&\quad + \left[\sum_{i=1}^{d} \alpha_i^* H_{xx} f_i(\mathbf{x}^*, \mathbf{z}^*) + \sum_{k=1}^{K'} \beta_k^* H_{xx} g_k(\mathbf{x}^*, \mathbf{z}^*)\right]\mathbf{x}'(0) \\
&\quad + \left[\sum_{i=1}^{d} \alpha_i^* H_{xz} f_i(\mathbf{x}^*, \mathbf{z}^*) + \sum_{k=1}^{K'} \beta_k^* H_{xz} g_k(\mathbf{x}^*, \mathbf{z}^*)\right]\mathbf{z}'(0) \\
&=: D_x F^T \cdot \boldsymbol{\alpha}'(0) + D_x G^T \cdot \boldsymbol{\beta}'(0) + Hx \cdot \mathbf{x}'(0) + Hz \cdot \mathbf{z}'(0) \ .
\end{aligned}
$$

We also consider the additional constraints given by Eqn. 3.13 and Eqn. 3.14. The former simply requires that $\left(\sum \alpha_i = 1\right) \implies \left(\sum \alpha_i' = 0\right)$. For the latter, we differentiate with respect to t, and simplify:

$$
\begin{aligned}
0 &\equiv \beta_k'(t) g_k(\mathbf{x}(t), \mathbf{z}(t)) \\
&\quad + \beta_k(t)\Big[\nabla_{\mathbf{x}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{x}'(t) + \nabla_{\mathbf{z}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{z}'(t)\Big] && (3.16) \\
&\equiv \nabla_{\mathbf{x}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{x}'(t) + \nabla_{\mathbf{z}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{z}'(t) && (3.17)
\end{aligned}
$$

This simplification (from Eqn. 3.16 to Eqn. 3.17) follows because all constraints g_k with $k \in \{1, \ldots, K'\}$ are active for all $t \in (-\epsilon, \epsilon)$. Thus, $g_k(\mathbf{x}(t), \mathbf{z}(t)) = 0$ (causing the first term to vanish), and $\beta_k(t) \neq 0$. Stacking expression Eqn. 3.17 for all K' constraints g_k into a single matrix, and evaluating at 0, we have

$$
\mathbf{0} = [D_{\mathbf{x}} G, D_{\mathbf{z}} G] \cdot \begin{bmatrix} \mathbf{x}'(0) \\ \mathbf{z}'(0) \end{bmatrix}
$$

Combining all of the constraints above, it must be the case that

$$\mathbf{0} \equiv \underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_{\mathbf{x}}G & D_{\mathbf{z}}G \\ D_x F^T & D_x G^T & Hx & Hz \end{bmatrix}}_{\in \mathbb{R}^{(1+K'+D)} \times \mathbb{R}^{d+K'+D+C}} \begin{bmatrix} \boldsymbol{\alpha}'(0) \\ \boldsymbol{\beta}'(0) \\ \mathbf{x}'(0) \\ \mathbf{z}'(0) \end{bmatrix} \tag{3.18}$$

$$=: Mv \tag{3.19}$$

□

Chapter 4

Pareto Gamut Discovery

Our algorithm for Pareto gamut discovery is outlined in Alg. 1. The user provides the performance objective F and the constraints defining the augmented design space $\mathcal{X}$. We output a *patch array* P and an *augmented buffer* B (Section 4.1), which contain complementary representations of the discovered Pareto gamut. Each iteration of the discovery algorithm begins with a set of N_s samples from the augmented design space. Each sample is optimized within its fixed context, so that it satisfies the fixed-context KKT conditions (Proposition 3.1). Critically, the context value for each initial sample is chosen randomly within our range, which allows us to explore multiple contexts simultaneously. We then apply Proposition 3.3 to compute a first order approximation to the Pareto gamut about each optimized sample. The resulting $(d-1+C)$-dimensional patches are stored and sampled to provide seed points for the next iteration. Discovery continues in this manner until the Pareto gamut converges or until a user-specified computation budget is reached. Finally, our exploration tool visualizes this precomputed information. Below, we detail the data structures and algorithms used to uncover the Pareto gamut.

4.1 Data Structure

Our continuation scheme discovers contiguous manifolds along the Pareto gamut, each of which may span a wide range of performance and context values. We represent

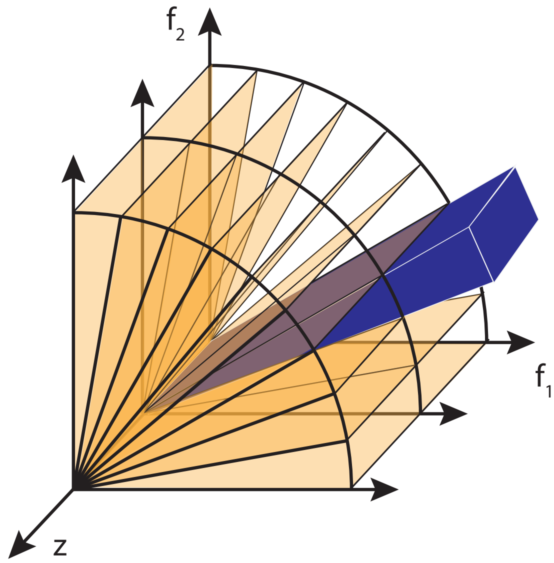

Figure 4-1: Visualization of the (hyper)cylindrical augmented buffer structure. The d performance dimensions are discretized using $d-1$ (hyper)spherical coordinates, and the C context dimensions are discretized using C Cartesian coordinates. The blue wedge indicates a single cell of the grid, which stores references to the patch points that fall in this region.

each manifold with a simple *patch struct*, which contains a unique patch ID and information about the patch in its continuous *and* discrete forms. For the continuous representation, we store (1) the locally-optimized center point $(\mathbf{x}^*, \mathbf{z}^*) \in \mathcal{X}$ and its image $F(\mathbf{x}^*, \mathbf{z}^*)$, and (2) the set of $d-1+C$ expansion directions that define the locally optimal manifold in augmented design space. We also use the patch struct to store (3) a set of discretized points (x, z) on this expanded patch (in augmented design space), along with the image $F(x, z)$ of each patch point in augmented performance space. The resulting patch struct is then appended to a *patch array*, which is a list containing information about all manifolds discovered so far.

Although the patch array is a natural way to store this information, it is not well-suited to measure coverage, convergence, or dominance efficiently. We address this with an acceleration structure: the *augmented buffer* (Fig. 4-1). As discussed in Section 3.1, any ray $\alpha \in \mathbb{R}^d_+$ from the origin will intersect the fixed-context Pareto front $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ at most once. This lets us discretize $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ using (hyper)spherical coordinates: the angle vector $\phi \in [0, \pi/2]^{d-1}$ indicates a performance trade-off (ray

ALGORITHM 1: Pareto gamut discovery

Input: Performance metrics F, augmented design space $\mathcal{X}$
Output: Augmented buffer B, Patch Array P
B:= Empty Augmented Buffer;
P:= Empty Patch Array;
repeat
  $(\mathbf{x}_s^0, \mathbf{z}_s^0), \ldots, (\mathbf{x}_s^{N_s}, \mathbf{z}_s^{N_s}) \leftarrow$ stochasticSampling(B, F, $\mathcal{X}$);
  for $i = 0 : N_s$ **do**
    $D(\mathbf{x}_s^i) \leftarrow$ selectFixedContextDirection(B, $\mathbf{x}_s^i$);
    $\mathbf{x}_o^i \leftarrow$ localFixedContextOptimization($D(\mathbf{x}_s^i)$, $F_{\mathbf{z}_s^i}$, $\mathcal{X}_{\mathbf{z}_s^i}$);
    $A^i \leftarrow$ firstOrderGamutApproximation($(\mathbf{x}_o^i, \mathbf{z}_s^i)$, F, $\mathcal{X}$);
    updateBufferAndPatchArray(B, P, $F(A^i)$);
  end
until *converged (see 4.2) or computation budget exceeded*;
return B, P;

direction), and the smallest radius r for which ϕ intersects the feasible performance space indicates the Pareto front candidate. Combining with the context variables $\mathbf{z}^*$, we use a (hyper)cylindrical discretization to represent our $(d-1+C)$-dimensional Pareto gamut. The context space $\mathcal{Z} \subset F(\mathcal{X})$ is discretized using C Cartesian coordinates, and the remaining $d-1$ dimensions form a (hyper)spherical discretization of the performance space. To judge convergence, coverage, and Pareto-optimality, the augmented buffer only requires each point's radius from its fixed-context origin. The buffer also retains each point's patch ID, to aid in a later step. We update the augmented buffer throughout the discovery process to reflect the best known solutions at each iteration.

4.2 Discovery Algorithm

We present a global/local discovery algorithm that consists of 3 simple steps: (1) we randomly select several seed points in joint context-design space; (2) we push each seed point toward its fixed-context Pareto front; and (3) we compute a first-order approximation about each optimized design point, to uncover a local neighborhood of Pareto optimal designs. This process is repeated until either the Pareto gamut converges or the permissible computation budget is exceeded.

Stochastic Sampling

To avoid local minima, each iteration begins with a user-specified number N_s of random samples $(\mathbf{x}^i, \mathbf{z}^i) \in \mathcal{X}$ for $i = 1 \dots N_s$. In the first iteration, these points are selected uniformly at random from $\mathcal{X}$. In subsequent iterations, for each of the N_s samples, we select a patch at random, traverse to a random point $\mathbf{x}^j$ on the patch, and select our new sample $(\mathbf{x}^i_s, \mathbf{z}^i_s)$ using the perturbative formula from Schulz et al. [2018, §6.2.2]. Notably, the context $\mathbf{z}$ is also perturbed in this process, encouraging thorough contextual exploration. All results are clamped to ensure that $(\mathbf{x}^i_s, \mathbf{z}^i_s) \in \mathcal{X}$.

Local Fixed-Context Optimization

After selecting our samples $(\mathbf{x}^i_s, \mathbf{z}^i_s)$, each one is optimized toward the Pareto front. Since contextual parameters are not optimization variables, this step considers a fixed-context problem $F_{\mathbf{z}^i_s}(\mathbf{x}^i_s; \mathbf{z}^i_s)$. Following Schulz et al. [2018], we select an optimization direction $D(x^i_s) \in F_{z^i_s}(\mathcal{X}_{z^i_s}) \subseteq \mathbb{R}^d$ based on the performance of $\mathbf{x}^i_s$, and optimize the scalarization $\|F_{\mathbf{z}^i_s}(\mathbf{x}; \mathbf{z}^i_s) - (F_{\mathbf{z}^i_s}(\mathbf{x}^i_s; \mathbf{z}^i_s) + D(\mathbf{x}^i_s))\|^2$ to obtain a local optimum $\mathbf{x}^i_o \in \mathcal{X}_{\mathbf{z}^i_s}$.

First-Order Gamut Approximation

For each locally-optimal $(\mathbf{x}^i_o, \mathbf{z}^i_s) \in \mathcal{X}$ produced using the optimization procedure in §4.2, we use Proposition 3.3 to extract a first-order approximation to the Pareto gamut, producing $d-1+C$ directions $\mathbf{v}^i = (\mathbf{x}', \mathbf{z}')$ (see Section 3.2.4). These directions span an affine subspace along which we can walk infinitesimally without leaving the Pareto gamut. We orthonormalize all $\mathbf{v}^i$, and find the maximal t for which $v^i_t := (\mathbf{x}^i_o, \mathbf{z}^i_s) + t\mathbf{v}^i \in \mathcal{X}$. The cross product of all v^i_t yields a (hyper)rectangular patch $A^i \subseteq \mathbb{R}^D \times \mathbb{R}^C$.

To sample $F(A^i)$ in performance space, we discretize A^i as a uniform grid (at a user-specified step size) and map each vertex by evaluating F. Any vertices $(\mathbf{x}, \mathbf{z}) \notin \mathcal{X}$ are discarded before evaluation. Because A^i is connected and F is continuous, the image $F(A^i)$ should be connected. However, the sampled points may skip over cells of the augmented buffer due to undersampling. We correct this with a projection step.

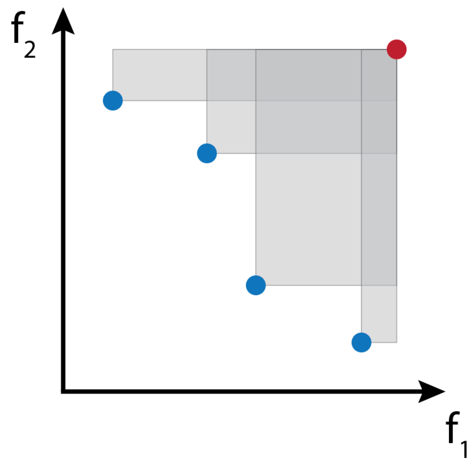

Figure 4-2: The *hypervolume* (gray) of a candidate Pareto front (blue) is the Lebesgue measure of the area strictly dominated by the points on the front, relative to a "worst case" *nadir point* (red). Hypervolume is strictly monotonically increasing w.r.t. Pareto optimality: a strictly dominant Pareto front will always have a larger hypervolume.

To promote the likelihood that the current patch contributes a sample to each cell intersecting $F(A^i)$, we lift a fixed triangulation of the grid in A^i to triangulate our (hyper)cylindrical augmented performance points in design space. We use this triangulation to compute $(d+C)$-dimensional barycentric coordinates for every buffer cell within the boundary of $F(A^i)$. We cannot directly interpolate along our approximation of $F(A^i)$, as this may introduce infeasible points when the gamut is non-convex. Instead, we use the barycentric coordinates to interpolate along A^i; then, we map the interpolated point(s) back to performance space for insertion into the buffer and patch array.

Gamut Convergence

To measure convergence, we must be able to measure the quality of our Pareto gamut estimation over time. For fixed-context optimizations, the quality of a candidate front is often measured using *hypervolume* [Bader and Zitzler, 2011]. Hypervolume measures the area of performance space that is strictly dominated by the candidate Pareto front points, relative to a *nadir point* – the upper bound of all functions (see Fig. 4-2). As new or improved points are found along the front, hypervolume increases. Thus, fixed-context approaches declare convergence when the *hypervolume improvement* falls below some threshold δ for N consecutive iterations.

To extend this to the Pareto gamut, we compute a hypervolume vector V^t, where V_i^t is the hypervolume of the i^{th} fixed-context Pareto front after iteration t. We consider our gamut converged if the hypervolume improvement $(V^t - V^{t-1})(V^t - V^{t-1})^T < \delta$ for N consecutive iterations. For all examples in Section 5, we set $\delta = 10^{-3}$ and $N = 3$; our nadir point is $[1]^d$, due to our normalized metrics.

Pareto Gamut Extraction

Once the gamut has converged, each buffer cell contains the Pareto-dominant point along its respective scalarization direction ϕ. For nonconvex problems, however, being the closest point along a ray to the origin is a necessary but not sufficient condition for Pareto optimality. Hence, for each fixed context in the buffer discretization, we extract $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ according to Definition 3.4.

This set of points (at most one per buffer cell) is often sufficiently dense to capture the Pareto gamut. However, discretization artifacts can cause our gamut to appear sparse in certain scenarios. For example, if the Pareto gamut is asymptotic to one or more of the augmented performance axes (as in Fig. 1-3), the buffer only provides a sparse sampling in this region by default. To address this issue without resorting to a fine discretization, we use our patch array to fill gaps in the Pareto gamut after extraction.

This procedure is based on the expectation that the true Pareto gamut contains large contiguous regions of the patches we uncover during our discovery process. Consider the set of points in the extracted Pareto gamut that belong to a given patch A^i, and let I be the pre-image of these points. Assuming that our expectation holds, the boundary of the point set I denotes the maximal region of the patch A^i (in augmented design space) that maps to the Pareto gamut. Then, all patch samples $(\mathbf{x}, \mathbf{z}) \in A^i$ within this boundary should be eligible for inclusion in the Pareto gamut. A dense sampling of this patch region is contained in the patch array, but the buffer (from which we extract the Pareto gamut) only admits one sample per A^i per cell. Thus, the vast majority of discretized patch samples are never included in the buffer or considered as candidates for the extracted Pareto gamut. By incorporating these samples directly through the post-processing step described above, we are able to drastically improve our coverage of the final Pareto gamut without incurring significant additional cost. This process may introduce dominated points, but this can be addressed by running a final Pareto optimality check for each fixed context before returning.

Chapter 5

Results

We test our algorithm's performance on two types of problems. First, we adapt analytic benchmarks to a contextual setting to validate correctness. Our subsequent examples in engineering design serve as case studies to explore the utility of our approach and of the Pareto gamut more broadly. For all problems, we normalize the parameters and metrics such that $\mathbf{x} \in [0,1]^D$, $\mathbf{z} \in [0,1]^C$, and $f_i \in [0,1]\ \forall i \in \{1 \ldots d\}$.

5.1 Analytic Results

Since the Pareto gamut is a new construction, there are no established benchmarks to verify the accuracy of our discovery algorithm. We adapt several fixed-context benchmarks to a contextual setting, providing a test suite for algorithms designed to uncover Pareto gamuts. We also derive two methods to verify the correctness of our discovered gamut: (1) comparison against analytic solutions for the Pareto set at any given context $\mathbf{z}^*$, and (2) validation against Proposition 3.2 (Lower Envelope Optimality), which only requires computing a Pareto front rather than a Pareto gamut.

Modified ZDT Functions The ZDT test suite [Zitzler et al., 2000] contains 5 continuous-domain problems for multi-objective optimization. The functions are designed to stress-test discovery algorithms, with challenging features like non-convexity,

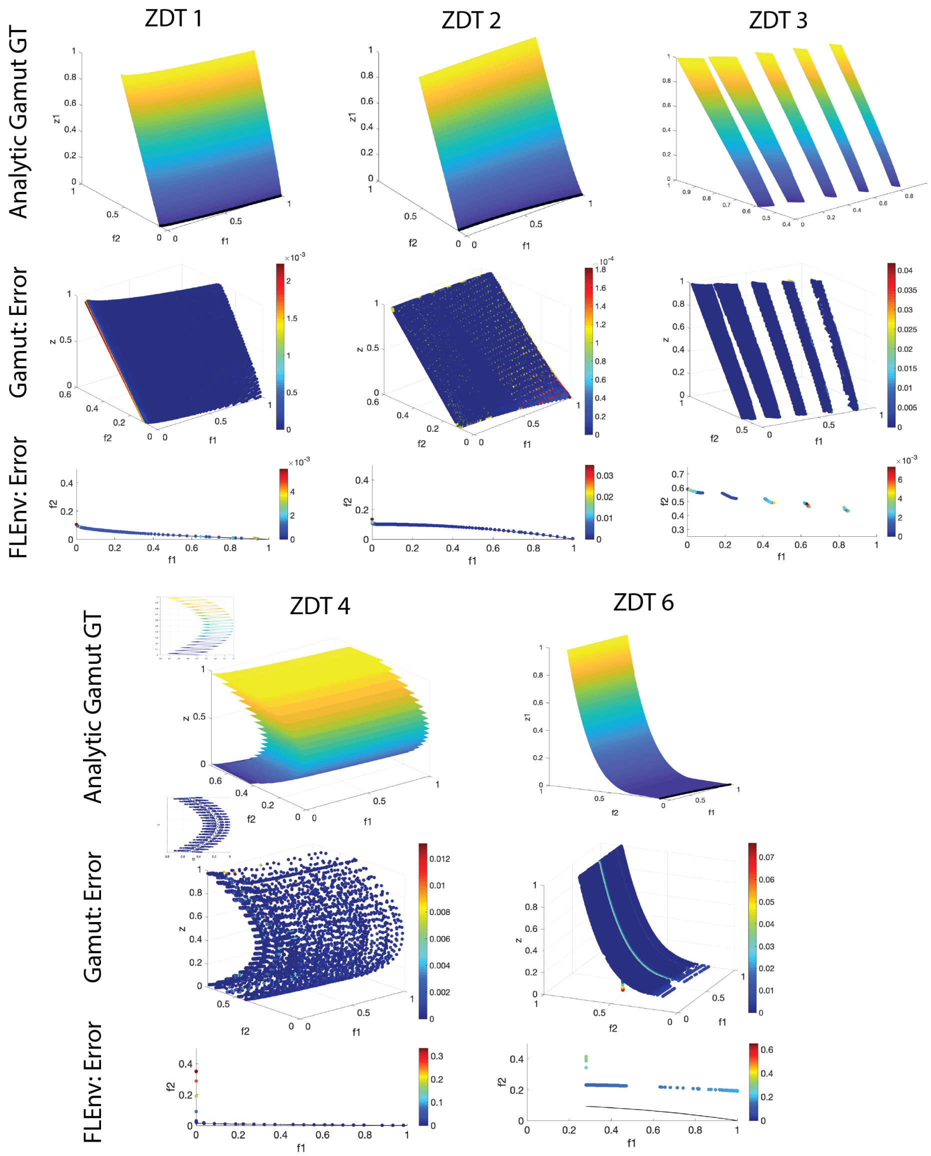

Figure 5-1: Our suite of contextual ZDT solutions. **(Top)** Analytic ground truth for the Pareto gamut, with $d = 2$ and $C = 1$. Colored by context value. **(Middle)** Our discovered gamut, colored by distance to the ground truth. **(Bottom)** Flattened lower envelope of our solution (scatter); colored by distance to the ground truth Pareto front of the corresponding free-context problem(black line). All discovered solutions use 200 cells along each dimension of the buffer.

multi-modality, and sparse solution density at the front. Each ZDT function has $d = 2$ performance metrics that depend on m design variables $\boldsymbol{y} = (y_1, \ldots, y_m)$. In the original problem, there are no contextual parameters ($C = 0$). To test $C \geq 1$, we treat some of the m design variables as context parameters. We let $m = D + C$ and define

$F(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \rightarrow \mathbb{R}^d \times \mathbb{R}^C$. We preserve the original function definitions with respect to $\hat{\boldsymbol{y}} = (x_1, \ldots, x_D, z_1, \ldots z_C) \in \mathbb{R}^m$. The only difference is our treatment of the contextual parameters $\mathbf{z}$, whose values will be explored rather than optimized. Fig. 5-1 (middle) shows strong agreement between our discovered Pareto gamut and the analytic ground truth.

By construction, the corresponding free-context problem $H_F(\boldsymbol{y})$ is the original ZDT problem over m design variables, and its Pareto front $H_F(\mathcal{P})$ is the well-established ZDT solution. By Proposition 3.2, our Pareto gamut's flattened lower envelope must also match this established solution. Fig. 5-1 (bottom) illustrates our method's validity in this regard.

Our choice of buffer discretization (with 200 cells in each dimension) results in a smooth, complete approximation when the Pareto front changes slowly as a function of the context (ZDT 1,2,3, and 6). ZDT 4 appears much more sparse, because the front oscillates quickly as z changes. Thus, when we check the fixed-context cells for Pareto dominance, the single best fixed-context value eclipses many other fronts. This can be resolved by discretizing the context more finely. The discrepancy in the lower envelope for ZDT 6 is also a discretization artifact, because the solution density grows exponentially smaller near the flattened lower envelope – thus, the fixed context at our smallest cell center ($z = 0.0025$) is still quite far from the ground truth envelope. Because we have identified the correct solution manifold, this issue is resolved by generating buffer samples along the exact context boundary ($z = 0$).

Fourier Benchmark The ZDT functions are helpful for validation against an established benchmark. However, the solution $\mathcal{P}$ is a single linear manifold in all 5 cases. We stress test our algorithm with a contextual Fourier benchmark, for which $\mathcal{P}$ is non-linear and disconnected (Fig. 5-2). Each objective function $f_i(\mathbf{x}, \mathbf{z})$ is

$$f_i(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{D} \sum_{j=1}^{N} \left[\alpha_{ij} \cos(3jx_i + \beta_{ij}^{2z_1}) \right] + \sum_{m=1}^{C} \sum_{n=1}^{N} \left[\gamma_{mn} \cos(3nz_k + \mu_{mn}^{2z_1}) \right] ,$$

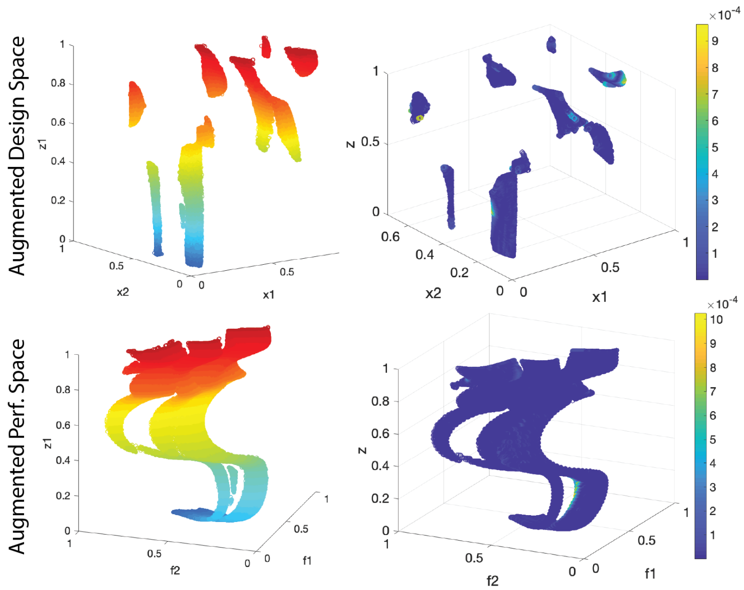

Figure 5-2: Fourier benchmark solution. **(Left)** Our discovered Pareto gamut (bottom) and its pre-image (top), colored by context value. **(Right)** Error between our gamut/pre-image and 100 precomputed fixed-context solutions. Strong agreement highlights our algorithm's empirical convergence, even in the case of a highly disconnected, non-linear pre-image (in augmented design space). Our solution also identifies an additional optimal region that is overlooked by the fixed-context approach.

where N is the desired polynomial order, and $\alpha_{ij}, \gamma_{mn} \in [0, 1]$ and $\beta_{ij}, \mu_{mn} \in [0, \pi]$ are randomly chosen coefficients.

5.2 Engineering Design Examples

We now present several engineering examples to demonstrate the potential applications and utility of contextual exploration.

Lamp Fig. 5-3 features a parametric lamp with $D = 21$ design variables controlling the length and orientation of each rod. Performance is measured with $d = 3$ metrics: mass, instability (L_2 distance between the stand's center and the lamp's projected center of mass), and illumination of a pre-determined focal point $p \in \mathbb{R}^3$ (average distance of the lamp's bulbs to p). The location of p is a natural context parameter, as the relative position between the lamp and the focal point changes for each application. By computing the Pareto gamut over p, designers can experiment freely with

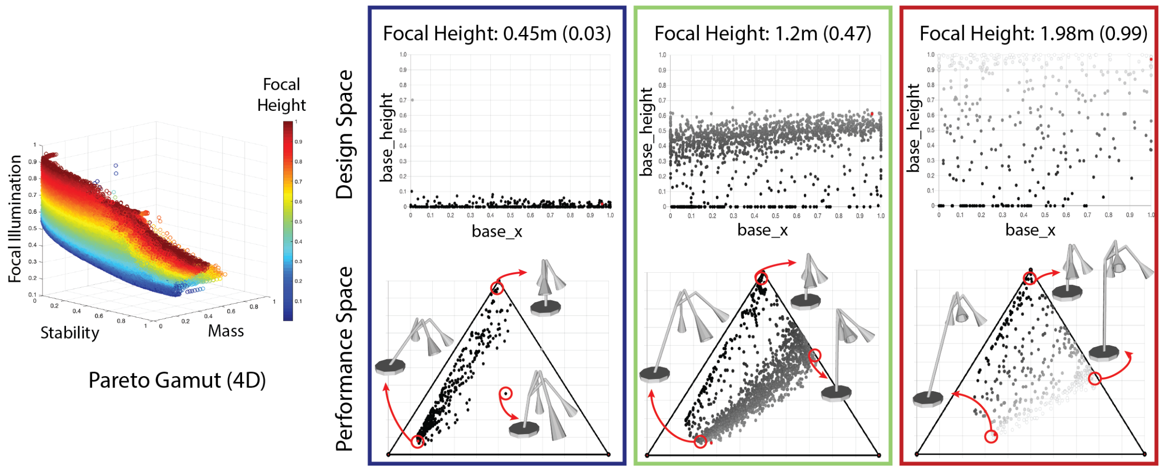

Figure 5-3: **(Left)** 4D Pareto gamut for our lamp example: performance values are plotted spatially, and color represents the context value. **(Right)** Fixed context Pareto sets (top) and fronts (bottom), where grayscale value indicates the height of the lamp's central rod. The Pareto set visualizes 2 (out of 21) design variables, including the height of the lamp's primary rod (vertical axis). As expected, a larger height range is used as the focal height increases. The Pareto fronts (each 3D) are embedded in a 2D space using barycentric coordinates. Each vertex represents the *worst* focal illumination (top corner), stability (lower left) and mass (lower right), respectively. The optimal designs w.r.t. each metric are located near the opposite edge. The sampling density in this example is lower to accomodate the higher dimensional (4D) space.

their layout, while having immediate access to the appropriate Pareto front. We consider the focal point height, p_z as our context. As expected, the Pareto set includes increasingly taller lamps as p_z increases. There are also many designs that persist throughout all contexts, such as the lightweight, stable design shown in Fig. 5-3. Our method propagates these optimal designs to *all* relevant contexts upon discovery in *some* context; this improves consistency across contexts, because points do not have to be discovered independently on each front.

Turbine Fig. 1-3 and Fig. 5-4 illustrate a simple blade design for a wind turbine with $D = 3$ design parameters: turbine radius r (central shaft to blade tip), turbine height h (perpendicular to the plane of rotation), and blade pitch angle α. The performance metrics are mass and power output, $P = \frac{1}{2} C_p \rho \pi r^2 v^3$, where ρ is air density, and v is wind speed. The coefficient of performance C_p is an empirically measured function

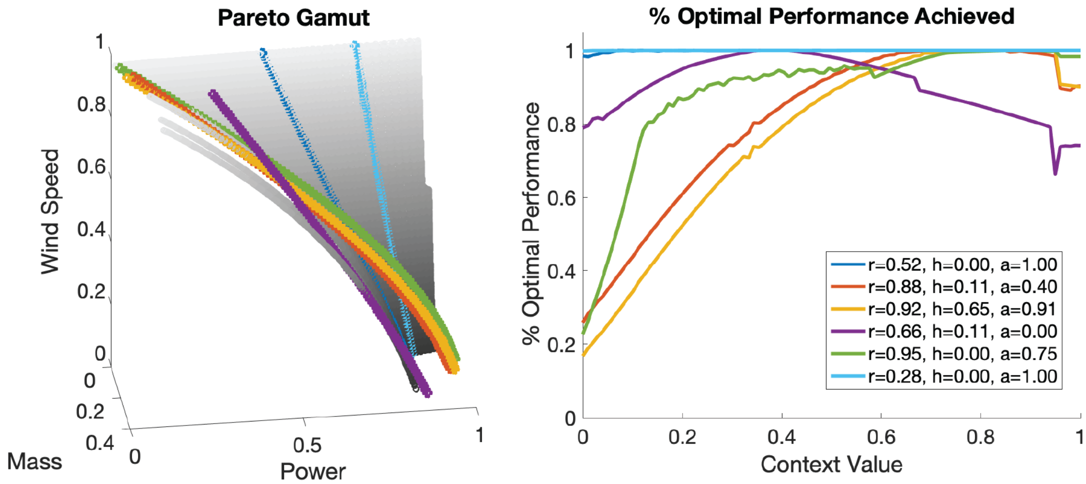

Figure 5-4: **(Left)** After computing the Pareto gamut (gray) for turbines over multiple wind speeds (see Fig. 1-3), engineers are able to visualize the performance of a particular design (colored line) relative to the achievable optimal performance in any context. **(Right)** Numerical performance curves for each individual design over the context space, providing additional insight about each design's desirability over the context range.

of the turbine's geometry and environmental context; we approximate C_p with data from Johnson [2006, Figure 9].

We use v (wind speed) as our contextual parameter, to explore the optimal designs over any wind speed that may occur at the turbine's installation site. The pre-image of our Pareto gamut shows that the Pareto set is highly dependent on the wind speed: turbine power only benefits from small and medium radii at low wind speeds, but the ceiling on useful radius values increases along with wind speed.

This presents a challenge for engineers, because the wind speed is dynamic; thus, it is critical to identify a final design that performs well with respect to a multitude of contexts. As shown in Fig. 5-4, the Pareto gamut aids this process by highlighting designs that are particularly strong (or weak) across many contexts. For each query design $q \in \mathbb{R}^D$, we compute $F_{\mathbf{z}^*}(q)\ \forall \mathbf{z}^* \in \mathcal{Z}$, and plot these performance values relative to the Pareto gamut for qualitative inspection. We also quantify the performance "sacrifice" that q incurs at each context, by finding the closest point $q^* \in F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ on the fixed-context slice of the gamut, then computing the ratio of their hypervolumes, $\mathrm{hv}(q)/\mathrm{hv}(q^*)$. This serves as a proxy for q's performance as a percentage of the optimal achievable value at this context.

Armed with this information, an engineer can make more informed decisions about the final dynamic-context design. For a site with wide variation in wind speeds, the blue designs may be suitable, as they are both Pareto-optimal throughout the full context range. However, in an area with characteristically high wind, the green curve may be preferable due to its near-optimality everywhere, and superior raw power output in the high-wind range.

On the surface, this discussion of characteristic wind speeds may seem reminiscent of the approach discussed in Section 1, which addressed dynamic contexts by computing the Pareto optimal designs corresponding to a single "representative" context. However, the Pareto gamut provides fundamentally different information. For example, consider a turbine site with reliably slow wind speeds, in the range of 4.4-5.2 m/s (normalized context value $z = [0.2, 0.6]$). By computing the Pareto optimal designs for an average context of $z = 0.4$ and selecting the design with maximal power, you would arrive at the purple solution. This solution is indeed optimal at the average context, but its performance relative to the achievable optima falls off dramatically to either side of the original query. Thus, the turbine will perform sub-optimally a large majority of the time. This sensitivity is only well-exposed when comparing the design to the Pareto gamut, emphasizing its ability to provide unique insights that were not readily observable with existing methods.

Bicycle Rocker and Seat Stay Fig. 5-5 shows two adjacent elements of a full suspension mountain bicycle: the rocker (triangular pivoting frame) and the seat stay (incident rod). Both the rocker and the stay have 3 design variables and 2 performance metrics (mass and approximate compliance).

In a typical engineering workflow, each part would be optimized independently after the shared pivot location is fixed by an engineer. However, many pivot locations may be explored before settling on a final design. Our method naturally facilitates this exploration by treating the shared pivot point location as a context variable for each part.

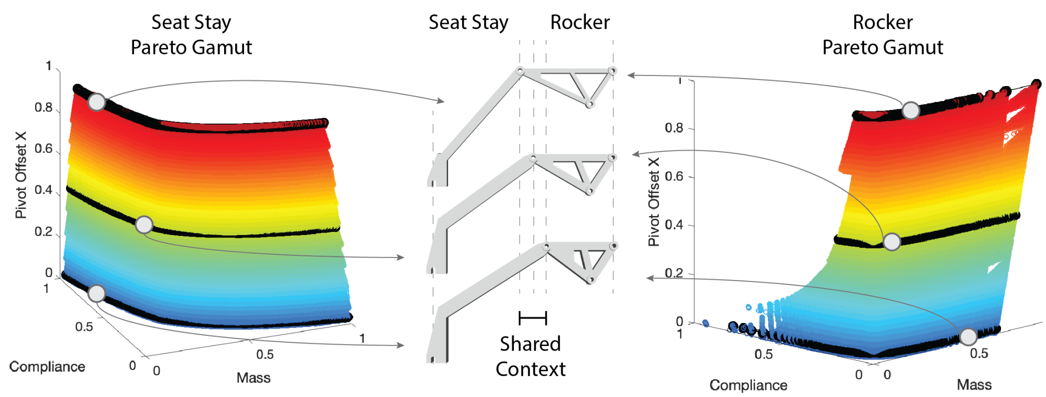

Figure 5-5: Pareto gamuts (left, right) for two adjacent elements of a bicycle suspension system (center, see Fig. 1-2 for full assembly). The context parameter for each part is the location of their mutual pivot point. The Pareto gamut is computed independently for each part. However, the shared contextual parameter permits higher-level design optimization, by allowing the engineer to explore different locations for the pivot point, while getting immediate feedback about the achievable performance trade-offs for each affected part.

In particular, we discover the Pareto gamut for each element independently. Then, when viewing the assembly, we control the context for both parts with a single slider, such that they are always aligned and compatible. As the context is updated, the engineer is able to pursue the appropriate Pareto fronts for both parts in real time. This provides valuable insight about the ripple effects of a given interface decision, allowing the engineer to engage in higher-level design optimization than is typically possible.

Thus, our framework serves as a stepping stone toward hierarchical optimization: context parameters allow engineers to decompose an optimization problem into tractable, intuitive pieces linked by adjustable constraints. By restricting to Pareto gamuts before moving up the hierarchy, we also mitigate the curse of dimensionality that would otherwise plague a complex, many-objective design problem of this nature.

Bicopter Controller We also investigate our method's applicability to other domains, such as the co-optimization of robot geometry and control [Spielberg et al., 2017]. We consider a simple 2D bicopter with two common performance metrics:

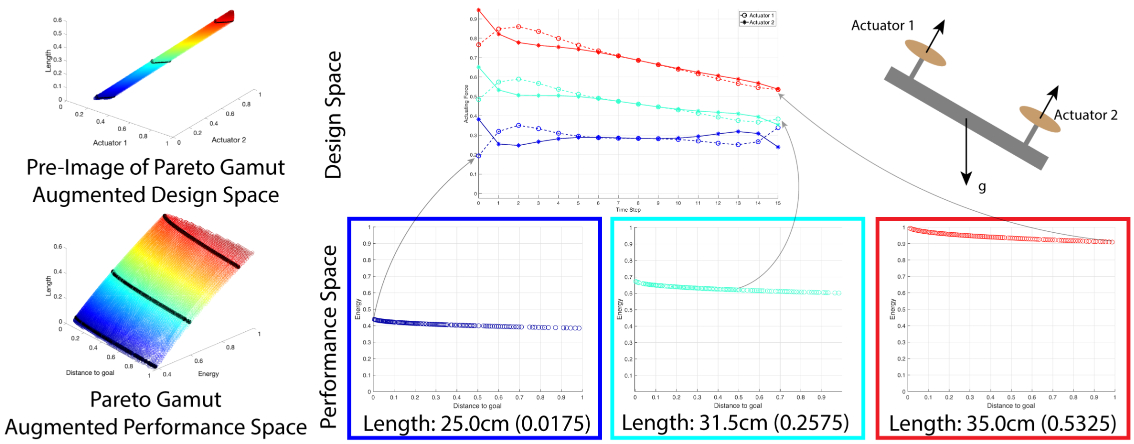

Figure 5-6: **Left:** Pareto gamut for the bicopter controller design problem. The augmented design space shows the average actuating forces over the $t = 16$ time steps for each of the two actuators. **Right:** Pareto sets and fronts for three fixed-contexts. We select a single point in each fixed-context Pareto set, and visualize both propellers' actuation over time. **Top Right:** Force analysis of the bicopter. As expected, the copter is able to reach the goal at any length, but the required actuating forces and energy cost increase with bicopter length. Even the lowest energy of the copter at the length of 35.0cm (the sequence in red) is larger than the biggest energy of the copter at the length of 25.0cm (the sequence in blue).

ability to reach a target, and overall energy usage [Tedrake, 2020]. We have $D = 2N$ design parameters that denote the actuation sequence for each of the two propellers over N time steps (we set $N = 16$). We consider the length of the bicopter, ℓ, as our context parameter. Thus, our Pareto gamut encodes the optimal controller trade-offs over an entire range of fixed robot geometries.

As shown in Fig. 5-6, the Pareto-optimal controllers become more energy-intensive as ℓ increases. This is expected, because the bicopter's mass increases with ℓ, and heavier copters require more energy. The available actuation energy is sufficient to reach the goal in all cases, so the achievable performance range is unhindered.

This problem has a fundamentally different structure than our previous examples, due to the highly dependent design variables. Performance is also sensitive to small changes in actuation, especially as small changes accumulate over our large time steps. We overcome these challenges by seeding our exploration algorithm with *anchor* points that are obtained via coarse optimization over the two performance metrics. The inclusion of anchor points encourages our discovery algorithm to explore feasible regions of the design space, which considerably reduces the accumulation-based issues. Our success in this initial experiment highlights the Pareto gamut's promise as a tool that could generalize to several problem domains.

5.2.1 Performance Benchmarks

For each of the four engineering examples, we benchmark our algorithm's performance in terms of the evaluation time (in seconds), the number of performance metric evaluations required, and the final quality (hypervolume) of each fixed-context Pareto front within our gamut. The results for this benchmark are shown in Table 5.1.

We also compare our performance against that of two state-of-the-art fixed-context approaches: the closely-related approach from Schulz et al. [2018] and the popular evolutionary algorithm, NSGA-II [Deb et al., 2002]. To approximate the gamut with these algorithms, we independently compute the Pareto fronts at N evenly-spaced fixed-context values $\mathbf{z} \in \mathcal{Z}$.

As shown in Table 5.1, the fixed-context hypervolumes are consistent across all methods, validating the quality of the fixed-context Pareto fronts extracted from our gamut. Moreover, our method consistently outperforms with respect to average time and average function evaluations per context. For the one case where this does not hold (bicopter, # evals), our cost is only marginally higher than NSGA, for significantly better hypervolume.

We evaluated this claim further by conducting an extensive set of tests on the Turbine example, as shown in Fig. 5-7. Both of the previous works require roughly an order of magnitude more function evaluations to achieve the same information density as our Pareto gamut with 200 context cells in the buffer (which is roughly equivalent to $N = 200$ evenly-spaced fixed-context Pareto fronts). This figure also highlights the power of our method's amortization across contexts: computing the full gamut becomes less expensive than individual front computation as soon as the user seeks more than 10 distinct contexts on average, or 35 in the worst observed case. This inflection point is low enough to offer immediate relevance in the design community, as engineers are already iterating through this many optimization processes to test

	Fixed Context Hyper-volume					#Function Evals	Time	Avg Time per Context†	Avg Evals per Context†
	Turbine ($D = 3, d = 2, C = 1$)								
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9				
NSGA	0.18	0.27	0.40	0.58	0.81	61,000	8.75s	1.75s	12,200
Schulz	0.18	0.27	0.40	0.58	0.81	44,538	37.9s	7.58s	8,908
Ours	0.16	0.26	0.37	0.68	0.83	124,713	52.1s	**0.26s**	**624**
	Lamp ($D = 21, d = 3, C = 1$)								
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9				
NSGA	0.57	0.54	0.46	0.35	0.23	5,000,000††	2320.8s	464.2s	1,000,000††
Schulz	0.55	0.52	0.46	0.34	0.22	251,692	7495.8s	1499.2s	50,338
Ours	0.58	0.55	0.47	0.35	0.21	349,839	8983.8s	**44.9s**	**1,749**
	BikeRocker ($D = 3, d = 2, C = 1$)								
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9				
NSGA	0.97	0.96	0.95	0.94	0.93	69,400	9.97s	1.99s	13,880
Schulz	0.96	0.91	0.95	0.94	0.93	215,106	41.4s	8.28s	43,021
Ours	0.96	0.95	0.94	0.94	0.93	2,583,066	310s	**1.55s**	**12,915**
	Bicopter ($D = 32, d = 2, C = 1$)								
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9				
NSGA	0.57	0.38	0.13	0.18	0.19	31,850	95.20s	19.04s	**6,370**
Schulz	0.61	0.52	0.43	0.33	0.22	109,103	228.0s	45.60s	21,821
Ours	0.61	0.52	0.43	0.33	0.22	1,574,358	299.8s	**1.50s**	7,872

† Divide corresponding total (time/evals) by 5 for NSGA-II and Schulz (number of fixed contexts computed); divide by 200 for our method (number of buffer cells along context dimension). †† NSGA-II is capped at $1,000,000$ function evaluations per fixed context.

Table 5.1: Performance comparison illustrating our method's amortized advantage over two state-of-the-art fixed-context approaches: Schulz et al. [2018] and NSGA-II [Deb et al., 2002]. We compare one run of our algorithm (yielding the Pareto gamut) to 5 runs of Schulz/NSGA (each yielding a single fixed-context front, at evenly spaced context values). **Our algorithm provides 40x denser coverage of the context space without sacrificing proportional efficiency/effectiveness**. We use an official Python benchmark [Blank and Deb, 2020] for NSGA-II, and a MATLAB implementation for Schulz et al. and ours. All experiments were run on an Intel Core i7 CPU. Reported results are averaged over 10 experimental runs, except the *Lamp*, which was computed only once due to the high time cost.

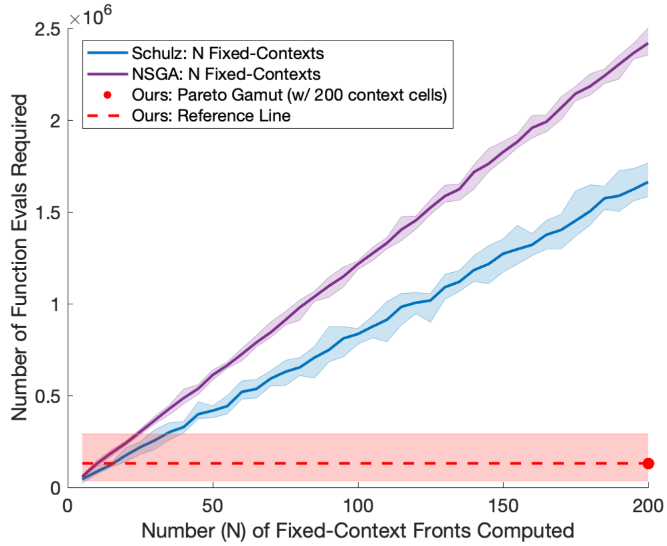

Figure 5-7: Number of function evaluations needed to compute N evenly-spaced fixed-context Pareto fronts with prior work (blue, purple), versus the number of evaluations required to compute the Pareto gamut with 200 context cells (red dot). The dashed red line is a reference, used to emphasize the critical point ($N = 10$, $evals \approx 10^5$) at which computing the full Pareto gamut becomes more efficient than repeated fixed-context computations with either method. Each experiment was run 10 times to mitigate the effects of randomness; lines represent the mean value of all 10 trials, while the shaded areas indicate the minimum and maximum cost for a given trial. Computed for the Turbine example.

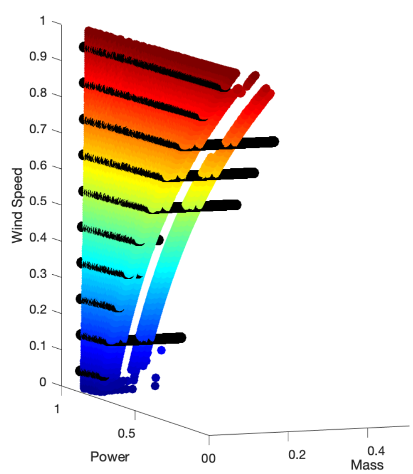

Figure 5-8: Pareto gamut density achieved by our method (rainbow, colored by context value) vs. Schulz et al. [2018] (black curves), assuming a comparable number of function evaluations ($\sim 10^5$) on the Turbine example. Schulz et al. [2018] is able to compute 10 fixed context solutions within this budget, which aligns with the methods' critical point shown in Fig. 5-7.

the effect of context in an ad-hoc manner.

Overall, these benchmarks highlight the promise of Pareto gamuts for practical engineering design. Our algorithm yields considerably denser coverage of the context range, while maintaining the quality of individual fixed-context Pareto fronts and incurring comparatively little additional computational cost. Coupled with the unique insights and opportunities discussed in Section 5.2, these results posit the Pareto gamut as a strong contender for the task of practical variable-context design optimization.

Chapter 6

Discussion and Limitations

The Pareto gamut opens many new possibilities for contextual exploration, but it is not without its challenges. The relative scaling between performance metrics can be problematic with contextual parameters, especially if the context can drastically alter the range of achievable values. Consider the turbine's power metric: with the physically common wind range of $v \in [2, 20]$m/s, the performance over the entire augmented design space ranged from -8×10^6 to 6×10^6. These are the factors we must normalize by in order to use our data structure. However, since power scales cubically with the context variable (windspeed), the range of achievable values at any particular context varies dramatically. In particular, the Pareto fronts at low wind speeds are confined to a small fraction of the available performance space – and thus, a small fraction of discretized buffer cells. This condensed front can create the impression that no tradeoffs exist at lower wind speeds, when in reality, the tradeoffs have just been muddled within our discretization. In this paper, we circumvented the issue by limiting our wind speeds to $v \in [4, 6]$m/s, so the discrepancy in achievable values was not detrimentally large. It would be worthwhile to explore this further, to increase the range of contextual parameters we can consider, while preserving the integrity of both the individual fixed-context Pareto fronts, and the relative change in performance as a function of the context.

The contextual parameters can also affect the design space in a non-trivial manner. For instance, to preserve the integrity of an L-bracket design as the angle shrinks (see

Fig. 1-2), the maximum thickness must be reduced, and the minimum length must increase. In this paper, we restricted the design space to the largest range that was admissible over *all* contexts. This may force the design space to be more restricted than necessary, which could conceal some optimal designs. This assumption was made for simplicity: it is not strictly necessary, as our theory permits linear and non-linear constraints on the design variables. Thus, the user could go beyond box constraints if desired. However, codifying the permissible range for each context can be challenging. Future work could improve our pipeline by automatically discovering the feasible design space across varying contexts.

Although our theory extends to any number of design parameters, contextual parameters, and performance metrics, the discovery and exploration of the full Pareto gamut may be limited to low-dimensional augmented performance spaces. Multi-objective optimization already struggles with the curse of dimensionality, as the potential surface area of each fixed-context Pareto front (and thus, the number of points required to approximate the front) grows exponentially in the number of performance metrics. For so called *many*-objective optimization problems (with $d \geq 4$), full front approximation is considered intractable [Schütze et al., 2019]. Our contextual variables inherently increase the dimension of the solution space, which implies that the full Pareto gamut would only be tractable for $d + C < 4$. This observation is supported by our Lamp example (with $d + C = 4$), which incurred a high computational cost even to obtain a perceptually sparse approximation of the Pareto gamut. For $d + C \geq 4$, it may be interesting (and necessary) to build an interactive optimization method for the discovery and traversal of the Pareto gamut. Schütze et al. [2019] proposed an interactive method that hinges on the continuation scheme of Martín and Schütze [2018] for fixed-context many-objective optimization. Since our first-order approximation generalizes their predictor step to include the effect of contextual parameters, our method could be adapted to a similar interactive exploration scheme for Pareto gamuts.

Chapter 7

Conclusion

To discover effective engineering solutions, it is critical to consider potential designs' performance with respect to multiple metrics *and* contextual settings. Existing multi-objective optimization schemes have made great progress with respect to the former goal, but few previous works have considered the latter. Instead, engineers have been forced to conduct sparse contextual exploration on their own, by running the optimization independently on a few selected or representative contexts. This process heavily depends on the engineers' existing intuition about the problem, which makes it difficult to obtain sufficiently insightful coverage and may prohibit novel discoveries that diverge from known intuition. By relegating contextual exploration to a secondary process, existing methods dramatically understate its impact on effective engineering design. Our work is the first to rectify these issues, by establishing contextual exploration as an equally primary concern for practical engineering design. We do this by incorporating contexts into the multi-objective optimization process, in concept and in practice.

At a conceptual level, our work codifies a precise language for multi-context multi-objective optimization through new objects like contextual parameters, augmented spaces, and the Pareto gamut. This language offers a common framework for engineering design, systematizing engineers' previously ad hoc practices for the exploration of e.g. mechanical boundary conditions, assembly constraints, and environmental factors.

To efficiently discover the Pareto gamut, we also develop a global/local optimization algorithm that hinges on our first-order approximation of the Pareto gamut about a known-optimal center point. This approximation generalizes previous continuation schemes by providing insight about both (1) the local Pareto front within the center point's fixed context, and (2) the way in which that Pareto front changes as a function of the context parameters. We facilitate exploration of the resulting gamut through several simple tools and visualizations. Lastly, we formulate a series of analytic and practical design problems for multi-context multi-objective optimization. By extending established analytical benchmarks to a contextual setting, we ground the problem in relation to previous works on fixed-context multi-objective optimization, while providing a baseline for future works on multi-context analysis. Most importantly, our suite of engineering design examples demonstrates the feasibility and utility of the Pareto gamut with respect to real-world engineering problems. Through dense coverage, our approach was able to assist with a variety of contextual needs, including (1) efficient adaptation of parts to many distinct environment specifications [L-bracket, bicycle lugs], (2) higher-level performance analysis for parts in dynamic environments [wind turbine, mounting bracket], and (3) hierarchical co-optimization of adjoining parts in an assembly [bicycle rocker and stay]. Each of these scenarios presents a unique, impactful, and currently under-explored research avenue that arose naturally from our general investigation of context parameters. This underscores the significance of our new framework, which will hopefully inspire efforts for higher-order reasoning on optimization results (as discussed in scenarios 2 and 3).

Even without these long-term advancements, our unified variable-context framework enables the simple, thorough contextual exploration that has traditionally proven elusive. Our algorithm's efficient, dense, and automatic contextual exploration will permit more informed design decisions by simultaneously increasing the amount of available information and freeing engineers' time and energy for their decision making tasks. Thus, our work serves as a critical first step toward comprehensive design optimization for practical engineering pursuits.

Bibliography

Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011. doi: 10.1162/EVCO_a_00009.

Gaurav Bharaj, David I. W. Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. Computational design of metallophone contact sounds. *ACM Trans. Graph.*, 34(6):223:1–223:13, October 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818108. URL `http://doi.acm.org/10.1145/2816795.2818108`.

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics*, 29(4):63:1–63:10, July 2010. ISSN 0730-0301.

Julian Blank and Kalyanmoy Deb. pymoo: Multi-objective optimization in python, 2020.

Michael Bogomonly. Partner spotlight: Using generative design with onshape and paramatters, June 2019. URL `https://www.onshape.com/cad-blog/partner-spotlight-using-generative-design-with-onshape-and-paramatters`.

Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. An asymptotic numerical method for inverse elastic shape design. *ACM Trans. Graph.*, 33(4):95:1–95:11, July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601189. URL `http://doi.acm.org/10.1145/2601097.2601189`.

Jin-Hee Cho, Yating Wang, Ing-Ray Chen, Kevin S. Chan, and Ananthram Swami. A survey on modeling and optimizing multi-objective systems. *IEEE Communications Surveys & Tutorials*, 19(3):1867–1901, 2017. doi: 10.1109/comst.2017.2698366. URL `https://doi.org/10.1109%2Fcomst.2017.2698366`.

Derek Covill, Steven Begg, Eddy Elton, Mark Milne, Richard Morris, and Tim Katz. Parametric finite element analysis of bicycle frame geometries. *Procedia Engineering*, 72:441 – 446, 2014. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2014.06.077. URL `http://www.sciencedirect.com/science/article/pii/S1877705814005931`. The Engineering of Sport 10.

Indraneel Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197, April 2002. ISSN 1089-778X.

Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part 1: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, August 2014. ISSN 1941-0026. doi: 10.1109/TEVC.2013.2281535. URL https://doi.org/10.1109/TEVC.2013.2281535.

Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, and Baining Guo. Fabricating spatially-varying subsurface scattering. *ACM Transactions on Graphics*, 29(4):62:1–62:10, July 2010. ISSN 0730-0301.

Tao Du, Adriana Schulz, Bo Zhu, Bernd Bickel, and Wojciech Matusik. Computational multicopter design. *ACM Trans. Graph.*, 35(6):227:1–227:10, November 2016. ISSN 0730-0301. doi: 10.1145/2980179.2982427. URL http://doi.acm.org/10.1145/2980179.2982427.

GE. Ge jet engine bracket challenge, 2013. URL https://grabcad.com/challenges/ge-jet-engine-bracket-challenge.

C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, Sep 2001a. ISSN 1573-2878.

Claus Hillermeier. *Nonlinear multiobjective optimization: a generalized homotopy approach*, volume 135. Springer Science & Business Media, 2001b.

Gary Johnson. *Wind Energy Systems (electronic edition)*. 2006. URL http://www.ece.k-state.edu/people/faculty/gjohnson/files/Windbook.pdf. Original work published 1994.

William Karush. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Dept. of Mathematics, University of Chicago, Chicago, Illinois, 1939.

Yuki Koyama, Issei Sato, and Masataka Goto. Sequential gallery for interactive visual design optimization. *ACM Trans. Graph.*, 39(4), July 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392444. URL https://doi.org/10.1145/3386569.3392444.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif., 1951. University of California Press. URL https://projecteuclid.org/euclid.bsmsp/1200500249.

Dingzeyu Li, David I.W. Levin, Wojciech Matusik, and Changxi Zheng. Acoustic voxels: Computational optimization of modular acoustic filters. *ACM Trans. Graph.*, 35(4), 2016. doi: 10.1145/2897824.2925960.

R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, Apr 2004. ISSN 1615-1488. doi: 10.1007/s00158-003-0368-6. URL https://doi.org/10.1007/s00158-003-0368-6.

Adanay Martín and Oliver Schütze. Pareto tracer: a predictor-corrector method for multi-objective optimization problems. *Engineering Optimization*, 50(3):516–536, 2018.

David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. A review and taxonomy of interactive optimization methods in operations research. *ACM Trans. Interact. Intell. Syst.*, 5(3), September 2015. ISSN 2160-6455. doi: 10.1145/2808234. URL https://doi.org/10.1145/2808234.

A. Messac, A. Ismail-Yahaya, and C.A. Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98, Jul 2003. ISSN 1615-1488.

Farzin H. Montazersadgh and Ali Fatemi. Stress analysis and optimization of crankshafts subject to dynamic loading, August 2007. URL https://www.steelsustainability.org/~/media/Files/Autosteel/Programs/LongProducts/crankshaft_full_report_2.pdf.

Julian Panetta, Abtin Rahimian, and Denis Zorin. Worst-case stress relief for microstructures. *ACM Trans. Graph.*, 36(4):122:1–122:16, July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073649. URL http://doi.acm.org/10.1145/3072959.3073649.

PHeller. Horst-link pivot placement and pedaling efficiency (norco art), October 2014. URL https://www.pinkbike.com/u/PHeller/blog/horst-link-pivot-placement-and-pedaling-efficiency-norco-art.html.

Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. Make it stand: Balancing shapes for 3d fabrication. *ACM Trans. Graph.*, 32(4):81:1–81:10, July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461957. URL http://doi.acm.org/10.1145/2461912.2461957.

J. Rakowska, R.T. Haftka, and L.T. Watson. Tracing the efficient curve for multi-objective control-structure optimization. *Computing Systems in Engineering*, 2(5):461 – 471, 1991. ISSN 0956-0521. doi: https://doi.org/10.1016/0956-0521(91)90049-B. URL http://www.sciencedirect.com/science/article/pii/095605219190049B. Computational Structures Technology 3- Part 2.

Ana B. Ruiz, Francisco Ruiz, Kaisa Miettinen, Laura Delgado-Antequera, and Vesa Ojalehto. Nautilus navigator: free search interactive multiobjective optimization without trading-off. *Journal of Global Optimization*, 74(2):213–231, Jun 2019. ISSN 1573-2916. doi: 10.1007/s10898-019-00765-2. URL `https://doi.org/10.1007/s10898-019-00765-2`.

Adriana Schulz, Jie Xu, Bo Zhu, Changxi Zheng, Eitan Grinspun, and Wojciech Matusik. Interactive design space exploration and optimization for cad models. *ACM Transactions on Graphics*, 36(4), jul 2017.

Adriana Schulz, Harrison Wang, Eitan Grinspun, Justin Solomon, and Wojciech Matusik. Interactive exploration of design trade-offs. *ACM Trans. Graph.*, 37(4): 131:1–131:14, July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201385. URL `http://doi.acm.org/10.1145/3197517.3201385`.

Christian Schumacher, Jonas Zehnder, and Moritz Bächer. Set-in-stone: Worst-case optimization of structures weak in tension. *ACM Trans. Graph.*, 37(6), December 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275085. URL `https://doi.org/10.1145/3272127.3275085`.

Oliver Schütze, Oliver Cuate, Adanay Martín, Sebastian Peitz, and Michael Dellnitz. Pareto explorer: a global/local exploration tool for many-objective optimization problems. *Engineering Optimization*, 0(0):1–24, 2019.

Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. High-contrast computational caustic design. *ACM Trans. Graph.*, 33(4):74:1–74:11, July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601200. URL `http://doi.acm.org/10.1145/2601097.2601200`.

Seven Cycles. Seven design philosophy – understanding chainstays, 2020. URL `https://www.sevencycles.com/chainstay-philosophy.php`.

Maria Shugrina, Ariel Shamir, and Wojciech Matusik. Fab forms: Customizable objects for fabrication with validity and geometry caching. *ACM Trans. Graph.*, 34(4), July 2015. ISSN 0730-0301. doi: 10.1145/2766994. URL `https://doi.org/10.1145/2766994`.

A. Spielberg, B. Araki, C. Sung, R. Tedrake, and D. Rus. Functional co-optimization of articulated robots. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5035–5042, 2017.

Ed Spratt. The tech behind the new atherton bikes, April 2019. URL `https://www.pinkbike.com/news/the-tech-behind-the-new-atherton-bikes.html`.

Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832), 2020. Downloaded on May 19, 2020 from http://underactuated.mit.edu/.

Erva Ulu, James Mccann, and Levent Burak Kara. Lightweight structure design under force location uncertainty. *ACM Trans. Graph.*, 36(4), July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073626. URL https://doi.org/10.1145/3072959.3073626.

Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.*, 31(4):86:1–86:11, July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185582. URL http://doi.acm.org/10.1145/2185520.2185582.

Nobuyuki Umetani, Yuki Koyama, Ryan Schmidt, and Takeo Igarashi. Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph.*, 33(4):65:1–65:10, July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601129. URL http://doi.acm.org/10.1145/2601097.2601129.

David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evol. Comput.*, 8(2):125–147, June 2000. ISSN 1063-6560. doi: 10.1162/106365600568158. URL https://doi.org/10.1162/106365600568158.

Lingfeng Wang and Emily Whiting. Buoyancy optimization for computational fabrication. *Computer Graphics Forum (Proceedings of Eurographics)*, 35(2), 2016.

Shawn Wasserman. Topology optimization comes to solidworks, October 2017. URL https://www.engineersrule.com/topology-optimization-comes-solidworks/.

Jiaxian Yao, Danny M. Kaufman, Yotam Gingold, and Maneesh Agrawala. Interactive design and stability analysis of decorative joinery for furniture. *ACM Trans. Graph.*, 36(2):20:1–20:16, March 2017. ISSN 0730-0301. doi: 10.1145/3054740. URL http://doi.acm.org/10.1145/3054740.

J. Zhang and L. Xing. A survey of multiobjective evolutionary algorithms. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, volume 1, pages 93–100, July 2017.

Qingnan Zhou, Julian Panetta, and Denis Zorin. Worst-case structural analysis. *ACM Trans. Graph.*, 32(4), July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461967. URL https://doi.org/10.1145/2461912.2461967.

Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. Two-scale topology optimization with microstructures. *ACM Transactions on Graphics*, 36(5):164:1–164:16, July 2017. ISSN 0730-0301.

Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, June 2000. ISSN 1063-6560. doi: 10.1162/106365600568202. URL http://dx.doi.org/10.1162/106365600568202.