

A GENETIC ALGORITHM APPROACH FOR TECHNOLOGY
CHARACTERIZATION

A Thesis

by

EDGAR GALVAN

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

MASTER OF SCIENCE

August 2012

Major Subject: Mechanical Engineering

A Genetic Algorithm Approach for Technology Characterization

Copyright 2012 Edgar Galvan

A GENETIC ALGORITHM APPROACH FOR TECHNOLOGY
CHARACTERIZATION

A Thesis

by

EDGAR GALVAN

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

MASTER OF SCIENCE

Approved by:

Chair of Committee,
Committee Members,

Head of Department,

Richard J. Malak
Andrew L. Johnson
Daniel A. McAdams
Jerald A. Canton

August 2012

Major Subject: Mechanical Engineering

ABSTRACT

A Genetic Algorithm Approach for Technology Characterization. (August 2012)

Edgar Galvan, B.S., Texas A&M University

Chair of Advisory Committee: Dr. Richard J. Malak

It is important for engineers to understand the capabilities and limitations of the technologies they consider for use in their systems. Several researchers have investigated approaches for modeling the capabilities of a technology with the aim of supporting the design process. In these works, the information about the physical form is typically abstracted away. However, the efficient generation of an accurate model of technical capabilities remains a challenge. Pareto frontier based methods are often used but yield results that are of limited use for subsequent decision making and analysis. Models based on parameterized Pareto frontiers—termed Technology Characterization Models (TCMs)—are much more reusable and composable. However, there exists no efficient technique for modeling the parameterized Pareto frontier. The contribution of this thesis is a new algorithm for modeling the parameterized Pareto frontier to be used as a model of the characteristics of a technology. The novelty of the algorithm lies in a new concept termed *predicted dominance*. The proposed algorithm uses fundamental concepts from multi-objective optimization and machine learning to generate a model of the technology frontier.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Richard Malak, and my committee members, Dr. Andrew Johnson, Dr. Daniel McAdams for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES.....	vii
1. INTRODUCTION	1
2. TECHNOLOGY CHARACTERIZATION.....	5
2.1. Conceptual Foundation.....	5
2.2. TCM Representation.....	6
3. PROBLEM FORMULATION AND PRIOR METHODS.....	9
3.1. Search Problem Formulation.....	9
3.2. Prior Methods	10
4. AN IMPROVED SEARCH ALGORITHM	13
5. PREDICTIVE PARAMETERIZED PARETO GENETIC ALGORITHM	16
5.1. Determining the Predicted Feasible Set.....	16
5.2. Assigning Fitness.....	19
5.2.1. Predicted Parameterized Pareto Dominance	19
5.2.2. Nondominated Sorting.	22
5.2.3. Diversity Preservation Operator.....	22
5.3. Selection	24
5.4. Main Loop	25
6. RESULTS AND ANALYSIS.....	27
6.1. Classical Pareto Test Problems.....	27
6.2. Parameterized Pareto Test Problems	29
6.3. Measuring Approximation Error	32

	Page
6.4. Simulation Results For the Classical Pareto Case	33
6.5. Simulation Results For the Parameterized Pareto Case.....	36
6.6. Engineering Case Study.....	39
6.6.1. TCM Generation.	39
6.6.2. Searching the TCM.	41
6.6.3. Target Achievement.	44
6.7. Engineering Case Study Results.....	45
7. DISCUSSION	50
8. SUMMARY	53
REFERENCES.....	54
APPENDIX.....	58
VITA	62

LIST OF FIGURES

	Page
Figure 1. Example of TCM composition for two competing technologies in the performance space. Figure (a) illustrates the separate TCMs before composition and figure (b) shows the composed TCM.	6
Figure 2. Illustration of dominance analysis on randomly generated design sites when classical Pareto dominance is (a) and is not (b) applicable.	15
Figure 3. An illustration of predictive parameterized Pareto dominance.	21
Figure 4. An Example of crowding distance metrics (a) PPR and (b) NSGA-II Crowding distance.	23
Figure 5. P3GA Flow Chart.	26
Figure 6. Illustration of the classical Pareto frontier for test problem ZDT1	28
Figure 7. Illustration of the classical Pareto frontier for test problem DTLZ2.	28
Figure 8. Illustration of the parameterized Pareto frontier for test problems 1a and 1b.	30
Figure 9. Illustration of the parameterized Pareto frontier for test problem 2a.	31
Figure 10. Illustration of the parameterized Pareto frontier for test problem 2b.	31
Figure 11. Illustration of the parameterized Pareto frontier for test problems 3a and 3b.	32
Figure 12. Hausdorff error as a function of number of generations for test problems ZDR1 (a) and DTLZ2 (b).	35
Figure 13. Illustration of P3GA Pareto-optimal solution for test problem DTLZ2 at (a) 10 generations and (b) 100 generations.	35
Figure 14. Mean Hausdorff distance as a function of number of generations for test problem 1, versions (a) and (b), at different values of the q parameter.	37
Figure 15. Mean Hausdorff distance as a function of number of generations for test problem 2, versions (a) and (b), at different values of the q parameter.	37

Figure 16. Mean Hausdorff distance as a function of number of generations for test problem 3, versions (a) and (b), at different values of the q parameter.	38
Figure 17. Illustration of the vertical distance between the true parameterized Pareto frontier and the approximations generated through (a) P3GA and (b) random approach for test problem 1a at 100 generations and $q = 6$	38
Figure 18. Illustration of the general design scenario using TCMs. No iteration between component and system level.	44
Figure 19. Illustration of (a) the mean distance between the parameterized Pareto-optimal points and the approximated parameterized Pareto frontier at various generations and (b) the corresponding mean constraint violation of Eq. (17).....	49

1. INTRODUCTION

It is important for engineers to understand the capabilities and limitations of the technologies they consider for use in their systems. Failure to appreciate what is achievable (or not achievable) by subsystem technologies can result in the selection of poor concepts, the derivation of poor design requirements (that are either unachievable or overly conservative), and excessive design iteration. However, it can be challenging for engineers to navigate the many competing considerations that arise in the development of large systems that have many interacting subsystems.

Several researchers have investigated approaches for modeling the capabilities of a technology quantitatively with the aim of supporting technology selection, design space exploration, and trade-off analyses. A common thread among these works is that in order to model the capabilities of a technology, one abstracts away information about its physical form. Mathematically, the result is a model defined in the space of performance attributes for a given technology. With these models, designers can represent the abstracted capabilities, performance traits, or metrics of a component or system.

It can be advantageous for engineers to use abstract models of competing technologies. Abstract models help engineers focus on the part of a problem most germane to decision making (i.e., what a technology can achieve rather than how it

This thesis follows the style of Journal of Mechanical Design.

achieves it). Abstract models can also provide engineers with the capacity to consider multiple physically heterogeneous technologies in the same variable space. For example, although batteries and fuel cells are very different physically, engineers can consider them both as energy storage technologies with the same performance attributes such as power density, energy density, and efficiency. Furthermore, abstract models can be desirable in a multi-organizational collaborative design setting. Because they abstract away low-level implementation details, they tend to hide proprietary information that collaborating organizations may not want to share.

Several approaches have been proposed in the literature for capturing and modeling the capabilities of a component or system. Under set-based design approaches to technology characterization, one uses a mathematical representation of the set of performance attributes that are technically feasible [1, 2]. Initially, the components or designs are cataloged into a hierarchical structure. Then, the performance attributes of the feasible set of designs are abstracted to form descriptions for use in higher levels in the hierarchy. The union of the basic sets represents the achievable performance characteristics of the system. Designs are eliminated from consideration through propagation of interval constraints in the performance space [3]. Taking an engine design example, the constraints may be design specifications for air flow rate, horsepower, torque, and so on. Finally, a user provided cost function is used to select one of the remaining designs. A drawback of this approach is that designers must make restrictive assumptions about what is desirable in the performance space. When design is done by this approach, many degrees of freedom are removed from the design problem.

This can result in eliminating the most preferred design from consideration [4].

Furthermore, as the constraints are made more complex, the inherent computational complexity of propagating the constraints increases rapidly [3].

Another approach involves the use of the Pareto-optimal frontier. Under this approach, one constructs a mathematical representation of the subset of performance attributes that are Pareto nondominated [5-7]. The benefits of this approach are that the dominance analysis step is relatively computationally inexpensive and will not eliminate the most preferred design [8]. The principle limitation of this approach is that designers must ensure that their preferences are monotonic in each attribute in order to perform dominance analysis. In many practical situations, a designer can formulate problems such that the monotonicity condition holds by defining the attributes appropriately. However, this reformulation is problem-specific and limits the reusability of the representation. This can be problematic since designers often wish to represent a system in terms of problem independent attributes, e.g., gear ratio, spring constants, etc. In these cases designers cannot apply Pareto dominance to eliminate designs from consideration. Malak and Paredis developed an extension of Pareto dominance—termed parameterized Pareto dominance—to address this limitation [9]. Using the concept of parameterized Pareto dominance, dominance analysis can be performed in the case where designers cannot determine a preference ordering for one or more attributes.

The efficient generation of an accurate technical capabilities model remains a challenge. Set-based methods can be used, but require restrictive assumptions and computationally challenging constraint propagation methods. Pareto frontier methods are

more general in this respect, but yield models that are of limited use for subsequent decision making and analysis. Models based on parameterized Pareto frontiers—termed Technology Characterization Models (TCMs)—are much more reusable (can compose, etc.) [10]. However, there exists no efficient technique for generating an estimate of the parameterized Pareto frontier. The contribution of this thesis is a new algorithm for generating an approximation of the parameterized Pareto frontier for use in modeling the characteristics of a technology. The novelty of the algorithm lies in a new concept termed *predicted dominance*. The algorithm uses fundamental concepts from multi-objective optimization and machine learning to search for solutions near the parameterized Pareto frontier.

This thesis is organized as follows. Section 2 is an overview of technology characterization models. Section 3 contains a detailed definition and mathematical formulation of the problem of interest and prior methods. Section 4 is an introduction to multi-objective genetic algorithms in an engineering context. Section 5 is a detailed description of the proposed algorithm. Section 6 is an analysis of the performance of the algorithm on test problems followed by a demonstration. The thesis concludes with Section 7 and 8 which are a discussion of the results and a summary, respectively.

2. TECHNOLOGY CHARACTERIZATION

2.1. Conceptual Foundation

A Technology Characterization Model (TCM) is a mathematical representation of the capabilities of technology or component in the abstracted performance space [10]. The concept of abstraction involves mapping the feasible low-level design variable space to a higher level performance space using engineering analysis models. By abstracting the performance characteristics of a component or technology, designers can focus on those variables that relate component-level performance to system-level performance while ignoring low-level design variables and analysis models. A key benefit of using TCMs is the support of composability through abstraction; see [9] for proof of composability. Abstracting away the low level design variables allows designers to compare competing technologies that have different analysis models and design variables in the same performance space. For example, suppose automotive designers wish to explore various engine types. Engine concepts of interest might include a turbocharged inline 4-cylinder, a V6, and a V8 Diesel. Although these concepts have different design variables and analysis models, they can be compared in the same performance space in terms of fuel economy, horse power, cost, etc. Figure 1 is an example of abstraction and composition in the case where a designer has established a preference for every performance attribute. Through composition, the single TCM in Figure 1b can capture the capabilities and limitations of the competing technologies in

Figure 1a. A designer could then search a single unified model that captures the salient characteristics of the competing technologies.

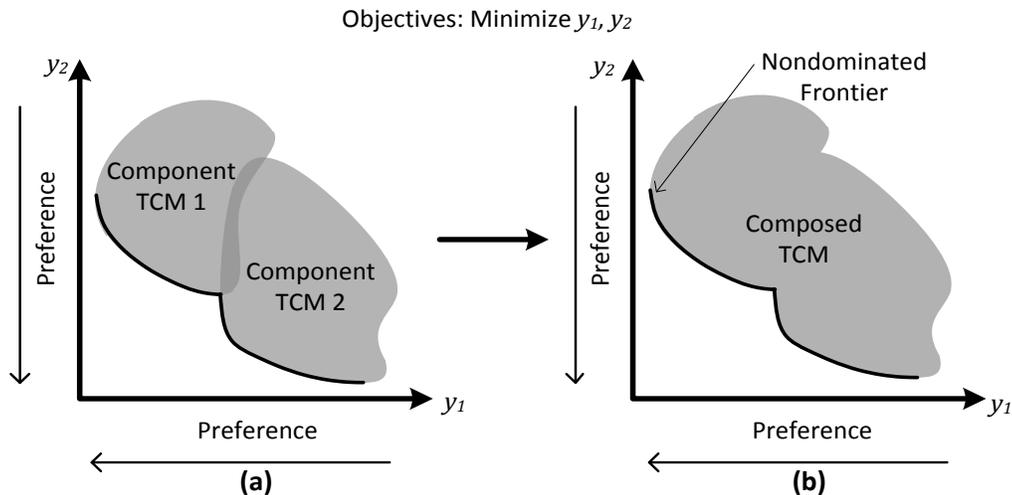


Figure 1. Example of TCM composition for two competing technologies in the performance space. Figure (a) illustrates the separate TCMs before composition and figure (b) shows the composed TCM.

2.2. TCM Representation

There are two principle methods for representing TCMs: (1) the entire feasible set of performance characteristic or attributes of a technology, and (2) a subset of the feasible set of performance attributes of a technology. The illustrations in Figure 1 are of TCMs that model the feasible set of performance attributes of a technology. However, many designs in the feasible space of performance attributes are provably inferior to others such that no rational designer would choose them. These designs are said to be

dominated, and the set of all nondominated designs is the nondominated frontier, illustrated in Figure 1 [11].

Since a rational designer will always prefer designs on the nondominated frontier, the region to be modeled can be significantly reduced by eliminating the dominated designs from consideration. In the case where the designer has established preference direction for every performance attribute, classical Pareto dominance can be used to determine the nondominated designs. Minimization will be used throughout this thesis without loss of generality since maximizing f is equivalent to minimizing $-f$. Suppose designer preferences are monotonically decreasing in each performance attribute y_i for $i = 1, \dots, M$. If $\mathbf{y} = [y_1, y_2, \dots, y_M]$ denotes a vector of performance attributes and $\mathbf{Y} \subseteq \mathbf{R}^M$ is the set of all feasible performance attributes, then classical Pareto dominance is defined as follows [12]:

Definition 1. *An alternative having attributes \mathbf{y} is Pareto dominated by one with attributes \mathbf{y}' if and only if, $y'_i \leq y_i \forall i = 1, \dots, M$ and $y'_i < y_i \exists i = 1, \dots, M$.*

Technology characterization models can also be generated in the case where a designer has established a monotonic preference direction for one or more performance attributes (termed dominators) but not others (termed parameters). For example, the designer of a transmission may know to reduce cost and increase reliability but may not have yet determined a preference direction for gear ratios. In this example, classical Pareto dominance cannot be used to eliminate undesirable (dominated) designs due to the presence of the parameter attribute [9]. Parameterized Pareto dominance, an

extension of classical Pareto dominance, can be used to eliminate designs that would never be preferred over the remaining designs [12]. For a system with attributes indexed $1, \dots, M$, let D denote the nonempty set of indices for the dominator attributes and P denote the set of indices corresponding to the parameter attributes such that $D \cup P = \{1, \dots, M\}$ and $D \cap P = \emptyset$. The mathematical formulation of parameterized Pareto dominance is as follows [12]:

Definition 2. *An alternative having attributes \mathbf{y}'' is parametrically Pareto dominated by one with attributes \mathbf{y}' if $y'_i = y''_i \ \forall i \in P$, $y'_i \leq y''_i \ \forall i \in D$ and $y'_i < y''_i \ \exists i \in D$, where P is the set of parameter attribute indices and D is the (non-empty) set of dominator attribute indexes.*

In the case where there are no parameters, $P = \emptyset$, parameterized Pareto dominance reduces to classical Pareto dominance. It is important to note that an alternative can only dominate another if the parameter attributes are *equal*; one solution cannot dominate another if the parameter values are not equal. Using parameterized Pareto dominance, designers can identify and eliminate designs that cannot possibly be the most preferred and therefore do not need to be included in the TCM.

3. PROBLEM FORMULATION AND PRIOR METHODS

3.1. Search Problem Formulation

In order to formalize the mathematical search problem for the parameterized Pareto set, it is helpful to first consider the case without parameters, i.e. the classical Pareto case. Suppose designer preferences are monotonically decreasing in each performance attribute, $\mathbf{y} = [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_k(\mathbf{x})]$, where \mathbf{x} is the vector of optimization or decision variables. The search problem for the Pareto frontier is posed as follows:

$$\begin{aligned} & \underset{\mathbf{x} \in \mathcal{S}}{\text{Minimize}} \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_k(\mathbf{x})]^T \\ & \text{subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m, \\ & \quad \quad \quad h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, n, \end{aligned} \tag{1}$$

where k is the number of objective functions, $g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m$ are the inequality constraints, and $h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, n$ are the equality constraints [8]. The set \mathcal{S} is nonempty feasible region of the decision variable space. The optimization problem has multiple objectives, $F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_k(\mathbf{x})$, that must be minimized simultaneously. The optimization problem has an infinite number of solutions that define the Pareto frontier.

The search problem for the parameterized Pareto frontier is an extension of Eq. (1). Intuitively, this search problem can be thought of as an application of Eq. (1) at every combination of feasible parameter values. Consider the case with k performance attributes of which $0 \leq u < k$ are parameters such that $\mathbf{p} = [p_1, p_2, \dots, p_u]$, where

$p_l = F_{k-u+l}(\mathbf{x})$. The feasible parameter region (set) is then $\mathbf{L} = [\mathbf{L}_1 \times \mathbf{L}_2, \dots, \mathbf{L}_u]$, where $\mathbf{L}_l = \{F_{k-u+l}(\mathbf{x}) : \mathbf{x} \in \mathcal{S}\}$. With this notation, Eq. (1) is extended to the case where one or more of the performance attributes are parameters as

$$\begin{aligned} \forall \mathbf{p} &= [p_1, p_2, \dots, p_u] \in \mathbf{L}: \\ \text{Minimize } \mathbf{F}(\mathbf{x}) &= [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_{k-u}(\mathbf{x})]^T \\ &\text{subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m, \\ &h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, n, \\ &F_{k-u+l}(\mathbf{x}) = p_l, \quad l = 1, 2, \dots, u, \end{aligned} \tag{2}$$

where $F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_{k-u}(\mathbf{x})$ are the dominator objective functions and $F_{k-u+1}(\mathbf{x}), F_{k+2-u}(\mathbf{x}), \dots, F_k(\mathbf{x})$ are the parameter functions. This optimization problem has an infinite number of solutions that define the parameterized Pareto frontier. In the case where there are no parameters, $u = 0$, the optimization problem reduces to the classical Pareto case in Eq. (1) since the set $\mathbf{L} = \emptyset$ and the constraint $F_{k+l}(\mathbf{x})$ is not present. The focus of this thesis is on an algorithm for approximating the set of solutions to Eq. (2).

3.2. Prior Methods

In the parameterized Pareto dominance work in the literature, two main approaches for identifying design sites in the performance space have been reported:

1. Using observational performance space data from part catalogs, data sheets, etc.
2. Randomly sampling design space data and mapping those designs to the performance space using engineering analysis models.

Prior case studies have shown TCMs generated from observational performance data to be acceptable representations of the achievable range of performance [13, 14]. TCMs generated in this way approximate the information contained in a data base. The TCM can then be used to search the generalized space of capabilities to find solutions that are not in the data base but may be technically feasible. In this way, designers can leverage prior knowledge (e.g., catalog data) to describe the range of capabilities effectively and quantitatively.

Observational data, however, is not always available to the designer. This is often the case for new innovative designs, highly complex components, and mass customization components. In order to address the case where observational data is not available, researchers have resorted to sampling engineering analysis models randomly or using design of experiments (DOE) procedure [10, 12, 15]. This work is significant because it extends the usefulness of TCMs. However, because random or DOE samples are unlikely to occur on or near the efficient set for many systems, this approach can require a very large number of samples.

A possible alternative to random sampling is to directly apply multi-objective optimization (MOO) algorithms designed to solve Eq. (1) to Eq. (2). For an overview of MOO algorithms, see [8]. Recall from section 3.1 that Eq. (2) can be thought of as an application of in Eq. (1) at every combination of feasible parameter values. Therefore, MOO algorithms can be applied at various combinations of parameter values to approximate the space of solutions to Eq. (2). Suppose the problem of interest has a single parameter, $u = 1$. In principle, a MOO algorithm could be applied at v different

parameter values to approximate the set $\mathbf{L} = \mathbf{L}_1$. The union of v solutions can then be used to define the parameterized Pareto frontier. However, a straightforward application of MOO algorithms to Eq. (2) may not be feasible for more complex problems. Suppose instead the search problem has u parameter attributes. The feasible space of parameter values is then $\mathbf{L} = [\mathbf{L}_1 \times \mathbf{L}_2, \dots, \mathbf{L}_u]$. In this case, the number of multi-objective search problems that must be solved grows according to v^u . If the required function evaluations are computationally expensive, this can become prohibitive even for low numbers of parameter attributes. Designers would benefit from a search algorithm that has a lower computational growth rate.

4. AN IMPROVED SEARCH ALGORITHM

Genetic algorithms (GAs) are search and optimization procedures motivated by the principles of evolution and natural selection [16]. A GA typically begins its search with a random set of solutions, or members, called the population. Each member of the population is evaluated and assigned a fitness value using a fitness function. Then a termination condition is checked; if the termination criterion is not satisfied three main operators modify the population of the solutions: selection, crossover, and mutation. Each new population is termed a generation. Multi-objective Genetic Algorithms (MOGAs) are a type of MOO algorithm that uses the general GA strategy to solve the search problem in Eq. (1). The goals of a MOGA are to:

1. Find a set of solutions close to the Pareto frontier.
2. Maintain diversity in the solution set.

Most MOGAs rely on the concept of dominance to assign higher fitness values to members closer to the Pareto frontier. In these algorithms, two members are compared on the basis of whether one dominates the other or not. The nondominated member is assigned a higher fitness value. The second goal is typically achieved through the use of diversity preservation operator. Several researchers have successfully approximated the set of solutions to Eq. (1) using MOGAs [17-19]. However, as discussed earlier, a straightforward application of MOGAs to Eq. (2) may not be feasible due to the exponential computational growth pattern. Suppose that the problem of interest has $u = 2$

parameters, and that in order to approximate the parameter space, Eq. (1) is to be solved at $v = 20$ different parameter values. If the MOGA is to be run with a population size $p = 100$ and number of generations $g = 250$. The total number of required function evaluations would be $pgv^u = 10^7$. If each function evaluation takes just 5 seconds to evaluate the computational runtime would be over 3 years. Although this process is highly parallelizable, an approach with lower computational expense is desirable.

In the proposed algorithm the general MOGA scheme and goals are adapted to be compatible with parameterized Pareto dominance. This way, the exponential computation growth pattern described above is avoided. An algorithm that solves the search problem in Eq. (2) should have similar goals as MOGAs does in solving Eq. (1). The principle difference being that rather than finding solutions close to the Pareto frontier, the algorithm must find solutions close to the parameterized Pareto frontier. The MOGA scheme is adapted by replacing the concept of classical Pareto dominance with parameterized Pareto dominance.

However as stated in definition 2, parameterized Pareto dominance can only be performed on members with equal parameter values. Because GAs are variants of randomized search algorithms, this is unlikely to occur. Figure 2 illustrates this difficulty. Figure 2a is an illustration of classical Pareto dominance analysis in the case where the designer has monotonically decreasing preferences for attributes y_1 and y_2 . The nondominated members are those that have no other member within the rectangle created by the origin and itself. In Figure 2a, member a is dominated by b while members c and b are mutually nondominating. Figure 2b is an illustration of the case

where the designer has monotonically decreasing preference for attribute y_2 but has not established a monotonic preference direction in y_1 . In this case, a member can only be dominated by another with equal parameter values. In Figure 2b, none of the members can be said to be dominated since dominance can only occur along the dashed lines.

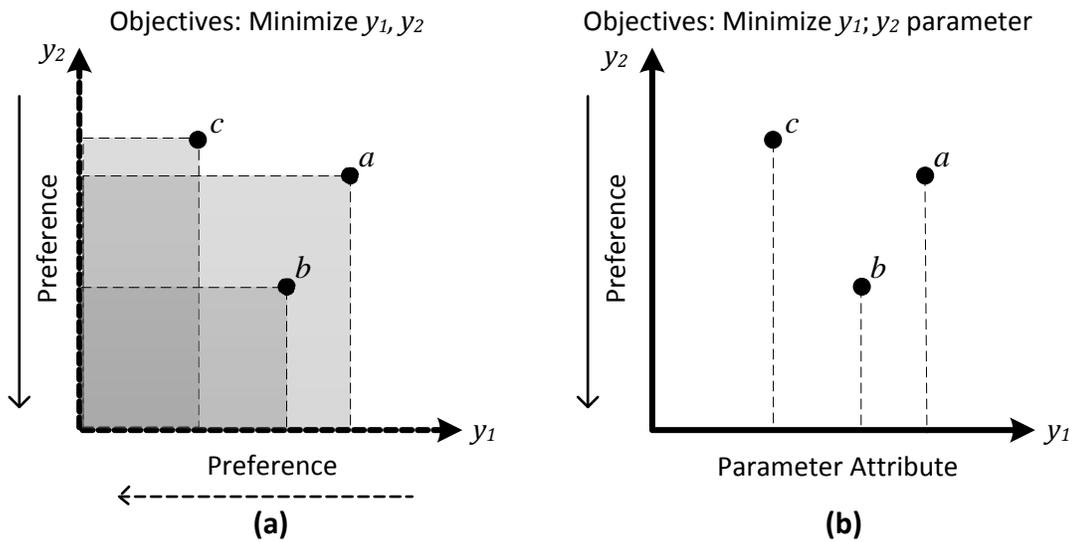


Figure 2. Illustration of dominance analysis on randomly generated design sites when classical Pareto dominance is (a) and is not (b) applicable.

To overcome this difficulty we propose that dominance analysis be performed using design sites that are *predicted* to be feasible rather than current members of the population. In the proposed algorithm, a machine learning technique is used to predict the feasible domain in the performance space by using the current population as training data. A nondominated member is one that is not parametrically Pareto dominated by any point in the predicted feasible domain. The following section contains a more detailed explanation of this concept and introduces an algorithm for solving Eq. (2).

5. PREDICTIVE PARAMETERIZED PARETO GENETIC ALGORITHM

Because of the reliance on predicted parameterized Pareto dominance, the proposed algorithm is called, Predictive Parameterized Pareto Genetic Algorithm (P3GA). The following is a high-level description of P3GA.

1. Randomly initialize population.
2. Determine predicted feasible domain based on the current population using machine learning.
3. Assign fitness based on predicted parameterized Pareto dominance and a diversity preservation operator.
4. If termination criteria have not been met, use selection, crossover, and mutation, operators to generate offspring and combine with parents. Else, terminate.
5. Return to step 2.

The procedure outlined closely parallels the general procedure of many MOGAs in the literature. The novelty of P3GA is in the concept of predicted parameterized Pareto dominance. The following subsections contain a detailed description of the procedures for accomplishing the above steps.

5.1. **Determining the Predicted Feasible Set**

In order to determine the predicted feasible set, P3GA relies on the kernel based support vector domain description (SVDD) proposed by Tax and Duin [20]. The SVDD method is a machine learning technique for modeling the boundary of a set of data in a

Euclidian space. For a detailed description of the procedure for generating domain descriptions using this technique see [14, 20]. The SVDD method determines whether a new point is in the data domain by using a hypersphere that envelops the training data. One tries to find the minimum-radius hypersphere that contains a set of N data points, $\{\mathbf{x}_i | i = 1, \dots, N\}$. The domain can then be represented by a hypersphere center at \mathbf{a} and radius R . Thus the most rudimentary constraint is

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 \quad \forall i. \quad (3)$$

However, because a hypersphere is typically a poor representation of the domain, a kernel function is used to nonlinearly remap the training data into a higher-dimensional feature space where a hypersphere is a good model. There are several valid kernel functions common in the literature [21]. The proposed algorithm uses the Gaussian kernel function

$$K_G(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) = e^{-q\|\mathbf{x}_i - \mathbf{x}_j\|^2}, \quad (4)$$

where $\Phi(\cdot)$ is the nonlinear mapping from the data space to the feature space. The q parameter determines how “tightly” or “loosely” the domain description is fit around the training data. The constraint in Eq. (3) becomes

$$\|\Phi(\mathbf{x}_i) - \mathbf{b}\|^2 \leq R^2 + \xi_i \quad \forall i, \quad (5)$$

where \mathbf{b} is the centroid of the feature space hypersphere and ξ_i are slack variables introduced that allow the exclusion of outliers. Rewriting in terms of the kernel function, the Wolfe dual problem can be developed from Eq. (5) as

$$\max_{\beta_i} W = \sum_i \beta_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j), \quad (6)$$

subject to the constraints $0 \leq \beta_i \leq C \forall i$ and $\sum_i \beta_i = 1$. For a detailed method for formulating the Wolfe dual problem see [22]. For each data point, \mathbf{x}_i for $i = 1, \dots, N$, there are three possible classifications:

- It is inside the hypersphere, which is indicated by $\beta_i = 0$,
- It is on the boundary of the hypersphere, which is indicated by $0 < \beta_i < C$,
- It is an outlier outside of the hypersphere, which is indicated by $\beta_i = C$.

Data on the boundary of the hypersphere are called *support vectors* and are essential to the domain description representation. The squared distance of the feature space image of a point, \mathbf{z} , to the centroid of the hypersphere is

$$R^2(\mathbf{z}) = K(\mathbf{z}, \mathbf{z}) - 2 \sum_i \beta_i K(\mathbf{x}_i, \mathbf{z}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j). \quad (7)$$

A new test point is inside the domain description if the distance from the feature space image of test point to the hypersphere centroid and is less than the radius of the hypersphere. The expression for classification, Eq. (7), is a simple algebraic expression that is fast to evaluate.

Outliers are data points for which $\beta_i = C$. The outliers are not part of the domain description. Choosing $C \geq 1$ yields no outliers. There must be at least one support vector, (i.e., $0 < \beta_i < C$), which means $\beta_i = C$ cannot occur for $C \geq 1$. In the proposed algorithm $C = 1$ implying that the training data does not contain outliers.

An important benefit of the SVDD method is that it can be constructed incrementally and decrementally [23]. This allows for a relatively inexpensive update procedure to be used when members are added or removed from the population (domain

description). For experimental analysis on the performance of the incremental SVDD procedure and the effects of the q parameter see [24].

5.2. Assigning Fitness

As stated in Section 4, multi-objective optimization algorithms have two goals:

- Find a set of solutions close to the nondominated boundary.
- Maintain diversity in the solution set.

These goals were used to guide the development of P3GA. The proposed algorithm uses nondominated sorting to assign higher fitness to member close to the predicted parameterized Pareto frontier and a crowding distance metric to preserve diversity in the population. The following sections describe these procedures.

5.2.1. Predicted Parameterized Pareto Dominance

Using the concept of predicted Parameterized Pareto dominance, a nondominated member is one that is not parametrically Pareto dominated by any member that is predicted to be feasible. For the remainder of this thesis, *predicted parametrically Pareto nondominated* members will be referred to simply as *nondominated* members and the *nondominated frontier of predicted feasible members* will be referred to as the *nondominated frontier*.

In the proposed algorithm, any point in the domain description is considered a predicted feasible member. Therefore, any member in the interior of the domain description must be dominated. If every member in the population is part of the domain description, ($C \geq 1$) only members on the boundary of the domain description can be

nondominated, i.e., support vectors. Therefore, the non-support vectors can be ignored when performing dominance analysis. Recall from Section 5.1 that the support vectors are classified when solving the Wolf dual problem, Eq. (6). This is significant since for most data sets, only a small portion of the data will lie on the hypersphere boundary and be classified as support vectors.

Using the SVDD approach, the nondominated members of the population are simply the support vectors that lie on the nondominated frontier. To test whether the support vector lies on the nondominated frontier, the space that dominates the support vector is sampled for feasibility. The space is bounded by the minimum bounding rectangle or envelope that contains the training data. If any member in that space is feasible, the member is classified as dominated. Figure 3 is an illustration of this procedure for two members in the population that are support vectors. As illustrated in Figure 3, the sampling procedure for a support vector can be terminated as soon as a feasible member is found. The uniform sampling approach illustrated in Figure 3 is used to search the space that dominates the support vector in P3GA. Under this approach, if there are n support vectors, u dominator attributes, and v samples along each dominator attribute, determining the nondominated members requires at most nv^u evaluations of Eq. (7). In practice the number of evaluations required is much lower than the upper limit since many of the support vectors are dominated after the first sample, see the dominated member in Figure 3. Furthermore, the number of samples along each dimension, v , is typically low since the nondominated members are likely near the edges

of the bounding rectangle, see the nondominated member in Figure 3. The pseudo code for determining the nondominated members is as follows:

1. Set S as the set of all support vectors.
2. Set the sample set T as the sites to be tested for set S .
3. Find the set S' corresponding to the feasible sites in T .
4. Update $S = S \setminus S'$.
5. Terminate if all sites have been tested, else return to step 2.

The nondominated set are those members that remain in the set S . The sample set in T is updated using uniform sampling. Even though determining the feasibility of a site is a simple algebraic expression, for large numbers of dominator attributes, u , a uniform sampling approach can become prohibitive due to the exponential growth pattern. A sampling or search approach that reduces the computational growth pattern of this step is left for future work.

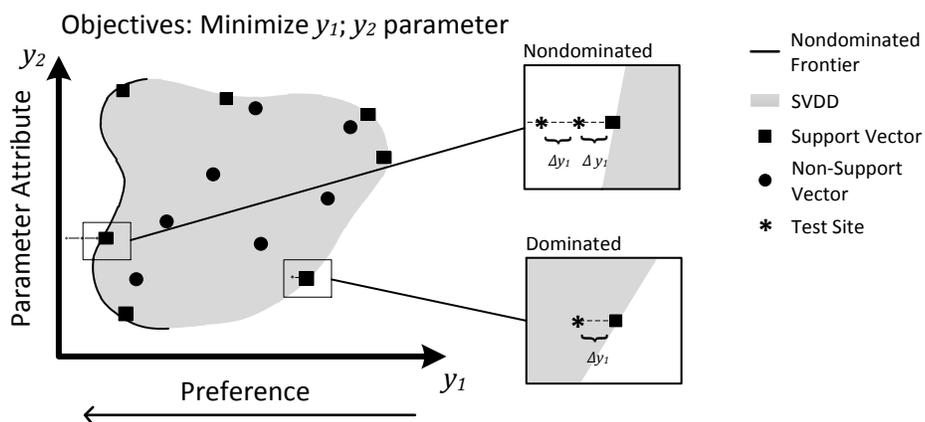


Figure 3. An illustration of predictive parameterized Pareto dominance.

5.2.2. Nondominated Sorting

In order to assign fitness to the population members, P3GA incorporates the Nondominated Sorting method developed by K. Deb[17]. In this method, before the selection operator is performed, the population is ranked on the basis of a member's non-domination. The pseudo code for the nondominated sorting procedure is as follows:

1. Set all nondominated sets P_j , for $i = 1, 2, \dots$ as empty sets.
2. Set non-domination rank counter $j = 1$.
3. Find the nondominated set P' of population P .
4. Update $P_j = P'$, $P = P \setminus P'$, and counter $j = j + 1$.
5. Return to step 3 if termination criteria is not met.

The Nondominated Sorting method is not an integral part of P3GA. It is possible to use only on the best nondominated frontier in the population. However, nondominated sorting technique is widely used and has been shown to promote population diversity and improve convergence [17, 25, 26].

5.2.3. Diversity Preservation Operator

In addition to being converged closely to the nondominated frontier, solutions must also be sparsely spaced along the nondominated frontier. Since P3GA deals with two space, decision variable space and performance space, 'diversity' among solutions can be defined in either of these spaces. Because the interest is in accurately representing the performance space, the focus is on diversity in that space. The diversity preservation operator assigns higher fitness to members that lie in less "crowded" regions of the

space. Several crowding distance metrics have been proposed in the literature. One example is Potential Pareto Region (PPR) proposed by Hallam [27]. In this approach, the nondominated set is sorted according to one objective and each two immediate neighbors delimit one PPR; an example is illustrated in Figure 4a. Another is the crowding-distance technique proposed for Nondominated Sorting Genetic Algorithm (termed NSGA-II) [17]. This technique sorts the set of solutions according to each objective function. The solutions are assigned distance values equal to the absolute difference of the objective values of the two adjacent solutions; an example is illustrated in Figure 4b.

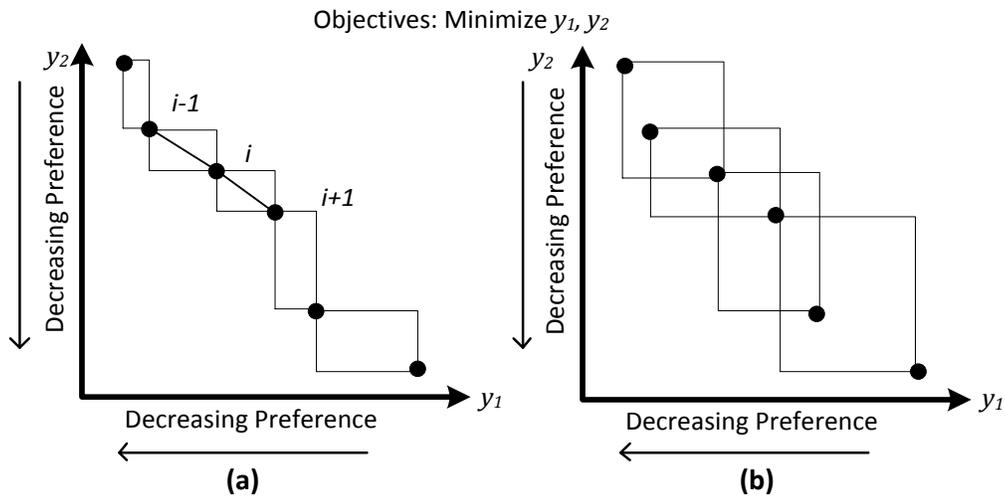


Figure 4. An Example of crowding distance metrics (a) PPR and (b) NSGA-II Crowding distance.

Although these crowding distance metrics have desirable properties, especially in regards to low computational complexity, they are not suitable for P3GA because they are not extendable to the case where one or more objective is a parameter. These crowding metrics require sorting based on ascending or descending objective values and comparing each member to its immediate neighbors. Because of the introduction of parameter values the members cannot be sorted in way that would give the crowding metric any meaning. A crowding distance based on the k-nearest neighbor (k-NN) in the Euclidian space is applied in P3GA. In this approach the 2 nearest neighbors of the population in the design space are determined. The two members with the highest Euclidean 1st neighbor distance between them are assigned the best crowding distance values. Of these two members, the member with the highest 2nd neighbor distance is assigned a better crowding distance. This process is continued until each member of the population has been assigned a crowding distance. Although this approach is more computationally expensive than many diversity preservation operators proposed in the literature, it is extendable to the case where one or more objectives is a parameter.

5.3. Selection

Once the population has been sorted into nondominated ranks, each rank is arranged in descending order of magnitude of the crowding distance values. Then the Crowded Tournament Selection operator proposed by Deb is used to determine the parents for the offspring [8]. The Crowded Tournament Selection operator is defined as follows. A solution i wins a tournament with another solution j if any of the following conditions are true:

1. If solution i has higher rank than j .
2. If solution i and j have equal rank but solution i has better crowding distance than j .

The first condition ensures that the chosen solution lies on the better nondominated frontier. The second condition resolves the tie that occurs when both solutions are on the same nondominated frontier. The solution in the less crowded area wins. Once the parents have been selected, the usual recombination and mutation operators are used to create offspring.

5.4. Main Loop

The proposed algorithm, P3GA, differs from most MOGA principally in the way the nondominated members are identified. The selection, crossover, and mutation operators are adopted directly from MOGAs in the literature. Rather than performing dominance analysis with respect to the current members of the population, dominance analysis is performed using solutions that are predicted to be feasible.

Initially, a domain description of the entire population is generated. Then the nondominated members are identified and temporarily removed from the population. The nondominated members are assigned rank 1. The crowding distance metric assigns a crowding distance value to each of the members in the nondominated rank. This process is continued until the entire population is classified into several fronts. The selection, crossover, and mutation operators reproduce the population according to the fitness value of the population members. This is continued until the termination criteria have been met. Figure 5 is a flow chart of the algorithm.

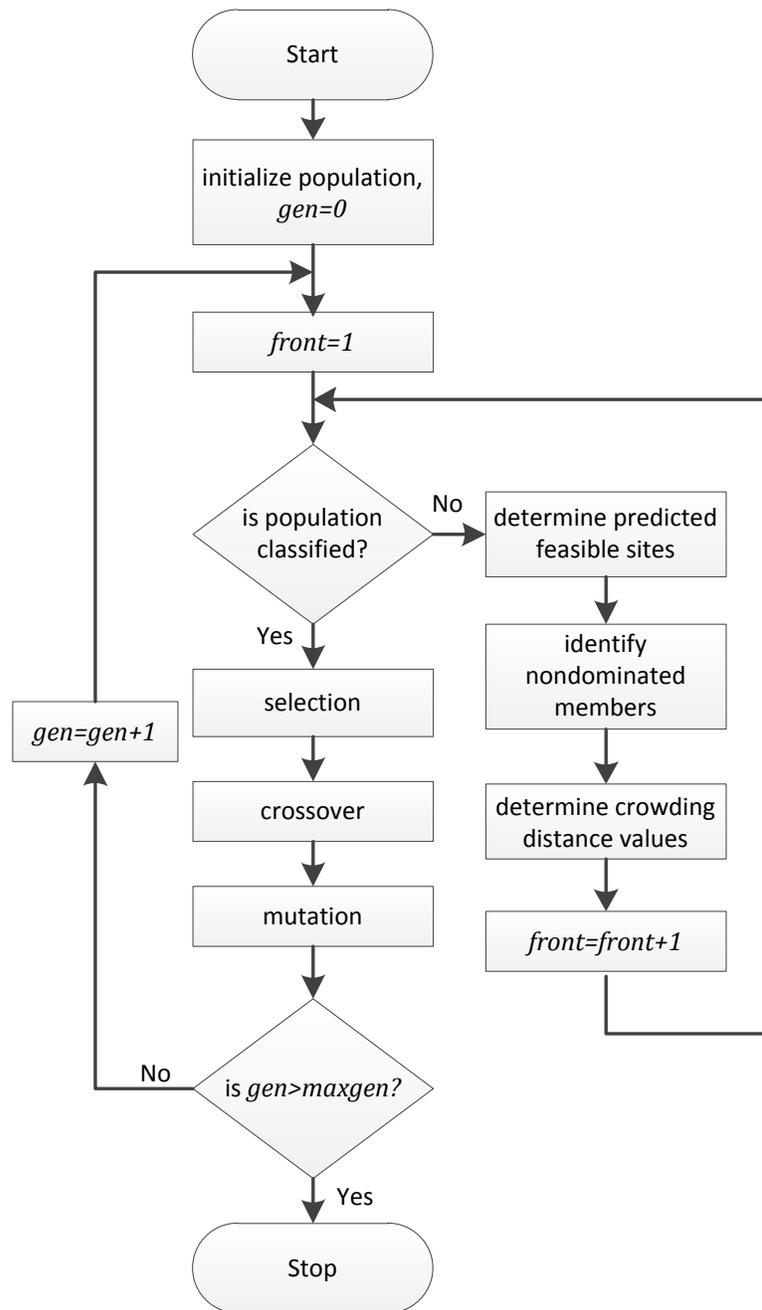


Figure 5. P3GA Flow Chart.

6. RESULTS AND ANALYSIS

In this section, some of the performance characteristics of P3GA are investigated. The algorithm, P3GA, is tested on several test problems which exhibit features that are likely to cause a difficulty in converging to the nondominated frontier. The performance of the P3GA algorithm is compared to the random approach used in earlier works. An engineering example is used to demonstrate the use of the P3GA along with TCMs in a design scenario.

6.1. Classical Pareto Test Problems

Recall from Section 3.1 that the search problem for parameterized Pareto frontier in Eq. (2), is an extension search problem for the classical Pareto frontier in Eq. (1). In the case where there are no parameters, the parameterized search problem reduces to the classical one. Consequently, P3GA can be applied to search problems without parameters, i.e., the classical Pareto case. Although P3GA was not developed for the purpose of searching for the classical Pareto frontier, examining its performance for this case can be used to help validate the concept of predicted dominance.

The test problems used for the classical Pareto case are ZDT1 and DTLZ2 proposed by K. Deb [28, 29]. These test problems are designed to exhibit features that may cause difficulty for MOGAs. Figures 6 and 7 are Illustration of the classical Pareto frontier for test problems ZDT1 and DTLZ2, respectively. The preference is to minimize

in all dimensions for both test problem. The mathematical formulation of the test problems and the Pareto frontier can be found in [28, 29].

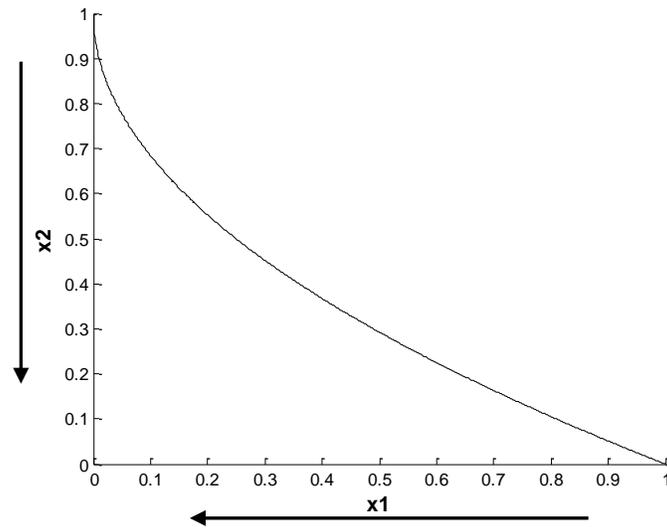


Figure 6. Illustration of the classical Pareto frontier for test problem ZDT1. The arrows indicate preference direction.

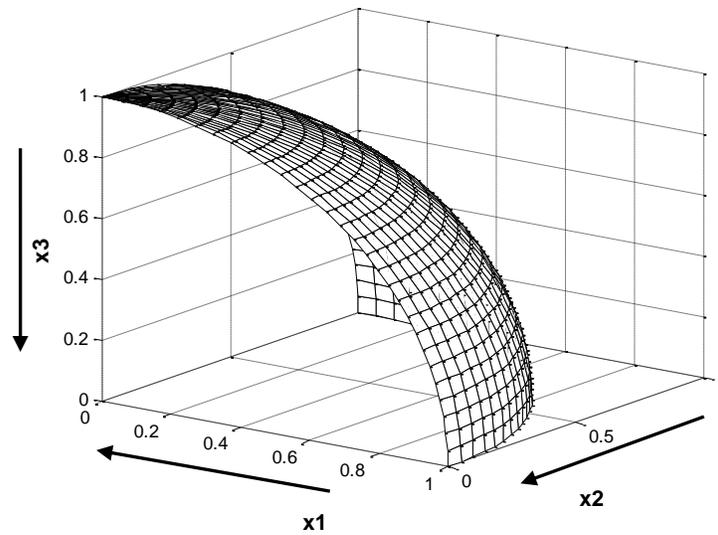


Figure 7. Illustration of the classical Pareto frontier for test problem DTLZ2. The arrows indicate preference direction.

6.2. Parameterized Pareto Test Problems

Although there are several widely used test problems for MOGAs in the classical Pareto case, they cannot be applied to the case with parameters in a straightforward manner. For this demonstration, new test problems were created that feature factors that may cause difficulty in converging to the nondominated frontier. The following factors have been identified by K. Deb [30]:

1. Convexity or non-convexity in the nondominated frontier,
2. Discontinuity in the nondominated frontier,
3. Non-uniform distribution of solutions in the nondominated frontier.
4. Multi-modality
5. Deception

The test problems were developed such that they exhibit a combination of these factors using the bottom-up approach outlined in [28]. Figures 8-11 illustrates the parameterized Pareto frontier of each of the test problems. The arrows in the figures indicate preference direction. The axes without preference directions are parameter attributes. Table 1 indicates the features exhibited by each test problem. The mathematical formulation of the test problems and the true nondominated frontier are provided in the appendix.

Table 1. Features that may cause difficulty in converging to the true nondominated frontier in each test problem.

Feature	1a	1b	2a	2b	3a	3b
Local parameterized Pareto attractors		X				
Discontinuous nondominated frontier				X		
Biased solution density away from frontier	X	X	X	X		X
Non-convex nondominated frontier			X	X	X	X
Variable solution density along frontier	X	X			X	X

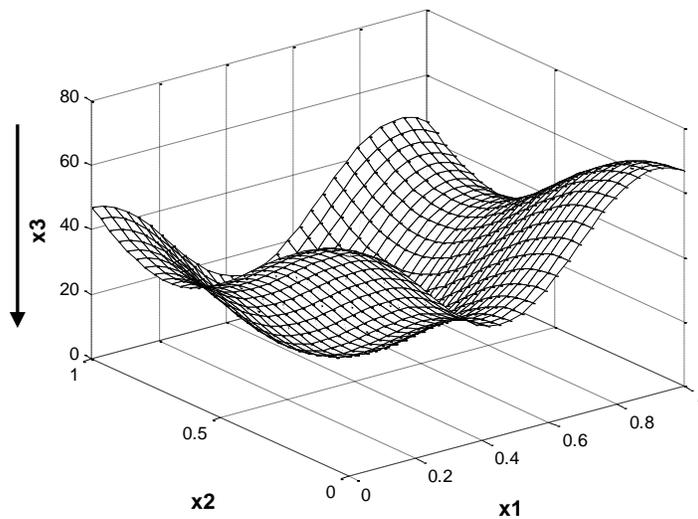


Figure 8. Illustration of the parameterized Pareto frontier for test problems 1a and 1b. The arrow indicates preference direction. Axes without preference directions are parameter attributes.

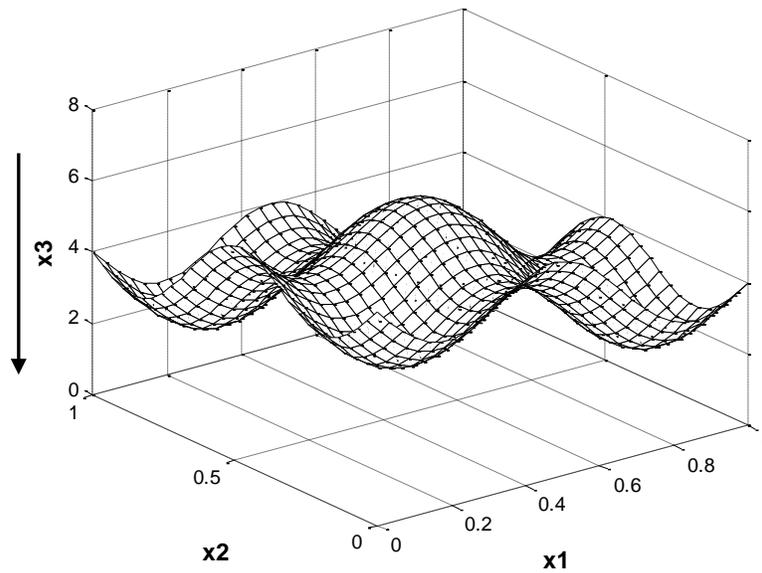


Figure 9. Illustration of the parameterized Pareto frontier for test problem 2a. The arrow indicates preference direction. Axes without preference directions are parameter attributes.

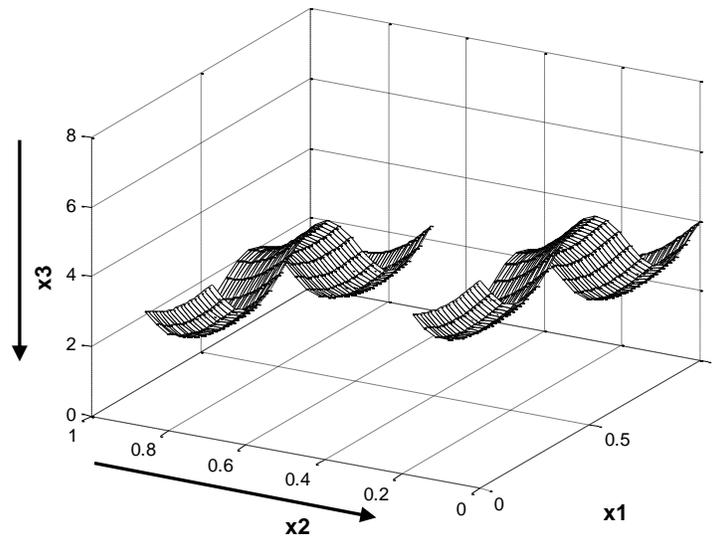


Figure 10. Illustration of the parameterized Pareto frontier for test problem 2b. The arrows indicate preference direction. Axes without preference directions are parameter attributes.

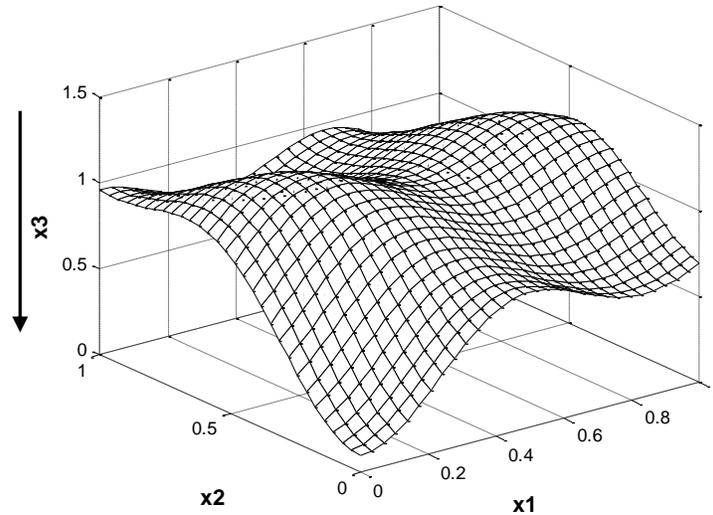


Figure 11. Illustration of the parameterized Pareto frontier for test problems *3a* and *3b*. The arrow indicates preference direction. Axes without preference directions are parameter attributes.

6.3. Measuring Approximation Error

In the literature, the performance of a MOGA is typically evaluated relative to the goals discussed in Section 4 [31]. However, because the purpose of P3GA and the focus of this thesis are on generating a model of the nondominated frontier, it is more appropriate to measure the quality of that model. For these simulations, mean Hausdorff distance error metric was used to measure the approximation error between the known nondominated frontier and the approximation. Mean Hausdorff distance has been proposed the standard measure of approximation error in the computer graphics community [32, 33]. The mean Hausdorff distance is a measure of the geometric

difference between two surfaces. The mean Hausdorff distance between two surfaces is be defined as follows. Given a point p and a surface S , the distance $e(p, S)$ is defined as

$$e(p, S) = \min_{p' \in S} d(p, p'), \quad (8)$$

where $d(\cdot)$ is the Euclidean distance between two points in E^n and n is the number of dimensions. The one sided distance between surfaces S_1 and S_2 is then

$$E(S_1, S_2) = \max_{p \in S_1} e(p, S_2). \quad (9)$$

The two sided Hausdorff distance is the maximum of $E(S_1, S_2)$ and $E(S_2, S_1)$. The *mean* distance between the two surfaces is the surface integral of the distance divided by the area of S_1

$$E_m(S_1, S_2) = \frac{1}{|S_1|} \int_{S_1} e(p, S_2) ds. \quad (10)$$

The two sided mean Hausdorff distance is the maximum of $E_m(S_1, S_2)$ and $E_m(S_2, S_1)$.

6.4. Simulation Results For the Classical Pareto Case

For the classical Pareto test problems three algorithms were compared

1. RAND: A random search algorithm.
2. NSGAI: The Nondominated Sorting Genetic Algorithm.
3. P3GA: Predicted Parameterized Pareto Genetic Algorithm.

The algorithms were executed 15 times on each test problem. The population was monitored and used as training data for the SVDD at various generation intervals. At each interval, the approximation error was determined by finding the mean Hausdorff distance between the Pareto frontier of the SVDD and the true Pareto frontier. For these results, RAND serves as an additional point of reference and randomly generates a

certain number of individuals per generation according to a uniform distribution in the input space. Hence, the number of fitness and function evaluations is the same as for the GAs.

Independent of the algorithm and test function, each simulation was carried out using the following parameters

Number of generations	:	100
Population size	:	100
Crossover rate	:	0.8
Mutation rate	:	0.01
Kernel function q parameter	:	4

No attempt was made to determine the best parameter settings. The simulation results are visualized in Figure 12. The figure illustrates the approximation error measured at various generations and for each algorithm. The error bars represent the 95% confidence interval. It can be seen that after 100 generations NSGAI and P3GA converge to a solutions with similar approximation errors for both test problems. As can be seen in the figure, NSGAI converges faster for test problem ZDT1. Figure 13 is an illustration of the P3GA Pareto-optimal solutions for test problem DTLZ2 at (a) 10 generations and (b) 100 generations for a single trial. The black and white mesh represents the true nondominated frontier while the approximation is shown in grayscale. For test problem DTLZ2, the approximation is worse at 100 generations than it was at 10 generations; this agrees with the results in Figure 12b. However, the Pareto-optimal solutions are closer to the true nondominated frontier after 100 generations.

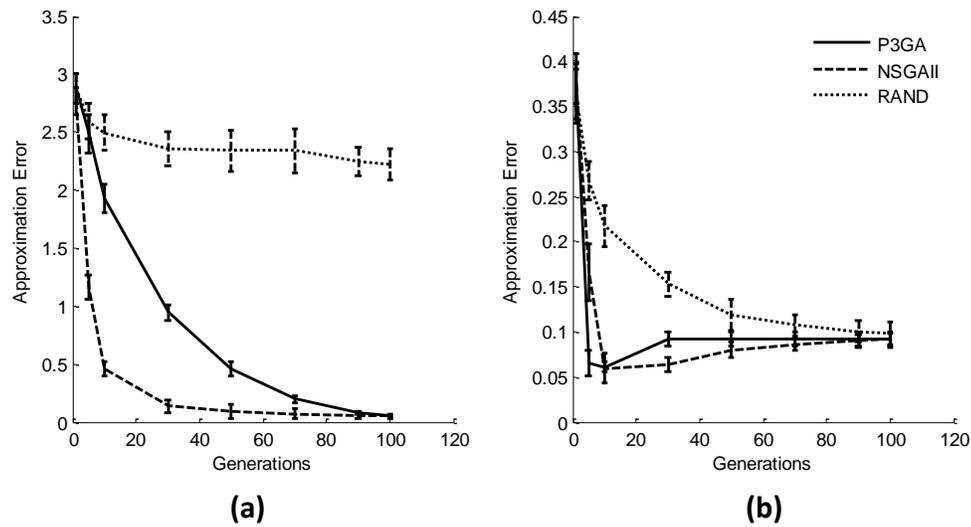


Figure 12. Hausdorff error as a function of number of generations for test problems ZDR1 (a) and DTLZ2 (b).

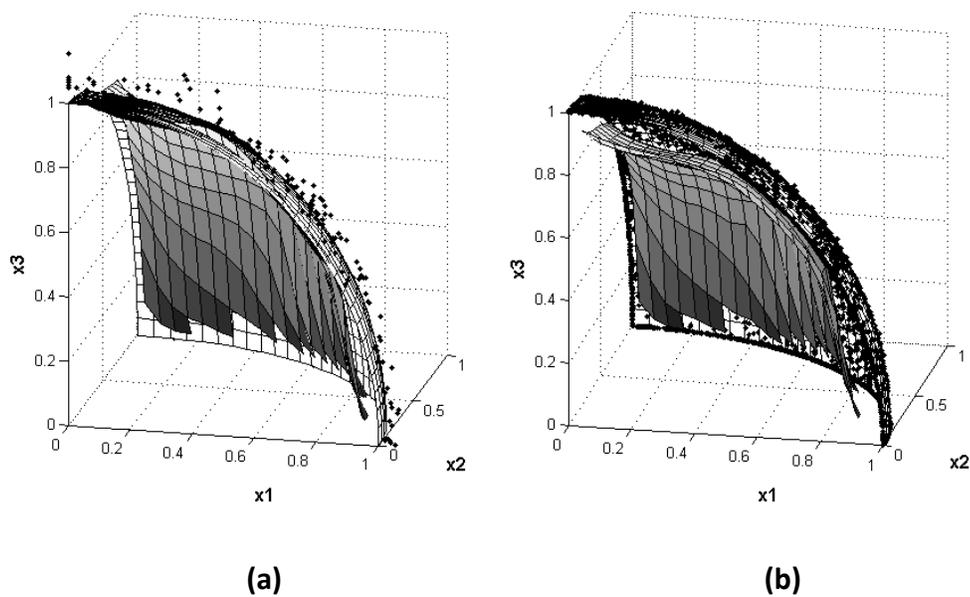


Figure 13. Illustration of P3GA Pareto-optimal solution for test problem DTLZ2 at (a) 10 generations and (b) 100 generations. The true Pareto frontier is shown in black and white the SVDD approximation is shown in grayscale.

6.5. Simulation Results for the Parameterized Pareto Case

For parameterized Pareto test problems, two algorithms were compared: RAND and P3GA. In this demonstration NSGAI was not considered because of the high computational expense of applying any traditional MOGA to case with parameters, see Section 4. The algorithms were executed 15 times on each test problem. Each simulation was carried out using the simulation parameters in the previous section. The test problem results are visualized in Figures 14-16. The figures illustrate the approximation error measured at various generations and for different values of the kernel function parameter q . The error bars represent the 95% confidence interval. Figure 17 is an illustration of the vertical distance between the true nondominated frontier and the approximations generated through (a) P3GA and (b) random approach for test problem 1a at 100 generations and $q = 6$. The figure is provided for the benefit of the reader rather than as an accurate indication of the approximation error.

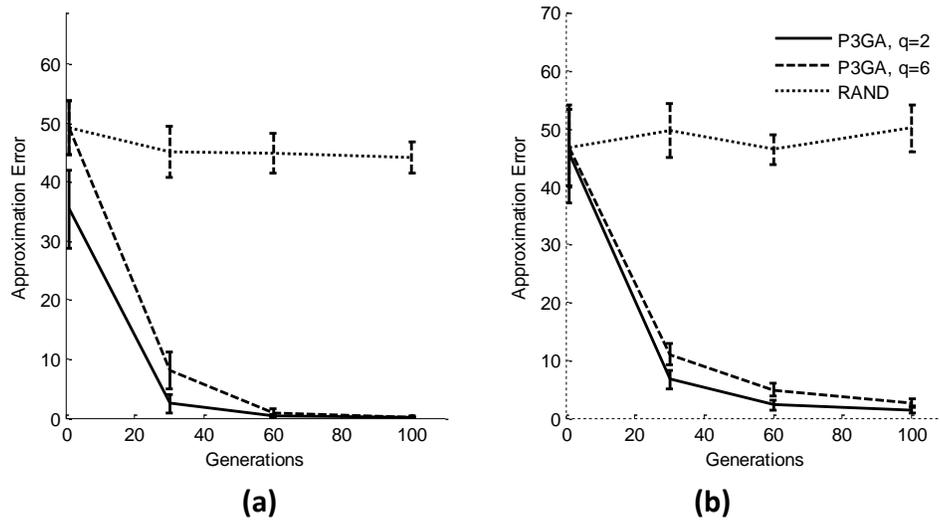


Figure 14. Mean Hausdorff distance as a function of number of generations for test problem 1, versions (a) and (b), at different values of the q parameter.

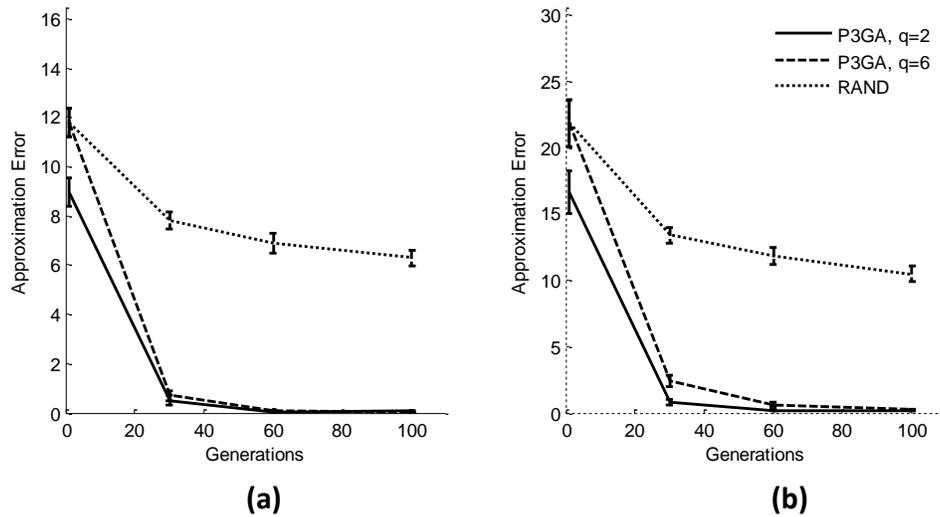


Figure 15. Mean Hausdorff distance as a function of number of generations for test problem 2, versions (a) and (b), at different values of the q parameter.

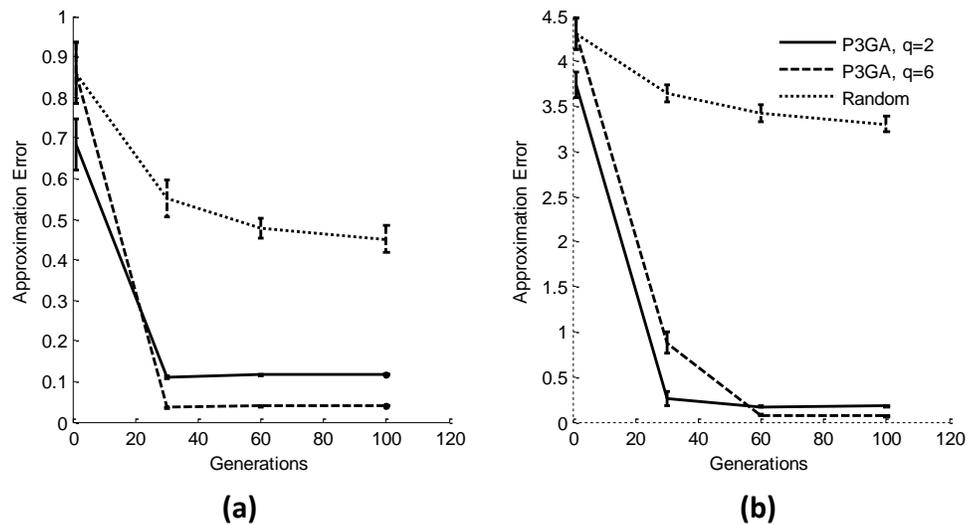


Figure 16. Mean Hausdorff distance as a function of number of generations for test problem 3, versions (a) and (b), at different values of the q parameter.

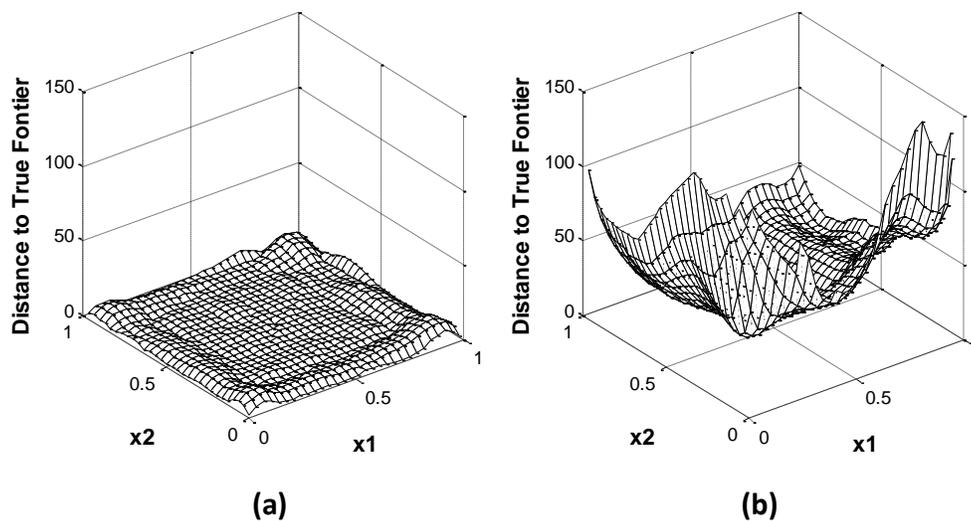


Figure 17. Illustration of the vertical distance between the true parameterized Pareto frontier and the approximations generated through (a) P3GA and (b) random approach for test problem 1a at 100 generations and $q = 6$.

6.6. Engineering Case Study

In this section, a utility vehicle (UV) design problem is used to demonstrate the use of P3GA in a systems design scenario. The UV is a four wheeled vehicle intended for off road use. In this design scenario, a suspension manufacturer wishes to characterize a suspension system using a TCM. The TCM is then communicated to the customer, the UV manufacturer, who uses the TCM to optimize the UV design. Our focus here is to demonstrate how the P3GA algorithm along with TCMs can be used in the design process rather than on the design results. This design demonstration has three stages: (1) generate a TCM of the suspension using P3GA, (2) search the TCM for optimal target values, and (3) use engineering optimization techniques to search for the design variable settings that correspond to the target values.

6.6.1. TCM Generation

In this demonstration it is assumed that the suspension designers have an implementation concept for the suspension and wish to characterize its achievable range of performance. Specifically, the designers wish to characterize a suspension system composed of a helical spring and a shock absorber. The designers have available a nearly continuous range of decision variables

d : Spring wire diameters [m],

D : Spring coil diameters [m],

p : Spring pitch [m],

N : Number of active coils [\cdot]

available to them. They wish to determine the achievable range (or parameterized Pareto frontier) of

$$C = F_1(\mathbf{x}) \quad : \quad \text{Suspension cost [USD]},$$

$$N_f = F_2(\mathbf{x}) \quad : \quad \text{Factor of safety } [·],$$

$$k = F_3(\mathbf{x}) \quad : \quad \text{Spring constant [N/m]},$$

$$c = F_4(\mathbf{x}) \quad : \quad \text{Damping ratio [Ns}^2\text{/m]}.$$

where $\mathbf{x} = [d, D, p, N]$ are the decision variables. The suspension cost is modeled as the sum of the component cost plus the assembly cost. The spring constant and factor of safety are determined using standard engineering models and an expected range of spring compression [34]. The available damping ratios are determined from catalog data. Given the set of component level attributes, $\mathbf{y}_{susp} = [C, N_f, k, c]$, the designers knows they will always prefer to lower cost, C , and increase the factor of safety, N_f . However, the designers have no established preference direction for the spring constant, k , or the damping ratio, c , since they do not know the design details of the entire UV system. The parameter attributes are denoted as $\mathbf{p} = [F_3(\mathbf{x}), F_4(\mathbf{x})]$. Suppose that \mathbf{S} is nonempty feasible region of the decision variable space. The feasible parameter region is then, $\mathbf{L}^2 = [\mathbf{L}_1 \times \mathbf{L}_2]$, where $\mathbf{L}_l = \{F_{2+l}(\mathbf{x}) : \mathbf{x} \in \mathbf{S}\}$. With this notation, the search problem for the nondominated frontier can be formulated as

$$\forall \mathbf{p} \in \mathbf{L}^2:$$

$$\underset{\mathbf{x} \in \mathbf{S}}{\text{Maximize}} \quad \mathbf{F}(\mathbf{x}) = [-F_1(\mathbf{x}), F_2(\mathbf{x})]^T \quad (11)$$

$$\text{subject to} \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m,$$

$$F_3(\mathbf{x}) = p_1,$$

$$F_4(\mathbf{x}) = p_2$$

where $g_i(\mathbf{x})$ represents the physical constraints in the decision variable space, e.g. lengths must be nonnegative. Using P3GA, the designers are able to approximate the solution to this search problem in the form of a TCM.

6.6.2. Searching the TCM

Once the designers have generated a TCM for the suspension, denoted $T(\mathbf{y}_{susp}) \leq R$, they are able to communicate the achievable range of performance to their customers. In this scenario, the customer is the manufacturer of the UV. In the interest of scope, all design parameters not pertaining to the suspension are fixed, e.g. vehicle mass, moment of inertia, etc. The UV suspension is assumed to be identical on the left and right sides so that a half car model can be used to simulate the vehicle vibration response. The suspension system models, F_1, \dots, F_4 , are not made available to the UV manufacture since they contain proprietary information that the suspension manufacturer does not want to divulge. The UV manufacturer wishes to tradeoff between component level attributes, \mathbf{y}_{susp} , for the front and rear suspensions $\mathbf{y}_{susp,f}$ and $\mathbf{y}_{susp,r}$, respectively. The component level attributes of interest for the UV system are then $\mathbf{y} = [\mathbf{y}_{susp,f}, \mathbf{y}_{susp,r}]$. The environmental variables determined by the manufacturer are

M : Mass of the UV [kg],

I_{yy} : Moment of inertia about the traverse direction[kgm²],

L_f : Front hub displacement from driver [m],

L_r : Rear hub displacement from driver[m],

y_{road} : Expected road profile [m].

To simplify notation, let $\mathbf{E} = [M, I_{yy}, L_f, L_r, y_{road}]$. Given the component level attributes and environmental variables, the UV manufacturer can compute three system level attributes:

$R = F_5(\mathbf{y}, \mathbf{E})$: Risk of injury due to vibrations,

$C_{sys} = F_6(\mathbf{y})$: Total cost of the suspension system,

$N_{sys} = F_7(\mathbf{y})$: Minimum system safety factor.

The risk of injury due to vibrations is determined according to the International Standard ISO 2631-5 which is a standard for quantifying whole-body vibration containing multiple shocks in relation to human health [35]. The cost of the suspension system is the total cost of the springs and dampers used in the four wheeled UV. The minimum system factor of safety is simply minimum factor of safety in the suspension system.

To be compatible with optimization algorithms, the UV designers must formalize their decision-making preferences in a computer-interpretable form. This task can be accomplished by using a utility function. A utility function, denoted $u(\mathbf{z})$, is a scalar function that relates system-level attributes to a utility value that designers seek to maximize. In this scenario the designers wish to minimize risk of injury and cost, and maximize the minimum safety factor. Therefore, the system level attributes are $\mathbf{z} = [F_5(\mathbf{y}, \mathbf{E}), F_6(\mathbf{y}), F_7(\mathbf{y})]$. Designers define a utility function such that attribute vectors that are more preferred lead to larger utilities. There are multiple approaches by which designers can do this [36]. Multi-attribute utility theory (MAUT) is suggested for this

step, where one elicits a utility function by answering a series of questions involving hypothetical choices involving lotteries [11]. However, the choice of decision rule is left to the designer. Once the designer has determined the utility function, the optimization problem can be formulated as follows:

$$\begin{aligned}
 \mathbf{y}^* &= \underset{\mathbf{y}}{\operatorname{argmax}} u([F_5(\mathbf{y}, \mathbf{E}), F_6(\mathbf{y}), F_7(\mathbf{y})]) \\
 \text{subject to } & T(\mathbf{y}_{\text{susp},f}) \leq R, \\
 & T(\mathbf{y}_{\text{susp},f}) \leq R, \\
 & g_{\text{sys},i}(\mathbf{y}, \mathbf{E}) \leq 0 \quad i = 1, 2, \dots, m.
 \end{aligned} \tag{12}$$

The constraint $g_{\text{sys},i}(\mathbf{y}, \mathbf{E}) \leq 0$ is imposed on the component level attributes by the system level constraints of the problem, e.g. the vehicle must be leveled. The constraint function $T(\cdot) \leq R$ represents the TCM generated using P3GA, where R is the hypersphere radius. The output of this search problem, \mathbf{y}^* , are the target attribute values for the components during detailed design optimization.

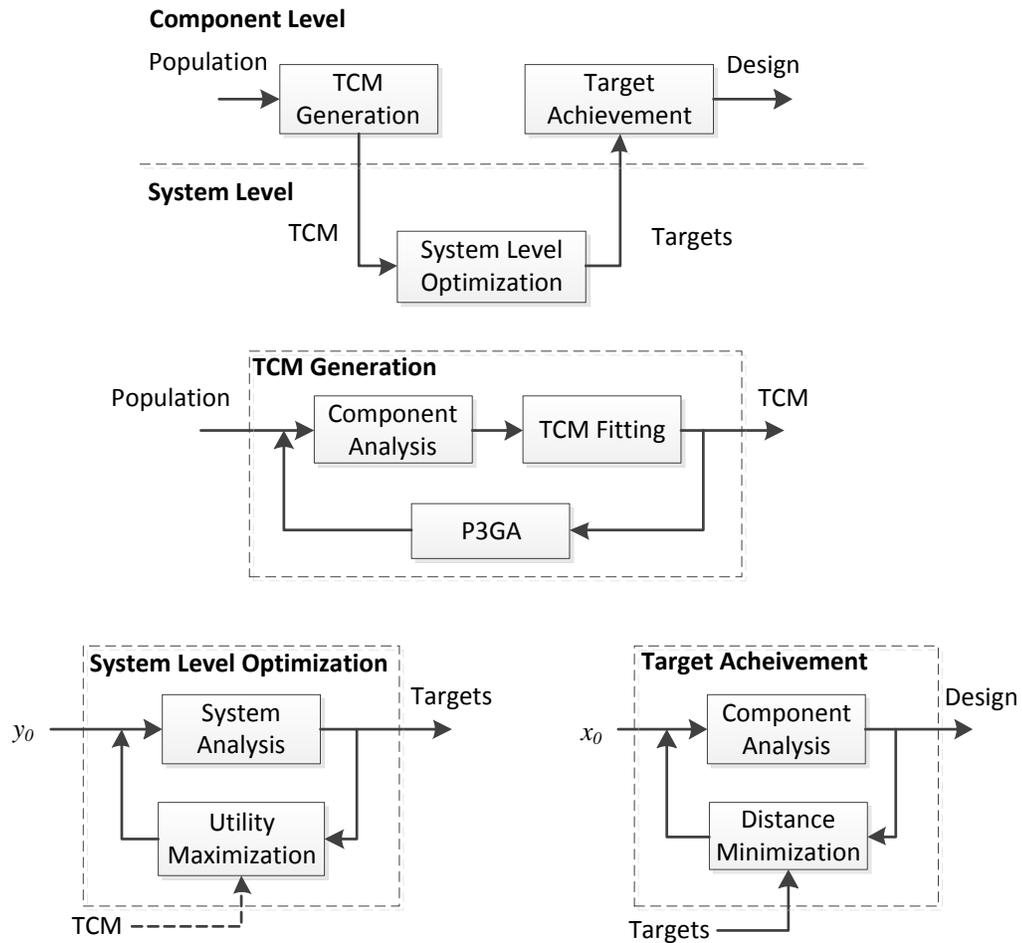


Figure 18. Illustration of the general design scenario using TCMs. No iteration between component and system level. The initial guess to the system and component levels are y_0 and x_0 , respectively.

6.6.3. Target Achievement

Once the target attributes, $\mathbf{y}^* = [\mathbf{y}_{susp,f}^*, \mathbf{y}_{susp,r}^*]$ have been identified by the UV manufacturer, the suspension manufacturer can use standard engineering optimization techniques to identify the corresponding design variable settings. The search problem can be formulated as a target-achievement problem, where the design

objective is to minimize the deviation from the target attribute vector. The target-achievement problem for the front suspension can be formulated as

$$\begin{aligned} \mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} & \| \mathbf{y}_{susp,f}^* - \mathbf{F}_{sys}(\mathbf{x}) \| \\ \text{subject to} & \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m, \end{aligned} \quad (13)$$

where $\mathbf{F}_{sys}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), F_3(\mathbf{x}), F_4(\mathbf{x})]$, and \mathbf{x}^* corresponds to the vector of component level decision variables for the front suspension. This search problem can be repeated similarly for the rear suspension target attributes, $\mathbf{y}_{susp,r}^*$. Figure 18 is an illustration of this design procedure. Note that only the TCM (description of capabilities) and the targets are communicated across the component-system level boundary. Also, there is no iteration between the component and system levels.

6.7. Engineering Case Study Results

The primary purpose of this engineering example is to demonstrate how P3GA and TCMs can be used to support design. However, the engineering example is also used to investigate the performance of P3GA. A TCM of the suspension was generated according to Eq. (11). The following parameter settings were used for P3GA:

Number of generations	:	100
Population size	:	200
Crossover rate	:	0.8
Mutation rate	:	0.01
Kernel function q parameter	:	6

In the engineering case study, the true nondominated frontier is not known; consequently, the mean Hausdorff distance between the true and approximated

nondominated frontiers cannot be determined directly. The general approach used in this section is to first find a set of points, $P = \{p_1, p_2, \dots, p_n\}$, on the true nondominated frontier. Given a point, p on the true nondominated frontier and the approximated nondominated frontier S , distance, $e(p, S)$, is defined as

$$e(p, S) = \min_{p' \in S} d(p, p'), \quad (14)$$

where $d(\cdot)$ is the Euclidean distance between two points in E^n and n is the number of dimensions. The mean distance between the points on the true nondominated frontier and the approximation S is

$$E_m(P, S) = \frac{1}{n} \sum_{i=1}^n e(p_i, S). \quad (15)$$

This approximation error metric is similar to the forward mean Hausdorff distance between the true nondominated frontier and the approximation S .

To find a point on the true nondominated frontier, the design problem was solved using the fully integrated approach. The fully integrated optimization (FIO) problem is an optimization of the combined suspension and UV systems; formulated as follows:

$$\begin{aligned} & \underset{\mathbf{x}_{sys}}{\text{Maximize}} && u(\mathbf{z}) \\ & \text{subject to} && g_i(\mathbf{x}_f) \leq 0, \quad i = 1, 2, \dots, m, \\ & && g_i(\mathbf{x}_f) \leq 0, \quad i = 1, 2, \dots, m, \end{aligned} \quad (16)$$

$$\text{where } \mathbf{z} = [F_5(\mathbf{y}, \mathbf{E}), F_6(\mathbf{y}), F_7(\mathbf{y})],$$

$$\mathbf{y} = [\mathbf{y}_{susp,f}, \mathbf{y}_{susp,r}],$$

$$\mathbf{y}_{susp,f} = [F_1(\mathbf{x}_f), F_2(\mathbf{x}_f), F_3(\mathbf{x}_f), F_4(\mathbf{x}_f)],$$

$$\mathbf{y}_{susp,r} = [F_1(\mathbf{x}_r), F_2(\mathbf{x}_r), F_3(\mathbf{x}_r), F_4(\mathbf{x}_r)],$$

$$\mathbf{x}_{sys} = [\mathbf{x}_f, \mathbf{x}_r]$$

where \mathbf{x}_f and \mathbf{x}_r are the front and rear suspension decision variables $[d, D, p, N]$, respectively. Solving the scalar optimization problem at the system-level results in a single point on the Pareto frontier in the system-level attribute space. The corresponding component-level attribute vector, \mathbf{y} , then lies on the parameterized Pareto frontier.

In order to generate multiple points on the true nondominated frontier, an approach similar to the weighted sum method was used [37]. The weighted sum method scalarizes a set of objectives into a single objective by premultiplying each objective with a user specified weight. The weights are varied to generate multiple solutions on the nondominated frontier. A drawback of the weighted sum approach is that it is impossible to obtain points on non-convex portions of the Pareto frontier [38]. To address this issue, a more general multilinear utility function is used instead [11]. Yet another difficulty of this approach is that varying the weights consistently and continuously may not necessarily result in an even distribution of Pareto-optimal points and a complete representation of the Pareto frontier [38]. Das and Dennis present the *normal boundary intersection (NBI) method* to address this limitation [39]. This method provides a means for obtaining an even distribution of Pareto-optimal points. However, an even distribution of Pareto-optimal points at the system level does not guarantee an even distribution of parameterized Pareto-optimal points at the component level. For this demonstration, it is assumed that the parameterized Pareto-optimal sites found using randomly generated multilinear utility functions are a sufficient representation of the

parameterized Pareto frontier. Developing a general approach for measuring the quality of a parameterized Pareto frontier approximation is outside of the scope of this thesis.

For this demonstration, 200 multi-linear utility functions were generated randomly. Then the design problem, Eq. (16), was solved for each utility function using the FIO approach. Ten random restarts were performed for each optimization run. The component attribute vector, $\mathbf{y} = [\mathbf{y}_{susp,f}, \mathbf{y}_{susp,r}]$, was recorded for each trial. The $\mathbf{y}_{susp,f}$ attribute vectors combined with the $\mathbf{y}_{susp,r}$ attribute vectors correspond to 400 sites on the true parameterized Pareto frontier in the component attribute space. The next step is to determine the minimum distance, $e(p, S)$, from the point, p , on the true parameterized Pareto frontier to the suspension TCM boundary, S . It is important to note that the TCM boundary also includes dominated points, see Figure 3. However, the TCM boundary point nearest a true parameterized Pareto-optimal point is likely also on the predicted parameterized Pareto frontier. With this assumption, Eq. (14) is approximated as

$$\begin{aligned} e(p, S) &= \min_{p'} d(p, p'), \\ \text{subject to } T(p') &= R, \end{aligned} \tag{17}$$

where $T(p')$ is the distance of the feature space image of point p' to the centroid of the hypersphere that encompasses the training data, and R is the radius of that hypersphere. The constraint $T(p') = R$ forces the solution to be on the boundary of the hypersphere, i.e., approximate parameterized Pareto frontier. All of the data is centralized from $[-1, 1]$ prior to determining $e(p, S)$ using the upper and lower bounds of the training data. Not performing this step would give a disproportionate amount of importance to variables

with large magnitudes, e.g., cost, over ones with low magnitudes, e.g., risk of injury. Finally, the mean distance is found using Eq. (15). The results are illustrated in Figure 19. The error bars represent the 95% confidence interval. Figure 19a illustrates the mean distance between the parameterized Pareto-optimal points and the approximated parameterized Pareto frontier at various generations. As can be seen in the figure, the error metric approaches zero rapidly. Figure 19b illustrates the corresponding mean constraint violation of Eq. (17). The low constraint violations—on the order of 10^{-8} —indicate that the solutions found lie on the boundary of the domain description.

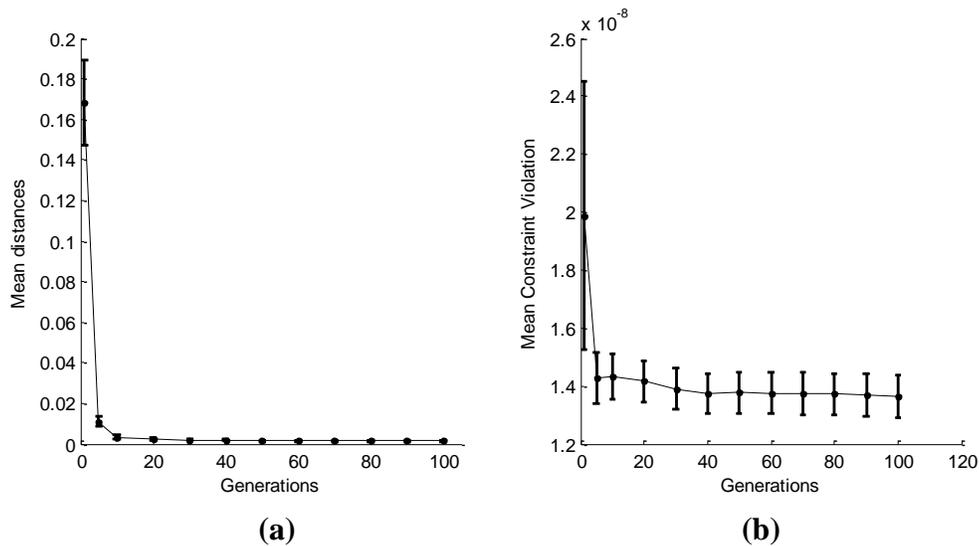


Figure 19. Illustration of (a) the mean distance between the parameterized Pareto-optimal points and the approximated parameterized Pareto frontier at various generations and (b) the corresponding mean constraint violation of Eq. (17).

7. DISCUSSION

For the classical Pareto case, a comparison of NSGAI, RAND, and P3GA was performed using two test problems from the literature, i.e., ZDT1 and DTLZ2. For this comparison, RAND serves as a point of reference. The proposed algorithm, P3GA, draws extensively on concepts from NSGAI, i.e., nondominated sorting and Crowded Tournament Selection. The principle difference between these two algorithms is that P3GA relies on the concept of *predicted* dominance. The purpose of the comparison between P3GA and NSGAI is to help validate the use of predicted dominance rather than to determine whether one algorithm outperforms another in the classical Pareto case. The principle result from this comparison is that for the test problems considered, P3GA and NSGAI converged to solutions with similar approximation error. This result is strong evidence that the use of predicted dominance in GAs is valid. As can be seen in Figure 12a, the approximation error for both P3GA and NSGAI decreased with the number of generations until about 10 generations; after which, the error increased slightly before leveling off. This result agrees with the result illustrated in Figure 13. After 10 generations, adding more sites to the true nondominated frontier “pushes” the approximation further away from the true nondominated frontier. A potential solution is to simply increase the kernel function q parameter; however, a thorough investigation into the effects of the q parameter on the performance of P3GA is outside of the scope of this thesis.

For the parameterized Pareto case, P3GA was compared only to the random approach since no efficient algorithm for modeling the parameterized Pareto frontier has been proposed in the literature. Although it is possible to apply MOOs to this problem, (see Section 3.2) the computational expense of applying this approach can be prohibitive even for relatively simple problems. As expected, P3GA significantly outperformed the random approach in the test problems considered. The poor performance of RAND in each test problem is a result of the factors introduced to cause difficulty in converging to the nondominated frontier. As can be seen in Figures 14-16, adjusting the kernel function q parameter affects the performance of P3GA. In the test problems considered, P3GA converged more slowly with the larger q parameter value. This was expected since increasing the q parameter results in a "tighter" domain description.

In problem $1a$, P3GA approaches the true nondominated frontier faster than in problem $1b$. As can be seen in Table 1, the principle difference between $1a$ and $1b$ is that $1b$ features local parameterized Pareto attractors while $1a$ does not. This indicates that P3GA may have difficulty converging to the nondominated frontier in the presence of local parameterized Pareto frontier attractors. However, P3GA still significantly outperforms RAND.

In problems $2a$ and $2b$, P3GA approaches the true nondominated frontier about the same rate. The mathematical formulation (see appendix) for $2a$ and $2b$ the same, the only difference being that problem $2a$ has two parameter values while $2b$ has only one. This causes the discontinuous frontier illustrated in Figure 10. This result indicates that the discontinuity in the frontier had little effect on the performance of P3GA in this case.

In problems *3a* and *3b*, P3GA again approaches the true nondominated frontier about the same rate. The principle difference between the two test problems is that *3b* features biased solution density away from the nondominated frontier. This result indicates that biasing the solution density has little effect on the performance of P3GA in this case.

The engineering example is provided primarily to demonstrate how P3GA along with TCMs can be used to support design space exploration. Using TCMs, component level designers are able to efficiently communicate their capabilities to the system level. At the system level, designers can search the space of capabilities in terms of only those variables that relate component performance to system performance. Once the system level designer has selected target component level attributes, the component designer solves a target achievement problem. This design approach eliminates the need for system-level and component-level designers to share complex engineering models and does not require potentially costly iterations between the component and system level designers. The results from the engineering problem indicate that P3GA is able to generate an accurate representation of the component capabilities for the engineering case study.

8. SUMMARY

In this thesis, an approach for generating an approximation of the parameterized Pareto frontier is presented. The algorithm uses fundamental concepts from multi-objective genetic algorithms and machine learning to search for solutions near the parameterized Pareto frontier. The algorithm borrows heavily from MOGAs in the literature but uses a novel update rule, i.e., predictive dominance. In the proposed algorithm, nondominated points are those that are not parametrically Pareto dominated by any solution that is predicted to be feasible. The algorithm may not be tractable for very large numbers of parameter attributes. The algorithm was demonstrated on two classical Pareto test problems from the literature and four parameterized Pareto test problems. In the classical Pareto case, P3GA performed similarly to NSGAI. In the parameterized Pareto test case, the P3GA algorithm significantly outperformed the random approach. These test cases provide evidence that the proposed approach can be effective in practice.

REFERENCES

- [1] Sobek II, D. K., Ward, A. C., and Liker, J. K., 1999, "Toyota's Principles of Set-Based Concurrent Engineering," *Sloan Management Review*, **40**(2), pp. 67-83.
- [2] Ward, A. C., 1989, "A Theory of Quantitative Inference Applied to a Mechanical Design Compiler," Ph.D. thesis, MIT, Boston.
- [3] Davis, E., 1987, "Constraint Propagation with Interval Labels," *Artificial Intelligence*, **32**(3), pp. 281-331.
- [4] Hazelrigg, G. A., 1996, *Systems Engineering: An Approach to Information-Based Design*, Prentice-Hall, Upper Saddle River, NJ.
- [5] Ferguson, S., Gurnani, A., Donndelinger, J., and Lewis, K., 2005, "A Study of Convergence and Mapping in Preliminary Vehicle Design," *International Journal of Vehicle Systems Modelling and Testing*, **1**(1-3), pp. 192-215.
- [6] Mattson, C. A., and Messac, A., 2003, "Concept Selection Using S-Pareto Frontiers," *AIAA Journal*, **41**(6), pp. 1190-1198.
- [7] Mattson, C. A., and Messac, A., 2005, "Pareto Frontier Based Concept Selection under Uncertainty, with Visualization," *Optimization and Engineering*, **6**(1), pp. 85-115.
- [8] Deb, K., 2001, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley & Sons, Chichester.
- [9] Malak, R. J., 2008, "Using Parameterized Efficient Sets to Model Alternatives for Systems Design Decisions," Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA.
- [10] Parker, R. R., and Malak, R. J., 2011, "Technology Characterization Models and Their Use in Designing Complex Systems," *Proc. ASME 2011 International Design Engineering Technical Conference and Computers and Information in Engineering Conference*, Washington, DC.
- [11] Keeney, R. L., and Raiffa, H., 1993, *Decisions with Multiple Objectives*, Cambridge University Press, Cambridge, UK.
- [12] Malak, R. J., and Paredis, C. J. J., 2010, "Using Parameterized Pareto Sets to Model Design Concepts," *Journal of Mechanical Design*, **132**(4).

- [13] Malak, R. J., Tucker, L., and Paredis, C. J. J., 2008, "Composing Tradeoff Models for Multi-Attribute System-Level Decision Making," Proc. ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, New York.
- [14] Malak, R. J., and Paredis, C. J. J., 2009, "Using Support Vector Machines to Formalize the Valid Input Domain of Models in Data-Driven Predictive Modeling for Systems Design," Proc. ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, San Diego, CA.
- [15] Galvan, E., and Malak, R. J., 2010, "Using Predictive Modeling Techniques to Solve Multilevel Systems Design Problems," Proc. 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, TX.
- [16] Zalzala, A. M. S., and Fleming, P. J., 1997, *Genetic Algorithms in Engineering Systems*, The Institute of Electrical Engineers, London, United Kingdom, England.
- [17] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-II," IEEE Transactions on Evolutionary Computation, **6**(2), pp. 182-197.
- [18] Abbass, H. A., 2002, "The Self-Adaptive Pareto Differential Evolution Algorithm," Proc. Congress on Evolutionary Computation, pp. 831-836.
- [19] Horn, J., Nafpliotis, N., and Goldberg, D. E., 1994, "A Niche Pareto Genetic Algorithm for Multiobjective Optimization," Proc. IEEE World Congress on Computational Intelligence First IEEE Conference on Evolutionary Computation, pp. 82-87.
- [20] Tax, D. M. J., and Duin, R. P. W., 1999, "Support Vector Domain Description," Pattern Recognition Letters, **20**(11-13), pp. 1191-1199.
- [21] Scholkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., and Platt, J., 2000, "Support Vector Method for Novelty Detection," Proc. Advances in Neural Information Processing Systems, pp. 582-588.
- [22] Wolfe, P., 1961, "A Duality Theorem for Nonlinear Programming," Management Science, **12**(5), pp.385-390.
- [23] Cauwenberghs, G., and Poggio, T., 2000, "Incremental and Decremental Support Vector Machine Learning," Proc. Neural Information Processing Systems, pp. 409-415.

- [24] Roach, E., Parker, R. R., and Malak, R. J., 2011, "An Improved Support Vector Domain Description Method for Modeling Valid Search Domains in Engineering Design Problems," Proc. ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, DC.
- [25] Zitzler, E., and Thiele, L., 1998, "Multiobjective Optimization Using Evolutionary Algorithms — a Comparative Case Study," Proc. Conference on Parallel Problem Solving from Nature, Amsterdam.
- [26] Ying, G., Lei, S., and Pingjing, Y., 2000, "Study on Multi-Objective Genetic Algorithm," Proc. 3rd World Congress on Intelligent Control and Automation, 2000, **1**, pp. 646-650.
- [27] Hallam, N., Blanchfield, P., and Kendall, G., 2005, "Handling Diversity in Evolutionary Multiobjective Optimization," Proc. IEEE Congress on Evolutionary Computation, 2005, **3**, pp. 2233-2240.
- [28] Deb, K., Thiele, L., Laumanns, M., Zitzler, E., Abraham, A., Jain, L., and Goldberg, R., 2005, *Evolutionary Multiobjective Optimization*, Springer Berlin Heidelberg. Chap. 6.
- [29] Deb, K., Pratap, A., and Meyarivan, T., 2000, "Constrained Test Problems for Multi-Objective Evolutionary Optimization", Proc. First International Conference on Evolutionary Multi-Criterion Optimization, pp. 284-298.
- [30] Deb, K., 1999, "Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, **7**(3), pp. 205-230.
- [31] Knowles, J. D., Thiele, L., and Zitzler, E., 2006, "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," Technical Report no. 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.
- [32] Cignoni, P., Rocchini, C., and Scopigno, R., 1998, "Metro: Measuring Error on Simplified Surfaces," *Computer Graphics Forum*, **17**(2), pp. 167-174.
- [33] Aspert, N., Santa-Cruz, D., and Ebrahimi, T., 2002, "Mesh: Measuring Errors between Surfaces Using the Hausdorff Distance," Proc. IEEE International Conference in Multimedia and Expo, 2002, Lausanne, Switzerland, **1**, pp. 705-708.
- [34] Shigley, J. E., and Mischke, C. R., 2001, *Mechanical Engineering Design*, McGraw-Hill, New York.

- [35] International Organization for Standardization, 2001, "Iso 2631-4:2001," <http://www.iso.org>
- [36] Kumar, D. K. D., Chen, W., and Kim, H. M., 2006, *Decision Making in Engineering Design*, American Society of Mechanical Engineers, New York, Chap. 18.
- [37] Zadeh, L., 1963, "Optimality and Non-Scalar-Valued Performance Criteria," *IEEE Transactions on Automatic Control*, **8**(1), pp. 59-60.
- [38] Das, I., and Dennis, J. E., 1997, "A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems," *Structural Optimization*, **14**(1), pp. 63-69.
- [39] Das, I., and Dennis, J. E., 1998, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems," *Siam Journal on Optimization*, **8**(3), pp. 631-27.

APPENDIX

For all test problems, the inputs are bound $0 \leq x_i \leq 1$ for $i = 1, \dots, n$.

Test Problem (1a)

$$f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]$$

where

$$f_1(\mathbf{x}) = \sqrt{\sin\left(\frac{x_1\pi}{2}\right)}$$

$$f_2(\mathbf{x}) = \sqrt{\cos\left(\frac{x_2\pi}{2}\right)}$$

$$g(\mathbf{x}) = 0.7 + \frac{9}{n-1} \sum_{i=3}^n x_i$$

$$h(\mathbf{x}) = 20 + 1.1f_1(\mathbf{x})^2 + 0.7f_2(\mathbf{x})^2 + 10 \cos(2.2\pi f_1(\mathbf{x})) + \cos(1.4\pi f_2(\mathbf{x}))$$

$$f_3(\mathbf{x}) = (1 + g(\mathbf{x}))h(\mathbf{x})$$

Outputs f_1 and f_2 are parameter attributes The nondominated frontier is described by $h(\mathbf{x})$. Here, $n = 10$.

Test Problem (1b)

$$f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]$$

where

$$f_1(\mathbf{x}) = \sqrt{\sin\left(\frac{x_1\pi}{2}\right)}$$

$$f_2(\mathbf{x}) = \sqrt{\cos\left(\frac{x_2\pi}{2}\right)}$$

$$g(\mathbf{x}) = 2 - \exp\left(-\left(\frac{x_3-0.2}{0.004}\right)^2\right) - 0.8 \exp\left(-\left(\frac{x_3-0.6}{0.4}\right)^2\right) + \frac{9}{n-1} \sum_{i=4}^n x_i$$

$$h(\mathbf{x}) = 20 + 1.1f_1(\mathbf{x})^2 + 0.7f_2(\mathbf{x})^2 + 10(\cos(2.2\pi f_1(\mathbf{x})) + \cos(1.4\pi f_2(\mathbf{x})))$$

$$f_3(\mathbf{x}) = (1 + g(\mathbf{x}))h(\mathbf{x})$$

Outputs f_1 and f_2 are parameter attributes. The true nondominated frontier is described by $h(\mathbf{x})$. Here, $n = 10$.

Test Problem (2a)

$$f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]$$

where

$$f_1(\mathbf{x}) = x_1$$

$$f_2(\mathbf{x}) = x_2$$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=3}^n x_i$$

$$h(\mathbf{x}) = 6 - 2f_1(\mathbf{x}) - \sin(3\pi f_1(\mathbf{x})) - 2f_2(\mathbf{x}) - \sin(3\pi f_2(\mathbf{x}))$$

$$f_3(\mathbf{x}) = g(\mathbf{x})h(\mathbf{x})$$

Outputs f_1 and f_2 are parameter attributes. The true nondominated frontier is described by $h(\mathbf{x})$. Here, $n = 10$.

Test Problem (2b)

The mathematical formulation for this problem is identical to test problem 2a except that only output f_1 is a parameter attribute; f_2 and f_3 are dominator attributes.

Test Problem (3a)

$$f_c(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]$$

where

$$f_1(\mathbf{x}) = \sqrt{\sin\left(\frac{x_1\pi}{2}\right)}$$

$$f_2(\mathbf{x}) = \sqrt{1 - x_2}$$

$$g(\mathbf{x}) = \sum_{i=3}^n x_i$$

$$u(y, g, l) = \frac{1}{4} \left(-\cos(2\pi l(y) (1 - g)) - 0.8\cos(2\pi l(y)(g - 1)) \right)$$

$$h(\mathbf{x}) = 1 + u(f_1(\mathbf{x}), 0.4, 2) + u(f_2(\mathbf{x}), 0.8, 5)$$

$$f_3(\mathbf{x}) = (1 + g(\mathbf{x}))h(\mathbf{x})$$

Outputs f_1 and f_2 are parameter attributes. The nondominated frontier is described by $h(\mathbf{x})$. Here, $n = 4$.

Test Problem (3b)

$$f_c(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]$$

where

$$f_1(\mathbf{x}) = \sqrt{\sin\left(\frac{x_1\pi}{2}\right)}$$

$$f_2(\mathbf{x}) = \sqrt{1 - x_2}$$

$$g(\mathbf{x}) = \sum_{i=3}^n x_i$$

$$u(y, g, l) = \frac{1}{4} \left(-\cos(2\pi l(y) (1 - g)) - 0.8\cos(2\pi l(y)(g - 1)) \right)$$

$$h(\mathbf{x}) = 1 + u(f_1(\mathbf{x}), 0.4, 2) + u(f_2(\mathbf{x}), 0.8, 5)$$

$$f_3(\mathbf{x}) = (1 + g(\mathbf{x}))h(\mathbf{x})$$

Outputs f_1 and f_2 are parameter attributes. The nondominated frontier is described by $h(\mathbf{x})$. Here, $n = 10$.

VITA

Name: Edgar Galvan

Address: 3123 TAMU
Department of Mechanical Engineering
Texas A&M University
College Station, Texas 77843

Email Address: e_galvan@tamu.edu

Education: B.S., Mechanical Engineering, Texas A&M University, 2010
M.S., Mechanical Engineering, Texas A&M University, 2012