

DETC2005-84852 (Draft)

A Study of Convergence and Mapping in Multiobjective Optimization Problems

Scott Ferguson
Ashwin Gurnani

Graduate Research Assistants
Department of Mechanical and Aerospace Engineering
University at Buffalo, Buffalo, NY

Joseph Donndelinger

Staff Research Engineer
Vehicle Development Research Lab
General Motors Research & Development Center
Warren, MI

Kemper Lewis

Associate Professor
Department of Mechanical and Aerospace Engineering
University at Buffalo, Buffalo, NY
Corresponding Author: kelewis@eng.buffalo.edu, (716) 645-2593 x2232

ABSTRACT

In this paper, we investigate the issue of convergence in multiobjective optimization problems when using a Multi-Objective Genetic Algorithm (MOGA) to determine the set of Pareto optimal solutions. Additionally, given a Pareto set for a multi-objective problem, the mapping between the performance and design space is studied to determine design variable configurations for a given set of performance specifications. The advantage of this study is that the design variable information is obtained without having to repeat system analyses. The tools developed in this paper have been applied to develop a Technical Feasibility Model (TFM) used by General Motors as well as a simple multiobjective optimization problem in this paper. The multi-objective problem is primarily used to illustrate the developed methodology.

1.0 INTRODUCTION

The goal of creating a Technical Feasibility Model is to develop a set of analytical tools to support the preliminary stages of the design process. Feasibility in this tool is assessed by ensuring that product specifications are mutually compatible from an engineering design perspective. This paper covers two issues critical to successful deployment of a Technical Feasibility Model. The first issue is the convergence of the multiobjective optimization problem when using a Multi-Objective Genetic Algorithm (MOGA) to generate a set of Pareto optimal solutions upon which the TFM is based. From a

business perspective, it is critical to understand the convergence behavior of the MOGA a priori so that adequate resources can be allocated to development of the TFM. The second issue is the nature of the correspondence between the Pareto-optimal solutions in the performance space and the corresponding variables and configurations in the design space. A point located in the performance space may map to many points in the design space. That is, the same performance may be obtained by multiple design configurations. Knowledge of this correspondence may be used to understand how slight changes in the performance space may change the configuration and values in the design space. It also lends insight into the robustness of solutions by providing the designer with several viable design alternatives for achieving a desired performance.

The study of convergence criteria is critical for computationally expensive problems since evaluation time could seriously inhibit the success of the preliminary design process. Mapping from the performance to design space is critical to the understanding how changes in product specifications affect the design configuration. This paper presents both the process used to evaluate the TFM development procedures in [1] and a discussion of the results of the evaluation. Section 2 provides background information into multiobjective optimization, convergence of MOGAs, and the issues pertaining to performance to design space mapping.

2.0 BACKGROUND

It has long been accepted in the engineering design community that product design can no longer be viewed with the perspective of reducing cost alone. Increased demands by consumers on products and processes as well as fierce competition amongst manufacturers have pushed the concept of multi-objective optimization as the methodology to be used for the design of new products. The challenge now is to design a product with low cost but at the same time satisfying other consumer demands such as a need for increased luxury and comfort in the use of the product. Designer also seek to provide attributes of “surprise and delight” in order to get the edge over competitors. Laws related to safety and quality of products further increase the number of objectives that need to be simultaneously satisfied in the design of modern day products. Thus, multi-objective optimization, which provides essential tools in achieving many goals simultaneously, is an important area of research and the primary focus of this paper [2].

Multi-objective problems rarely have a single solution and usually have a set of multiple points forming the solution set. These solutions, called Pareto-optimal solutions are those in which any improvement in one objective must result in the degradation of at least one other objective since the objectives conflict with each other [3]. Mathematically, a feasible design variable vector, \bar{x}' is Pareto optimal if and only if there is no feasible design variable vector, \bar{x} , with the characteristics shown in Eq. (1).

$$\begin{aligned} f_i(\bar{x}) &\leq f_i(\bar{x}') && \text{for all } i, i = 1 \dots n \\ f_i(\bar{x}) &< f_i(\bar{x}') && \text{for at least one } i, 1 \leq i \leq n \end{aligned} \quad (1)$$

where n is the number of objectives and the use of the *less than* operator indicates an improvement in an objective, since it is assumed that the objectives are minimized.

With the increase in availability of computation resources, heuristic optimization methods such as genetic algorithms that are essentially computationally intensive have been extended for the use of multi-objective problems. The advantage of using a Genetic Algorithm for multi-objective problems, called a Multi-Objective Genetic Algorithm, is that the final result is a set of multiple solutions that do not dominate each other. If the MOGA is run long enough, the solution set obtained can be approximated to be the Pareto set [4]. Knowledge of designs that make up the Pareto set is invaluable since these designs are the best solutions to the multi-objective problem.

The research presented in this paper has been conducted to aid in the preliminary phases of the vehicle development process. The proposed framework for this system is computationally intensive and incorporates evaluations in different software packages [5-6]. Since each evaluation of the objective functions is extremely expensive in terms of computation time, issues pertaining to the convergence of the MOGA to the Pareto set become very important. Some of these issues are related to the accuracy of the Pareto frontier, the spread of Pareto points and the existence of clusters since all these parameters depend on the convergence of the MOGA. Azarm [7] has developed

various metrics that enable the designer to either monitor the quality of the Pareto frontier or use it to compare the solution obtained from different multi-objective optimization methods. Zitzler [8] provides the C and S metrics that directly measure the convergence of Pareto solution set. Deb [9] has proposed a metric that evaluates the convergence of a solution set to a reference set while Veldhuizen [10] proposes an error ratio to determine if a solution set has converged to the true solution set. In this paper, a study is presented that compares non dominated solution sets, obtained by using smaller number of function evaluations, to the true Pareto set. This is critical since each function evaluation is computationally expensive, as mentioned earlier.

In addition to reducing the number of function evaluations to obtain the Pareto set, vehicle development teams also require knowledge of the relationships vehicle attributes and vehicle design parameters. The desired attributes or objective function values (also referred to as performance measures) for the new vehicle design are available *a priori* from marketing, as these attributes are developed to maximize customer satisfaction. To provide the design variable information corresponding to the desired performance measures, vehicle development engineers would need to work backwards within their analysis systems, which is already known to be computationally intensive. To avoid these additional analyses, up-front mapping of design variable values to the existing set of Pareto points could be used to determine the design configuration of the new vehicle.

Mapping of the performance space to the design space is not new to engineering design and has been recognized as a challenging task since the mapping can be one-to-many, with one objective function point mapping back to multiple design points [11]. Past work includes the use of a visualization technique called *Cloud Visualization* to determine design variable values for a given point in the performance space [12]. The use of design variable mapping has also been shown to accelerate the design process for a multi-piece propshaft [13]. Additionally, mapping between performance and design spaces is critical in morphing systems where changing from one optimal configuration to another can potentially create drastic changes in the design configuration [14]. In this work, data obtained from the MOGA for Pareto set generation is used to determine the design variable values of the new design using a mapping between the performance and design spaces.

Given the background to the work presented in this paper, Section 3 discusses in detail the MOGA convergence studies mentioned earlier in this section. Section 4 presents the theory used for the performance to design space mapping. Section 5 presents an application of the work in this research to a simple case study problem and Section 6 provides some concluding remarks and areas of future work.

3.0 MOGA CONVERGENCE

The first step of constructing a Technical Feasibility Model relies upon the usage of a Multi-Objective Genetic Algorithm (MOGA) to solve a multiobjective optimization problem. The solution to this problem is a set of non-dominated solutions that compose the Pareto frontier. Metamodeling techniques are then

used to fit a constrained second order polynomial to these Pareto points. This surface is then used to assess technical feasibility as well as the optimality of a given test point. To ensure that the entire frontier was populated, an exhaustive number of evaluations were used. For the purposes of this paper, the MOGA process used to create the Pareto frontier will be referred to as the ‘exhaustive’ MOGA. The solution of this exhaustive MOGA serves as the Pareto frontier benchmark when comparing different Pareto frontier solutions. However, such a large number of evaluations for a multiobjective system may result in extreme computational expense. Therefore, for such complex systems, completing such a large number of functional evaluations may not be practical, or even feasible.

The large number of evaluations used in the exhaustive MOGA raises a significant research question. This question addresses the extent to which the quality of the frontier is affected when changing the maximum allowed number of evaluations. By investigating the convergence of the MOGA, it may be possible to determine a tradeoff between the number of designs evaluated and the quality of the Pareto frontier. Understanding this tradeoff will enable the effective evaluation of problems of increased computational complexity. First, however, it is necessary to determine a method of comparing the results of different MOGA test cases. This method is described in the following steps:

1. Complete an exhaustive sampling of the model: As mentioned earlier, this results in a set of objective function values that are assumed to be the true Pareto set.
2. Specify indifference thresholds: This refers to the change in each objective function within which the designer is indifferent to all objective function values.
3. Discretize the performance space using the defined thresholds: Using the indifference thresholds to establish the discretization sizes for each objective, the performance space is divided into a collection of “hyperboxes” (for problems with more than 3 objectives). For a problem with 3 objectives, the performance space would be discretized into a set of equally sized rectangular cuboids, where the size of each cuboid is the indifference threshold value for each objective.
4. Represent the exhaustive MOGA as a collection of hyperboxes in the performance space: Using the discretized performance space, it is possible to visualize the resultant view of the Pareto frontier as seen in Figure 1.

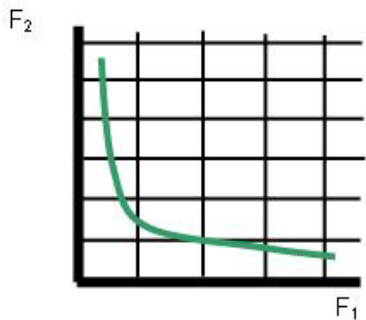


Figure 1. Representing the Pareto Frontier as a Collection of Hyperboxes

To determine if a hyperbox in the performance space is part of the Pareto frontier, at least one non-dominated design must be present in a given hyperbox. If there exist multiple design points in the same hyperbox, the design engineer is said to be indifferent to all these designs. Pictorially, this scenario is shown in Figure 2.

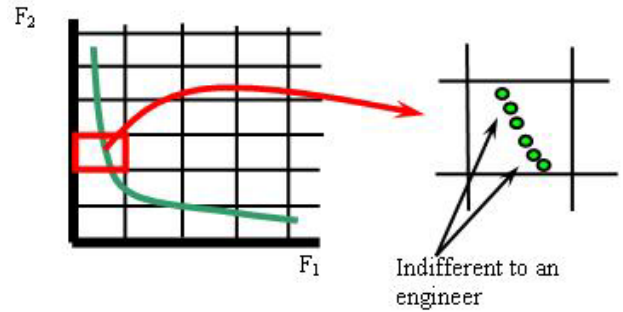


Figure 2. Identification of Indifferent Designs Within a Performance Space Hyperbox

For the problem shown in Figure 1, the hyperboxes populated by the Pareto points (determined using the MOGA) are shown in Figure 3.

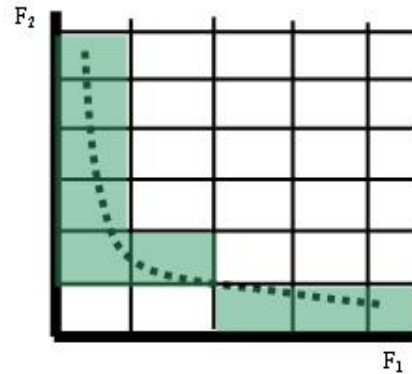


Figure 3. Identification of Hyperboxes Filled in the Performance space by the Pareto Frontier

5. Compare the results of other MOGA runs to the hyperbox solution set of the exhaustive study: MOGA cases using different number of function evaluations are investigated and compared to the exhaustive Pareto frontier. These cases were developed to analyze the tradeoff of maximum allowed evaluations to the quality of the Pareto frontier. The number of function evaluations available is treated as a constraint in the setup of the MOGA.

In order to effectively generate the best final population, an algorithm for implementing the MOGA when using the maximum number of available evaluations is developed. The first stage starts with a small initial population, and maintains a constant population size for a limited number of evaluations (i.e., a third of the available evaluations). This stage is designed to drive the members of the population to the Pareto frontier. However, by doing so, it is not guaranteed that the points of the population are evenly distributed along the frontier. To remedy this, the second stage allows the population of the MOGA to grow to accommodate all identified

non-dominated designs for the remaining number of design evaluations. Graphically, this is shown in Figure 4.

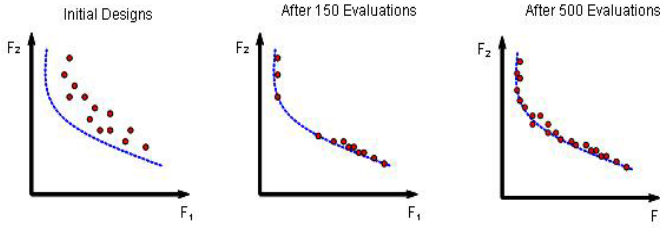


Figure 4. Depiction of MOGA for 500 Evaluation Test Case

As with the exhaustive MOGA, the filled hyperboxes in the performance space for each MOGA population are identified. Comparing these hyperboxes, the number of hyperboxes that are filled by both the exhaustive MOGA and each test case is recorded. As the number of design evaluations increases, so does the number of exhaustive MOGA hyperboxes filled by the test case. Therefore, a complete frontier that is captured with a smaller number of evaluations than the exhaustive MOGA is inherently more effective.

The ability to determine the true Pareto frontier of a problem allows for the generation of a Technical Feasibility Model. However, the feasibility of a test point and its optimality with respect to the Pareto frontier is only a portion of the information needed in the preliminary design process. Mapping a set of technical specifications to their location in the design space provides invaluable insight into how the system behaves, and how it will react to change. Mapping from the performance to design space is not one-to-one, however, and becomes a non-trivial task. An approach to address this issue is outlined in the next section.

4.0 PERFORMANCE TO DESIGN SPACE MAPPING

Development of the Pareto frontier representation allows the designer to determine if a new preliminary design concept is feasible and optimal. However, this information is incomplete, as it provides no knowledge of the design variables that compose that design. Understanding the relationship between the performance and the design space is the next logical progression in developing a preliminary design within the TFM.

The primary goal of this part of this section is to determine corresponding design variable information given a desired set of performance values for a multi-objective problem. Design variable information is desired in the form of a mean value and a design tolerance to allow for robust design. For the purpose of this study, indifference thresholds that have been defined earlier are used for the performance space. Using the indifference thresholds to establish the discretization sizes for each objective, the performance space is divided into a collection of hyperboxes as discussed earlier for the convergence approach.

Each performance space hyperbox maps to some region of the design space that is also discretized into hyperboxes. In order to

determine the corresponding design variable configuration for a given set of performance measure values, the hyperbox corresponding to the performance values is identified and mapped back to a design space hyperbox. The centroid of this mapped hyperbox is the design that would be used to obtain the desired performance measures, with the design tolerance range determined from the span of the hyperbox. The nature of mapping between performance space hyperbox and design space hyperbox can be of three types as discussed below.

Type 1: Individual Performance space hyperbox maps to one design space hyperbox: In this case, the centroid of the design space hyperbox is the design variable vector and the tolerance is half the discretization range. This is shown in Figure 5.

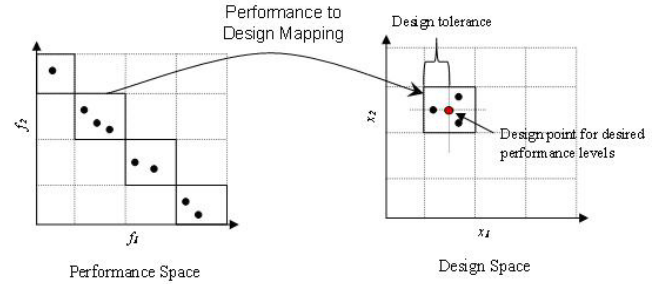


Figure 5. One Performance Space Hyperbox Mapped to One Design Space Hyperbox

Type 2: Performance space hyperbox maps to multiple, *adjacent* design space hyperboxes: In this case, a hyperbox can encompass all the mapped adjacent hyperboxes. The design variable values correspond to the centroid of this overlapping box with the tolerance values determined from the extremes of the enveloping hyperbox. This is shown in Figure 6.

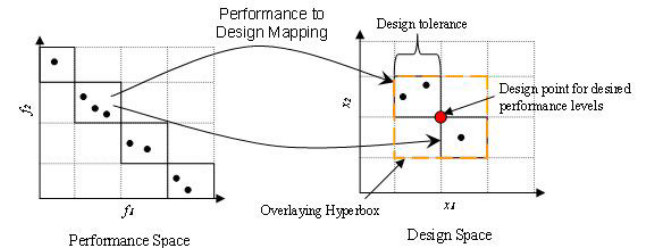


Figure 6. One Performance Space Hyperbox Mapped to Multiple Adjacent Design Space Hyperboxes

Type 3: Performance space hyperbox maps to multiple, *non adjacent* design space hyperboxes. As shown in Figure 7, one performance space hyperbox maps to different design space hyperboxes that are spaced out in the design space. It is hypothesized that it is meaningful to place an overlaying hyperbox over the scattered mapped design space hyperboxes. To validate this hypothesis, the following study is carried out.

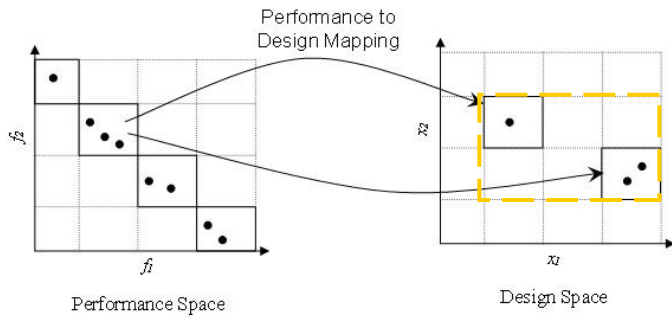


Figure 7. One Performance Space Hyperbox Mapped to Multiple, Non-Adjacent Design Space Hyperboxes

For a set of non adjacent, mapped design space hyperboxes, a hyperbox is placed in order to envelope these hyperboxes. Three design points are selected from this hyperbox and evaluated to determine their objective function values. If the evaluated objective function values fall into the same performance space hyperbox from where the design space hyperboxes were originally mapped, the hypothesis is assumed validated. The three points selected are the center of the box, a point corresponding to a quarter of the way in all dimensions of the hyperbox, and a point corresponding to three quarters of the way in all dimensions of the hyperbox. The 2-D representation is shown in Figure 8.

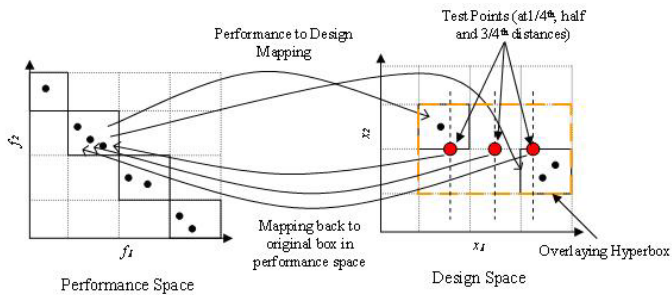


Figure 8. Determining Performance Space Hyperbox for Test Points in Overlaying Hyperbox

These three points generated shown in Figure 8 are evaluated to determine the corresponding objective function values. The objective function values are converted to indices that represent hyperboxes in the performance space. The performance space hyperboxes corresponding to these points are compared to the original Pareto frontier hyperboxes, and an exact or adjacent match is found.

The rationale in doing so is that a design that maps to an adjacent hyperbox might be very close to the original hyperbox in terms of actual objective function values. Additionally, it is ascertained that if a large number of designs (but not the entire set) map to the original or its adjacent performance space hyperbox, the hypothesis of placing an overlaying hyperbox in the design space is validated. This scenario is depicted in Figure 9 below.

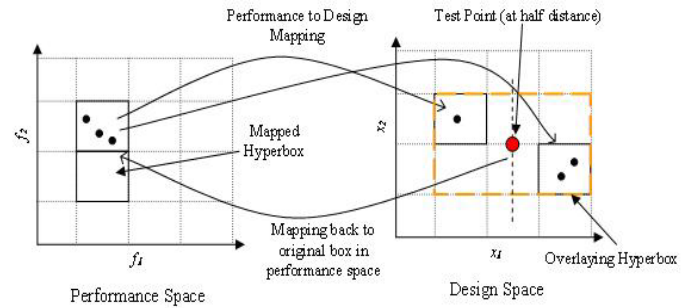


Figure 9. Test Point Mapping to Performance Space Hyperbox Adjacent to Original

Thus, for *Type 3* mapping, that is mapping of one performance space hyperbox to multiple, non adjacent design space hyperboxes, it is shown that a larger, overlaying hyperbox can be placed over all the design space hyperboxes. The design variable values and tolerances correspond to the center of this overlaying hyperbox. The procedure to determine the design variable values for a given set of performance measures from the determined mapping is described as follows:

- For the given performance measure values, use the bounds of the objective function values obtained from the MOGA and chosen discretization size to determine the performance space hyperbox indices.
- Compare indices of given performance space measures to existing indices in mapping data sheet.
- Obtain design variable values and tolerances for matching performance space hyperbox.

An important point of note here is that this map only contains indices of performance space hyperboxes that are populated with Pareto optimal designs. The rationale behind using just the Pareto optimal hyperboxes is the same as mentioned in the background section. Since the given performance measure values are known to maximize customer satisfaction, the point is assumed to be in the region of the Pareto set. Therefore, only design variable values for designs with performance metrics in the vicinity of the Pareto set are returned from this mapping study. For the Technical Feasibility Model, this mapping is critical since knowledge of the design variable values for a new design that is feasible in the engineering domain is obtained from it.

Now that the technical information for this paper has been presented, this work is applied to a simple case study problem involving 2 design variables and 3 objective functions. Convergence behavior and mapping studies are carried out for this problem and the results are presented in the next section.

5.0 CASE STUDY

The methods presented in the previous sections are developed for application to large, complex, multiobjective optimization problems. As part of this work, a Technical Feasibility Model was developed in conjunction with the General Motors R & D Center to test the feasibility of potential new automotive

designs. This problem involved 5 objective functions, 11 design variables and 3 constraints. However, in order to present complete details of the mapping and convergence studies and to be able to visually represent the results, a simple 3 objective problem with 2 design variables and no constraints is selected as a case study for this paper. Though this problem is simplistic in nature, it exhibits the necessary properties of problems for which the technology in this paper has been developed. The multiobjective problem used is stated in Eq. (2) below.

Minimize:

$$\begin{aligned} f_1 &= x^2 + (y-1)^2 \\ f_2 &= x^2 + (y+1)^2 + 1 \\ f_3 &= (x-1)^2 + y^2 + 2 \end{aligned} \quad (2)$$

Subject To:

$$-2 \leq x, y \leq 2$$

The design space is divided into discrete regions; each sized 0.1 units for each design variable. The bounds for the objective function values are determined from the MOGA results. These bounds are shown in Table 1.

	f_1	f_2	f_3
Lower Bound	0.0041	1.0006	2.0000
Upper Bound	4.0243	5.1950	4.0794

Table 1. Upper and Lower Bounds on Objective Functions

The results of the convergence and mapping studies are discussed next. In Sections 5.1, the issue of convergence of the MOGA to the Pareto set is studied for this case study problem while in Section 5.2 the mapping of the 3-D objective function space to the 2-D design space is studied.

5.1 RESULTS: CONVERGENCE

The generation of the Pareto frontier was completed using 10000 unique design evaluations. This number was selected since a high fidelity surface for gap analysis and surface fitting was required. For this simple problem, this analysis could be completed without much computational expense. However, for more complex problems, this may be infeasible due to time and/or computational constraints. Using this frontier as our exhaustive sampling of the problem, 6829 non-dominated designs were found. Next, a discretization size of 0.2 was selected for all objectives. Using this discretization size, the 6829 unique non-dominated designs were placed into **252 unique hyperboxes** in the performance space. Figure 10 shows a plot of the performance space, where each dot represents the centroid of a filled hyperbox of the Pareto frontier.

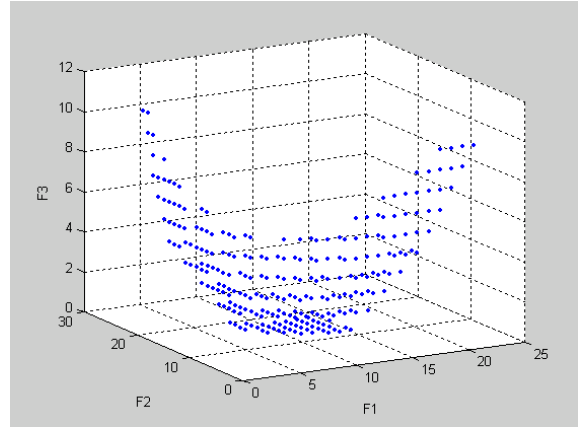


Figure 10. Index Representation of the Pareto Frontier

The 252 identified hyperboxes represent the target goal of any MOGA that is run on this problem. Obviously, as the discretization size changes, so will the number of hyperboxes filled by points from the Pareto frontier. The next step is to determine how well MOGA solutions developed using fewer evaluations can accurately capture the behavior of the frontier.

To start the convergence study, 150 evaluations are first completed to move the initial designs to the boundary of the Pareto surface. These points were evaluated by creating a random unique population of size 20, and maintaining a constant population across as many generations needed to reach 150 evaluations. For this case, 146 non-dominated designs are found and are placed into hyperboxes in the performance space. These designs map to 62 unique hyperboxes in the performance space. Comparing these 62 hyperboxes to the 252 that comprise the Pareto frontier, 52 of the hyperboxes are located on the frontier. Plotting the indices of these hyperboxes in the performance space, Figure 11 shows the representation of the true frontier after 150 evaluations.

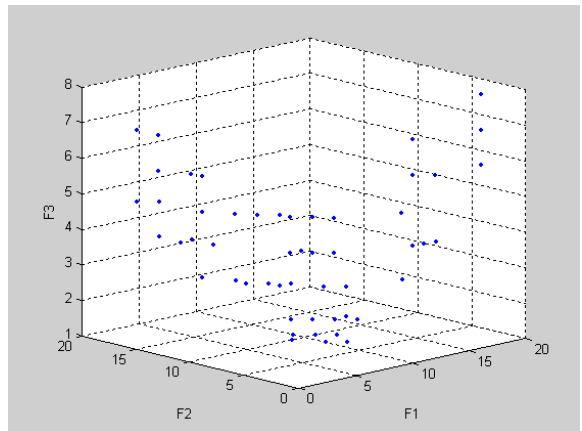


Figure 11. Index Representation of the Frontier after 150 Evaluations

Though the surface in Figure 11 is sparsely populated, the hyperboxes that have been identified are scattered across the entire representation of the frontier.

This data is now used as the starting point for the second phase of the convergence study. Using these designs as the initial population, new instances of the MOGA are created that have a

different number of total evaluations as stopping criteria. For this paper, 5 different cases are investigated. These MOGAs are designed to terminate after 500, 1000, 2000, 5000, and 7500 total evaluations. After the maximum number of allowed evaluations has been reached, the non-dominated points are placed into the appropriate hyperboxes. These hyperboxes are then compared to the exhaustive MOGA. Those hyperboxes that have the same index are recorded and are considered to represent the true Pareto frontier. The results of this analysis for the case study problem are shown in Figure 12.

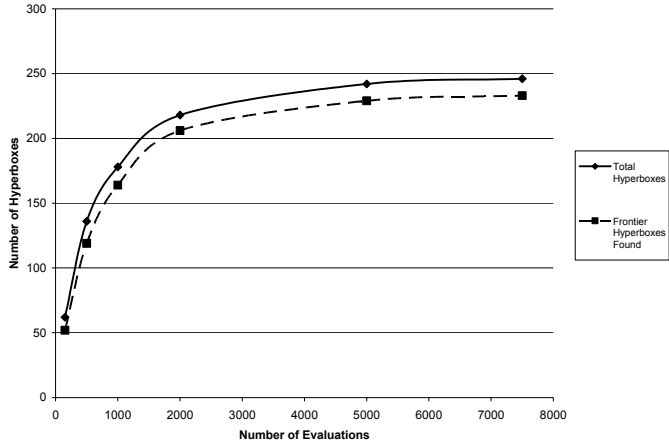


Figure 12. Number of Hyperboxes Filled as Evaluations Increase

The data in Figure 12 presents important information regarding the convergence of the MOGA. Completing only the first 150 evaluations – and creating the initial population used for the result of the studies – nearly 21 percent of the 252 hyperboxes comprising the Pareto frontier has been captured. At 500 total evaluations, nearly half of the hyperboxes that compose the Pareto frontier are filled by at least one design. Nearly 82 percent of the frontier has been captured by 2000 evaluations, and increasing the number of evaluations by another 250 percent only yields an extra 9 percent of the entire surface. By 7500 evaluations, roughly 93 percent of the Pareto frontier has been identified. Figure 13 demonstrates how the entire frontier of this problem is gradually accounted for, and begins to level off, as the number of evaluations increases.

These results show that a large number of evaluations can potentially be saved while still capturing the behavior of the frontier. However, this problem only contains three objective functions comprised of two design variables, and no constraints. A more challenging problem may not (for example the GM problem on which this was initially tested) capture such a large portion of the Pareto frontier in a small number of evaluations. Added objectives, design variables, and constraints, make developing the solution computationally more expensive. Therefore, it may be necessary to modify the definition by which an exhaustive MOGA hyperbox can be considered to be captured.

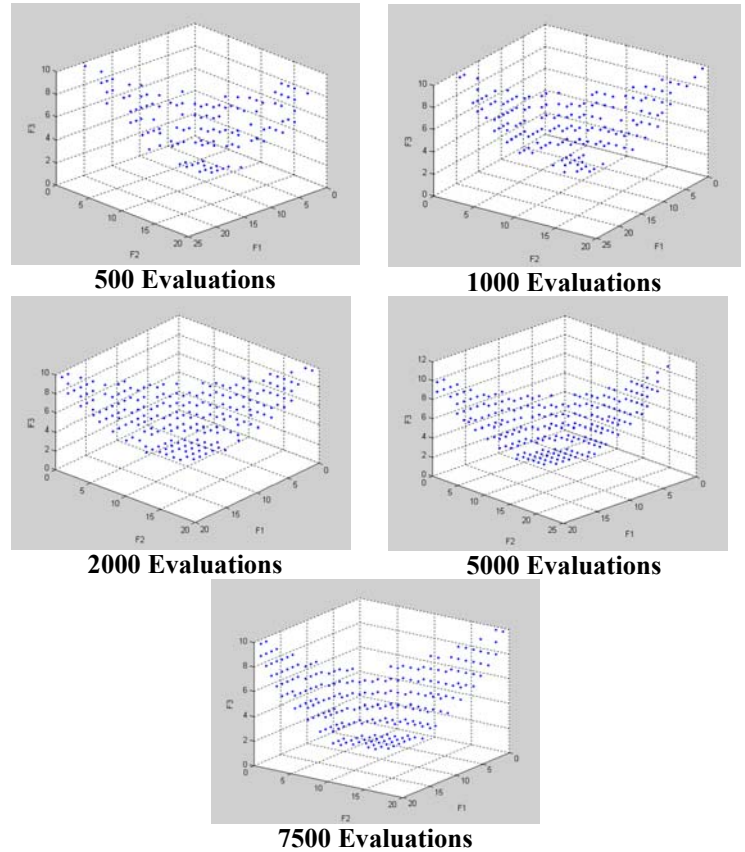


Figure 13. Index Representation of the Frontier at Different Evaluation Limits

One of the most powerful aspects that these results do not take into consideration is the hyperboxes that are adjacent to the exhaustive MOGA hyperboxes. Using different adjacency constraints, it becomes possible to provide more information about how well the different numbers of evaluation cases capture the behavior of the exhaustive MOGA frontier.

As the performance space for this system is three-dimensional, adjacency can occur in more than the standard two-dimensional space. Two objects can be considered adjacent so long as the absolute difference in any given dimension is no greater than one. Using this rule, Figure 14 shows the first four levels of adjacency possible in an n-dimensional space. The first level of adjacency is when all indices of the two hyperboxes being compared are the same in *all* dimensions. When the two hyperboxes compared are the same in *all but 1 dimension*, the two hyperboxes that are adjacent share a common plane, or face. When the indices comparison holds for *all but 2 dimensions*, the two compared hyperboxes share a common edge. Finally, when the indices of the hyperboxes are the same in *all but 3 dimensions*, the two compared hyperboxes share only a common point. This concept can be expanded into n-dimensional space, as a rule of thumb for considering how much flexibility in adjacency an engineer is willing to allow.

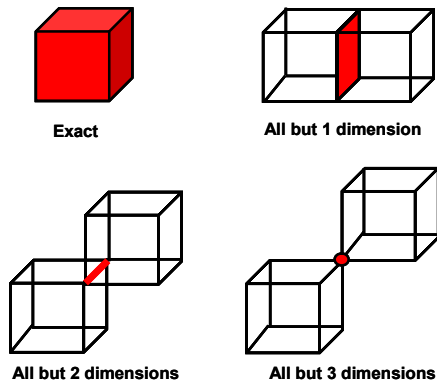


Figure 14. Levels of Adjacency

Applying the different adjacency constraints to the analysis allows for an exhaustive MOGA performance space hyperbox to be considered captured as long as a hyperbox from a MOGA test case is adjacent. As the performance space for the problem investigated here is only three-dimensional, the first four levels of adjacency are the only applicable cases. The data for this analysis is presented in Figure 15.

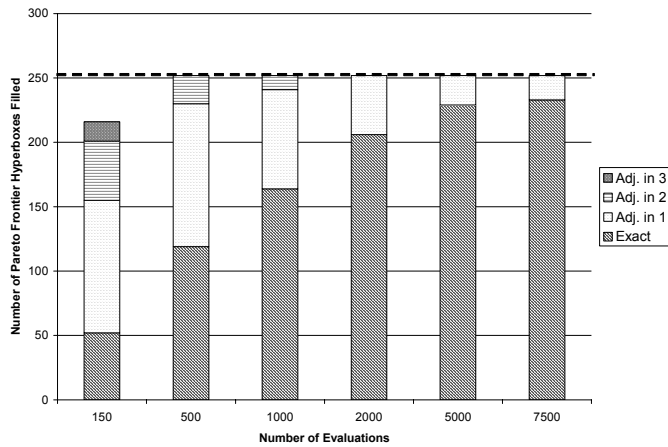


Figure 15. Convergence Data

From Figure 15, it can be seen that for the larger evaluation cases, implementing any level of adjacency corresponds to all 252 hyperboxes of the Pareto frontier being captured. For the 500 and 1000 evaluation cases, allowing for one level of adjacency results in a 200 and 150 percent increase respectively, in captured frontier hyperboxes. Allowing for two levels of adjacency, both the 500 and 1000 evaluation cases fully capture the behavior of the Pareto frontier. Allowing any level of adjacency significantly increases even the 150 evaluation case, demonstrating the degree to which the solutions for that MOGA run were scattered over the frontier.

Viewing the performance space as a series of hyperboxes allows for the possible reduction in total evaluations needed by a MOGA to effectively capture the behavior of the Pareto frontier. While a hyperbox may be filled when using a smaller number of evaluations, the number of designs located in each hyperbox will be less than seen in the exhaustive MOGA. However, such an advantage plays a significant role in a larger,

more complex problem. To further decrease the needed number of evaluations, incorporating different levels of adjacency provides the ability to capture a greater percentage of the frontier. As demonstrated in this problem, 2000 evaluations with an analysis of one level of adjacency are as effective in capturing the behavior of the Pareto frontier as the exhaustive MOGA case of 10000 evaluations. This is a significant 25 percent reduction in needed evaluations, which may prove invaluable as computational time and expense for system analyses increases.

The advantage of convergence can clearly be seen for a higher dimensional problem, as shown in Figure 16. This is the problem posed by General Motors, consisting of 5 objective functions, 10 design variables, and 3 constraints. The exhaustive MOGA for this study used 80,000 design evaluations, an obvious increase from the 10,000 evaluations needed to capture the frontier of the example problem. Here, the 20,000 evaluation case could exactly capture only 20% of the hyperboxes of the Pareto frontier. Allowing for the first level of adjacency, the different cases captures at least double the original amount of frontier hyperboxes. Increasing the levels of adjacency allows a significant number of hyperboxes to be captured, reducing the need for an exhaustive number of evaluations and returning a suitable frontier representation.

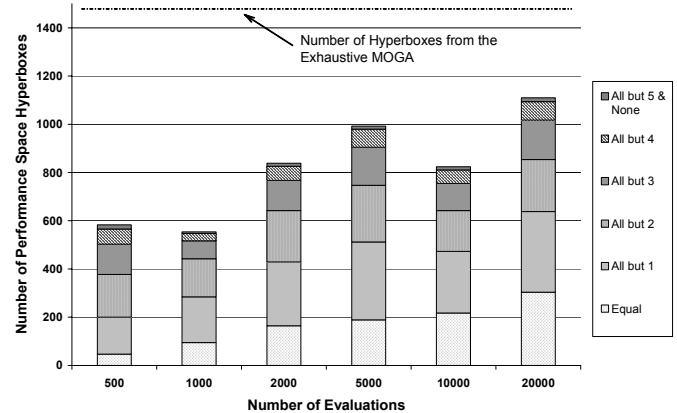


Figure 16. Convergence Data for the GM Problem

The study of convergence in this section was aided and completed with the application of discretizing the performance space into a collection of hyperboxes. In the next section, the issue of mapping from the performance space to the design space is addressed with the help of discretizations within the design space.

5.2 RESULTS: MAPPING

For the mapping study, the 3-D performance space is also discretized into equal sized hyperboxes, each side having a length of 0.2 units. Using this discretization size and the bounds listed in Table 1, a total of 4851 performance space hyperboxes exist of which 252 are populated with Pareto points as seen in the convergence study. Some important mapping measures determined in this study are presented in Table 1.

Number of Performance Space Hyperboxes containing one design mapping to ONE design variable hyperbox (contains only one design point)	21
Number of Performance Space Hyperboxes containing more than one design mapping to ONE design variable hyperbox	70
Number of Performance Space Hyperboxes containing more than one design mapping to MULTIPLE, adjacent design variable hyperboxes	143
Number of Performance Space Hyperboxes containing more than one design mapping to MULTIPLE, non-adjacent hyperboxes.	18
<i>Total Number of Performance Space Hyperboxes</i>	252

Table 1. Mapping Measures Determine from the Mapping Study

The results presented in Table 1 indicate the different mapping types discussed earlier. The first two entries in Table 1 correspond to *Type 1* mapping where one performance space hyperbox maps to one design variable hyperbox. The first entry corresponds to having one design point in the performance space hyperbox while the second entry corresponds to multiple designs in one performance space hyperbox. The third entry in Table 1 is the number of performance space hyperboxes corresponding to *Type 2* mapping, while the number of performance space hyperboxes exhibiting *Type 3* mapping is given in the fourth entry. The total number of populated hyperboxes is listed in the last entry. In addition to the numbers presented in Table 1, the results of this study also provides a map (a data spreadsheet) containing indices of the 252 performance space hyperboxes with the corresponding design variable values and respective tolerances. The procedure to determine the design variable values for a given set of performance measures from the determined map has been described earlier and is applied here to an example.

Consider a hypothetical design with performance values given as shown in Eq. (3).

$$(f_1, f_2, f_3) = (0.8, 2.75, 2.33) \quad (3)$$

It is desired to determine the design variable values and corresponding tolerances that would result in the above performance measures. It is important to note that for the given case study problem, the design variable values can be easily backed out from the analytical expressions given in Eq. (2). However, the studies presented in this paper are applicable for problems that are inherently large in terms of number of objective functions, design variables and system constraints and also include computationally intensive analyses that need to be performed. For this example problem, design variable information for the given set of performance values is obtained from the mapping as well as computed analytically from the problem definition and the two results are compared. The steps listed at the end of Section 4 are used first to determine the

design variable information from the results of the mapping study.

- Use bounds and discretizations to determine indices of Performance space hyperbox: Using the bounds of Table 1 and discretization size of 0.2, the indices for the given set of performance values are computed and result as **(4, 9, 2)**.
- Compare indices to performance space indices: Comparing the indices (4, 9, 2) to the results of our study, it is identified that the given performance values lie in hyperbox number 71, out of the possible 252.
- Determine the design variable information: Reading off the design variable values from the data obtained from mapping, the design variable values are given in Eq. (4).

$$(x, y) = (0.45 \pm 0.05, 0.25 \pm 0.05) \quad (4)$$

Thus, without going back to the analytical functions that form the system analyses, design variable information is obtained for a new design with desired set of performance values. Computing the design variable values analytically, the result obtained is shown in Eq. (5).

$$(x, y) = (0.4675, 0.2375) \quad (5)$$

As seen from Eq. (5), the design variable values obtained analytically lie in the tolerance range specified by the mapping results. For computationally intensive problems, determining design variable values without having to run the system analyses again is extremely useful. The mapping of the performance to design space for the given point is shown in Figure 17.

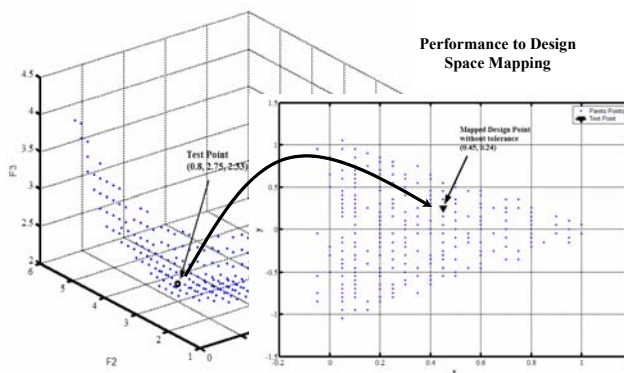


Figure 17. Performance to Design Space Mapping for Test Point

In Figure 17, the graph on the left is the 3-D Performance space with the *centroid* of populated Performance space hyperboxes shown in blue while the test point is shown in black. The graph on the right is the 2-D design space again with the design variable values for the Pareto points in blue and the test point shown as a black inverted triangle. All information for Figure 17 is derived from the data obtained in the mapping study.

Thus, in conclusion, it is seen that within the given tolerance, the mapping study is able to provide useful design variable

information given a set of specifications on objective function values. In addition, this is done without having to go back to the analyses and back solving the objective functions to determine the design variable values. This tool is even more useful when the objective functions are not analytical functions but values obtained from a black-box like analyses system where mathematical functions are not available to solve for the design variable values. To aid in the preliminary vehicle design process, the TFM incorporates feasibility assessment, optimality, performance to design mapping, and convergence information into a single, automated tool. This has been very valuable in developing and studying the problem posed by General Motors. In the next section, some conclusions and areas of future work for this research are presented.

6.0 CONCLUSIONS AND FUTURE WORK

In this paper, studies have been presented to analyze the convergence behavior of a Multi-Objective Genetic Algorithm by reducing the number of function evaluations that can be performed. Though the process is dependent on the application, it can be concluded that a set of non dominated solutions can be obtained in place of the true Pareto set using a smaller number of function evaluations. This is achieved by dividing the MOGA into two steps, where the first step has the MOGA using a part of the available function evaluations to cluster around one region of the Pareto set and the second step uses the remaining available function evaluations to populate the remaining region of the Pareto surface.

This paper also includes a study of the mapping between the performance and design space. It is shown that for a new design that is close to the Pareto solution set, design variable values and corresponding tolerances can be determined without computing the analyses again. This is done by dividing the two spaces into discrete regions of indifference called hyperboxes and studying the relationship of the hyperboxes in the performance and design spaces.

Sources of future work for this research include investigating methods for convergence of the MOGA without explicitly running the algorithm exhaustively and having a representation of the “true” Pareto set. This would include the development and use of metrics that assess the goodness of a set of non dominated designs obtained from a smaller number of function evaluations. Additionally, the performance to design mapping would need to be expanded to include regions away from the Pareto set since Pareto optimal design information might not be available. This would require meta-modeling techniques that generate an approximation of the system analyses and can be used to obtain more design variable data.

ACKNOWLEDGMENTS

We would like to thank the General Motors Corporation and the National Science Foundation Grant DMII-9875706 for their support of this research..

REFERENCES

[1] Lewis, K., Ferguson, S., Gurnani, A., and Donndelinger, J., 2005, “An Approach to Feasibility Assessment in

- Preliminary Design”, *Submitted to the Proceedings of IDETC*, Long Beach California
- [2] Eschenauer, C. M., Koski, J., and Osyczka, A., 1990, *Multicriteria Design Optimization*, Springer-Verlag, New York.
- [3] Pareto, V., 1906, *Manuale di Economica Polittica*, Società Editrice Libràia, Milan, Italy; translated into English by A. S. Schwier, as *Manual of Political Economy*, Macmillan, New York, 1971.
- [4] Eddy, J. and Lewis, K., 2001, "Effective Generation of Pareto Sets Using Genetic Programming," ASME Design Technical Conferences, Design Automation Conference, DETC2001/DAC-21093.
- [5] Fenyes, P. A., Donndelinger, J. A., and Bourassa, J.-F., 2002, “A New System For Multidisciplinary Design and Optimization of Vehicle Architectures”, 9thAIAA Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, AIAA-2002-5509.
- [6] Gu, X., and Fenyes, P. A., 2004, “Application of the Integrated Framework for Architecture Development (IFAD) for Conceptual Vehicle Design”, 10thAIAA Symposium on Multidisciplinary Analysis and Optimization, Albany NY, AIAA-2004-4315.
- [7] Wu, J., and Azarm, S., “Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set,” *Journal of Mechanical Design*, **123**(1), pp. 18-25.
- [8] Zitzler, E., and Thiele. L., 1998, “Multiobjective Optimization using Evolutionary Algorithms – A Comparative Case Study,” In Eiben, A. E., et al., *Proceedings of the 5th International Conference: Parallel Problem Solving from Nature – PPSNV*, Amsterdam, The Netherlands, Springer, pp. 292-302.
- [9] Deb, K., and Jain, S., “Running Performance Metrics for Evolutionary Multiobjective Optimization,” *KanGAL Report No. 2002004*, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India, 2002.
- [10] Van Veldhuizen, D. A., 1999, “Multiobjective Evolutionary Algorithm: Classifications, Analyses, and New Innovations,” Ph. D. Dissertation, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.
- [11] Kasprzak, E. and Lewis, K., 2001, "Pareto Analysis in Multiobjective Optimization Using the Colinearity Theorem and Scaling Method," *Structural and Multidisciplinary Optimization*, **22**(3), pp. 208-218.
- [12] Eddy, J. and Lewis, K., 2002, "Visulization of Multi-Dimensional Design and Optimization Data Using Cloud Visualization," ASME Design Technical Conferences, Design Automation Conference, DETC02/DAC-34130.
- [13] Lee, Yuejun E., and Black, Derrick R., 2004, "Improved RFQ Process by Optimization Technique", SAE Motorsports Engineering Conference & Exposition, Dearborn, MI, 2004-01-1155.
- [14] Ferguson, S., and Lewis, K., 2004, “Effective Development of Flexible Systems in Multidisciplinary Optimization,” 10thAIAA Symposium on Multidisciplinary Analysis and Optimization, Albany NY, AIAA-2004-4309.