

---

## **A study of convergence and mapping in preliminary vehicle design**

---

**Scott Ferguson and Ashwin Gurnani**

Department of Mechanical and Aerospace Engineering,  
University at Buffalo, SUNY,  
Buffalo, NY USA  
E-mail: smf7@eng.buffalo.edu      E-mail: agurnani@eng.buffalo.edu

**Joseph Donndelinger**

Vehicle Development Research Lab,  
General Motors Research and Development Center,  
Warren, MI USA  
E-mail: joe.donndelinger@gm.com

**Kemper Lewis\***

Department of Mechanical and Aerospace Engineering,  
University at Buffalo, SUNY,  
Buffalo, NY USA  
Fax: 716 645 2684      E-mail: kelewis@eng.buffalo.edu  
\*Corresponding author

**Abstract:** In this paper, we investigate the issue of convergence in multi-objective optimisation problems developed for vehicle analyses when using a Multi-Objective Genetic Algorithm (MOGA) to determine the set of Pareto optimal automobile configurations. Additionally, given a Pareto set for a multi-objective problem, the mapping between the performance and design space is studied to determine new automobile design configurations for a given set of performance specifications. The advantage of this study is that the automobile's design information is obtained without having to repeat system analyses. The tools developed in this paper are applied both to a simple multi-objective optimisation problem to illustrate the methodology and to a preliminary vehicle design framework to develop a Technical Feasibility Model (TFM) for use in the early stages of automobile design.

**Keywords:** multiobjective optimisation, performance space, design space, algorithm convergence, vehicle feasibility.

**Reference** to this paper should be made as follows: Ferguson, S., Gurnani, A., Donndelinger, J. and Lewis, K. (2005) 'A study of convergence and mapping in preliminary vehicle design', *Int. J. Vehicle Systems Modelling and Testing*, Vol. 1, Nos. 1/2/3, pp.192–215.

**Biographical notes:** Scott Ferguson received his BS (2002) and MS (2004) in mechanical engineering from the University at Buffalo. He is currently working toward his PhD and his dissertation focuses on the design and application of reconfigurable systems.

Ashwin Gurnani received his BS (2002) and MS (2003) in mechanical engineering from the University at Buffalo. He is currently working toward his PhD and his dissertation focuses on decision-based design and the study of rationality in design decisions.

Joseph Donndelinger is a Staff Research Engineer in the Vehicle Development Research Laboratory at the General Motors Research and Development Center in Warren, Michigan. His ten years of experience with General Motors and the Ford Motor Company have broadly spanned the vehicle development process, with assignments in sub-system design, vehicle integration, advanced technology development, and process reengineering. His research interests include optimisation, decision-making, and methods for design conceptualisation, decomposition and synthesis. He received an MS degree in industrial engineering and a BS degree in mechanical engineering, both from the University of Illinois at Urbana-Champaign.

Kemper Lewis is an Associate Professor in Department of Mechanical and Aerospace Engineering at the University at Buffalo – SUNY, USA. He is also the Executive Director of the New York State Center for Engineering Design and Industrial Innovation (NYSCEDI). His research areas include design theory, decision-based design, distributed design, reconfigurable systems, multi-disciplinary optimisation, and multi-objective design. He has over 100 technical publications in these research areas. His awards include the Society of Automotive Engineers Teetor Award, the ASME Black and Decker Best Paper Award, the State University of New York Chancellor's Award for Excellence in Teaching, and the National Science Foundation Career Award.

---

## **1 Introduction**

The goal of creating a Technical Feasibility Model (TFM) is to provide for rapid assessments of design feasibility in the preliminary stages of the design process. Feasibility in this tool is assessed by investigating whether or not product specifications are mutually compatible from an engineering design perspective. This paper covers two issues critical to successful deployment of a TFM. The first issue is the convergence of the multi-objective optimisation 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 behaviour 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 because evaluation time could seriously inhibit the success of the preliminary design process. Mapping from the performance to design space is critical to understanding how changes in product specifications affect the design configuration. This paper presents both the TFM development procedures in Gurnani et al. (2005) and a discussion of the TFM development results. Section 2 provides background information into multi-objective optimisation, convergence of MOGAs, and the issues pertaining to the mapping of performance space to design space.

## 2 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 optimisation 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. Designers also seek to provide ‘surprise and delight’ attributes to get an 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 optimisation, which provides essential tools in achieving many goals simultaneously as discussed in Eschenauer et al. (1990), is an important area of research and the primary focus of this paper.

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. The theory behind Pareto optimality is introduced in Pareto (1906). 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 equation (1)

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

in which  $n$  is the number of objectives and the use of the ‘less than’ symbol indicates an improvement in a criteria (minimisation of objectives).

With the increase in availability of computational resources, heuristic optimisation methods, such as genetic algorithms that are 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 MOGA, 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 as shown in Eddy and Lewis (2001). 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. Details of this framework have been published in Fenyes et al. (2002), as well as in Gu and Fenyes (2004). 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. Wu and Azarm (2001) have 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 optimisation methods. Deb and Jain (2002) have proposed a metric that evaluates the convergence of a solution set to a reference set, while van Veldhuizen (1999) proposes an error ratio to determine if a solution set has converged to the true solution set. In this paper, a study that compares non-dominated solution sets, obtained by using a smaller number of function evaluations, to the true Pareto set, is presented. This is critical because each function evaluation is computationally expensive.

In addition to reducing the number of function evaluations to obtain the Pareto set, vehicle development teams also require knowledge of the relationships between 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 maximise 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 recognised as a challenging task because the mapping can be one-to-many, with one objective function point mapping back to multiple design points as shown by Kasprzak and Lewis (2001). Past work includes the use of a visualisation technique called *Cloud Visualisation* to determine design variable values for a given point in the performance space. This work has been presented by Eddy and Lewis (2002). The use of design variable mapping has also been shown by Lee and Black (2004) to accelerate the design process for a multi-piece propshaft. Additionally, Ferguson and Lewis (2004) discuss the criticality of mapping between performance and design spaces in morphing systems where changing from one optimal configuration to another can potentially create drastic changes in the design configuration. 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 MOGA convergence

The first step of constructing a Technical Feasibility Model relies upon the usage of a MOGA to solve a multi-objective optimisation 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 used to assess technical feasibility as well as the optimality of a given test point. To ensure that the entire frontier is populated, an exhaustive number of evaluations are used. For the purposes of this paper, the MOGA process used to create the Pareto frontier is referred to as the ‘exhaustive’ MOGA. The solution of this exhaustive MOGA serves as the Pareto frontier benchmark while comparing different Pareto frontier solutions. However, such a large number of evaluations for a multi-objective 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 trade-off between the number of designs evaluated and the quality of the Pareto frontier. In order to determine this trade-off, a convergence study is presented in Section 3.1. Section 3.2 then applies the convergence study to a test problem and presents the results for that problem.

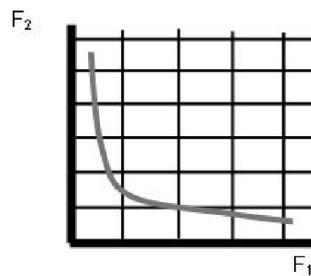
#### 3.1 Convergence of a multi-objective problem Pareto set

Understanding this trade-off 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:

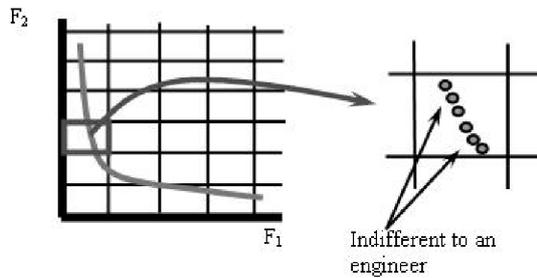
- *Complete an exhaustive sampling of the model.* As mentioned earlier, this results in a set of objective function values that are assumed to be the true Pareto set.
- *Specify indifference thresholds.* These are intervals for each objective within which the designer is indifferent to all objective values. Each interval is based on the designer’s preferences over a design objective and is a function of his or her experience and knowledge of the problem. It is assumed that the designer has sufficient information to state preferences over the objectives and to specify the indifference thresholds. As an example, in vehicle design, a design engineer may decide that the difference between 30.0 mpg and 30.2 mpg is not critical and, therefore, these two ratings could be viewed as being equivalent.
- *Discretise the performance space using the defined thresholds.* Using the indifference thresholds to establish the discretisation sizes for each objective, the performance space is divided into a collection of ‘hyperboxes’ (for problems with more than three objectives). For a problem with three objectives, the performance space would be discretised into a set of equally sized rectangular cuboids, where the cuboid dimensions are the indifference threshold values for each objective.

- *Represent the exhaustive MOGA as a collection of hyperboxes in the performance space.* Using the discretised performance space, it is possible to visualise the resultant view of the Pareto frontier as seen in Figure 1. 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. For the problem shown in Figure 1, the hyperboxes populated by the Pareto points (determined using the MOGA) are shown in Figure 3.
- *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 are developed to analyse the trade-off of maximum allowed evaluations to the quality of the Pareto frontier. The number of function evaluations available is treated as a constraint in the set-up of the MOGA.

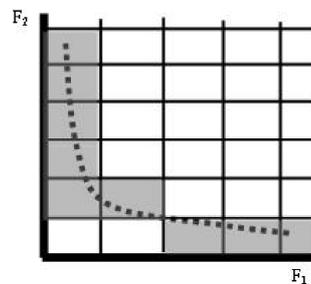
**Figure 1** Representing the pareto frontier as a collection of hyperboxes



**Figure 2** Identification of indifferent designs within a performance space hyperbox

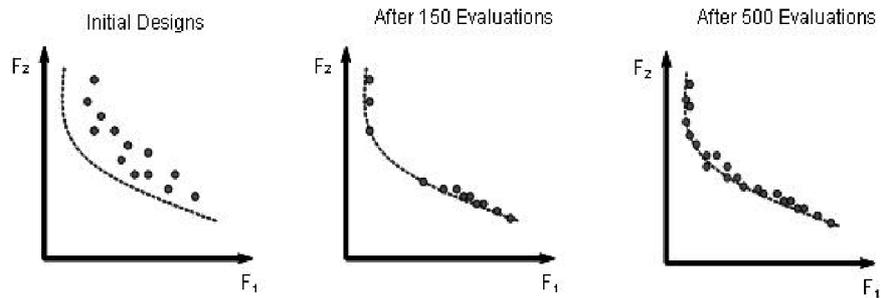


**Figure 3** Identification of hyperboxes filled in the performance space by the pareto frontier



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 (e.g., 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.

In order to present the complete details of the technical background and to be able to visually represent the results, a simple three objective problem, with two design variables and no constraints, is selected as a working example problem. Though this problem is simplistic in nature, it exhibits the necessary properties of problems for which the technology in this paper has been developed. A convergence study of this problem is completed in the next section.

### 3.2 Convergence study for working example problem

The multi-objective problem used in this working example is stated in equation (2) below.

Minimise :

$$f_1 = x^2 + (y-1)^2$$

$$f_2 = x^2 + (y+1)^2$$

$$f_3 = (x-1)^2 + y^2 + 2. \quad (2)$$

Subject to :

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

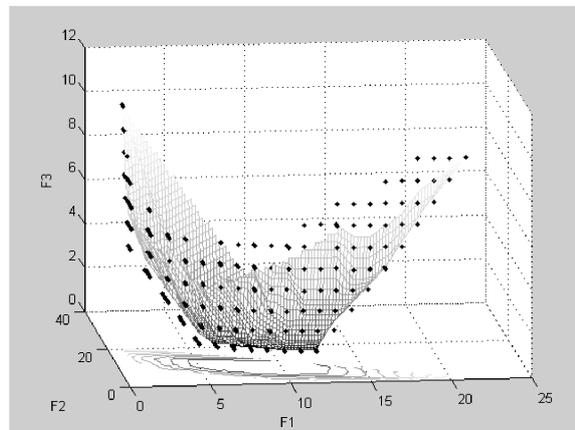
The design space, defined by the design variables, is divided into equally spaced squares. The side of each square has a length of 0.1 units. The bounds for the objective function values are determined from the MOGA results. These bounds are shown in Table 1.

**Table 1** Upper and lower bounds on objective functions for the working example problem

	$F_1$	$F_2$	$F_3$
Lower bound	0.0041	1.0006	2.0000
Upper bound	4.0243	5.1950	4.0794

The generation of the Pareto frontier is completed using 10000 unique design evaluations. This number is selected since a high fidelity surface for gap analysis and surface fitting is 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, 6,829 non-dominated designs are found. Next, a discretisation size of 0.2 units is selected for all objectives. Using this discretisation size, the 6,829 unique non-dominated designs are placed into 252 *unique hyperboxes* in the performance space. Figure 5 shows a plot of the performance space, where each dot represents the centroid of a filled hyperbox of the Pareto frontier.

**Figure 5** Index representation of the pareto frontier



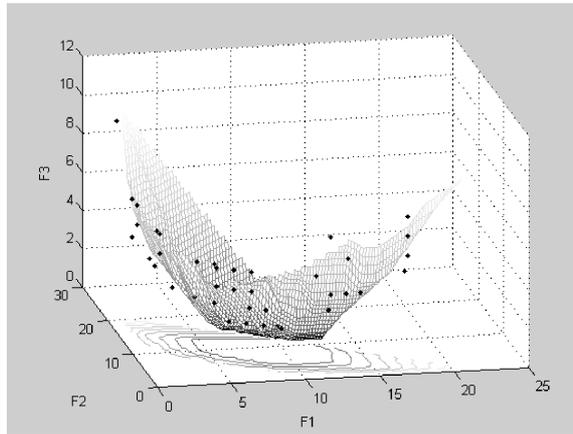
The 252 identified hyperboxes represent the target goal of any MOGA that is run on this problem. Obviously, as the discretisation size changes, so will the number of hyperboxes filled by points from the Pareto frontier. The next step is to determine how well MOGA solutions developed using fewer evaluations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

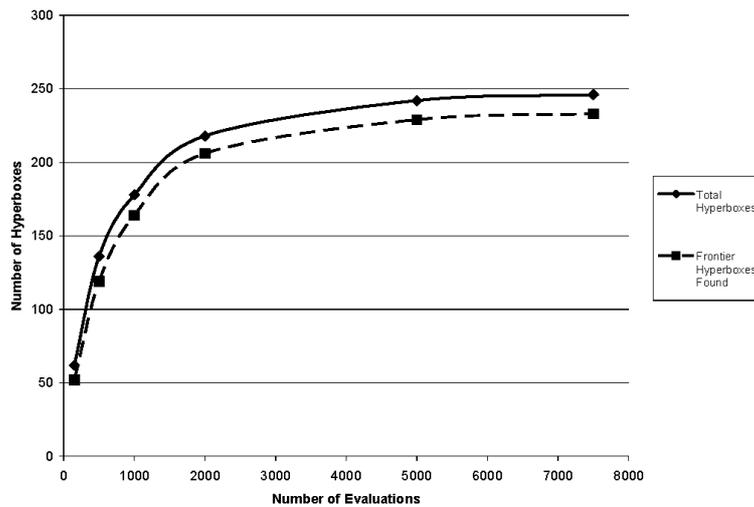
these 62 hyperboxes to the 252 that comprise the Pareto frontier, 52 of the hyperboxes are located on the frontier. Plotting the indices of these hyperboxes in the performance space, Figure 6 shows the representation of the frontier after 150 evaluations.

Though the surface in Figure 6 is sparsely populated, the hyperboxes that have been identified are scattered across the entire span 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 study, five 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 7.

**Figure 6** Index representation of the frontier after 150 evaluations

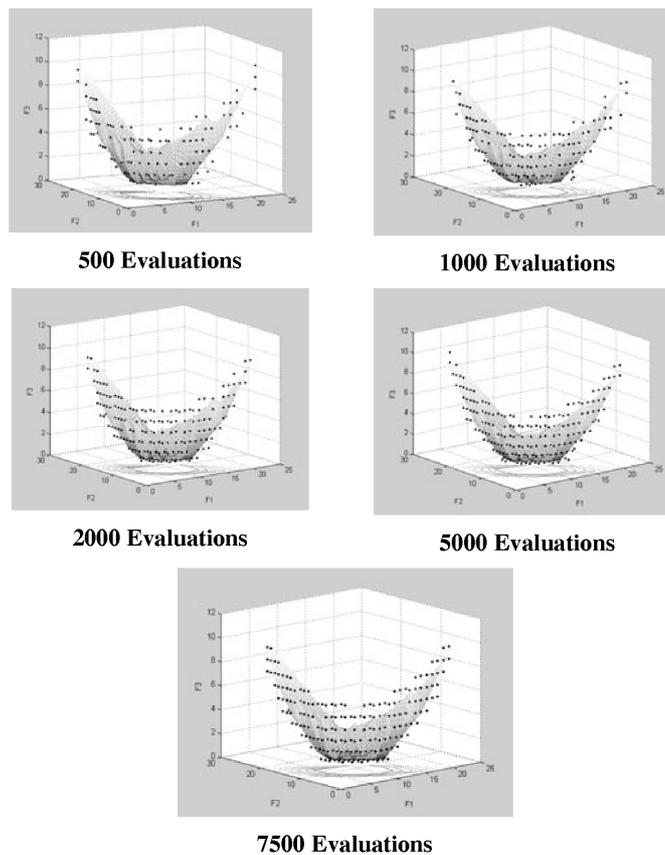


**Figure 7** Number of hyperboxes filled as evaluations increase



The data in Figure 7 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% 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% of the frontier has been captured by 2,000 evaluations, and increasing the number of evaluations by another 250% only yields an extra 9% of the entire surface. By 7,500 evaluations, roughly 93% of the Pareto frontier has been identified. Figure 8 demonstrates how the entire frontier of this problem is gradually accounted for as the number of evaluations increases.

**Figure 8** Index representation of the frontier at different evaluation limits

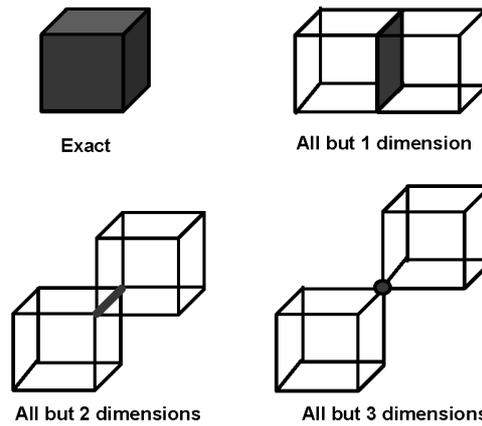


These results show that a large number of evaluations can potentially be saved while still capturing the behaviour of the frontier. However, this problem only contains three objective functions comprised of two design variables, and no constraints. For a more challenging problem, such a large portion of the Pareto frontier might not be captured in a small number of evaluations. Additional 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.

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 behaviour 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 9 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.

**Figure 9** Levels of adjacency



This concept can be expanded into  $n$ -dimensional space and used as a guideline for considering how much flexibility in adjacency an engineer is willing to allow. Note that each of the uniform hyperboxes in the discretised performance space (from the indifference thresholds) has been assigned an index. Adjacency is specified in terms of these indices. For example, for a five objective problem a hyperbox with index set  $(0,1,1,1,1)$  is adjacent to a hyperbox with index set  $(0,1,1,1,2)$  in all but one dimension. The same hyperbox  $(0,1,1,1,1)$  is adjacent to a hyperbox with index set  $(1,1,1,1,2)$  in all but two dimensions and so on. Generalising this idea, consider the performance space of an  $n$ - objective problem with two hyperboxes having indices  $(x_1, x_2, \dots, x_n)$  and  $(y_1, y_2, \dots, y_n)$ . The number of dimensions that the two hyperboxes are adjacent in is given as  $(n - L)$  where  $L$  is defined in equation (3)

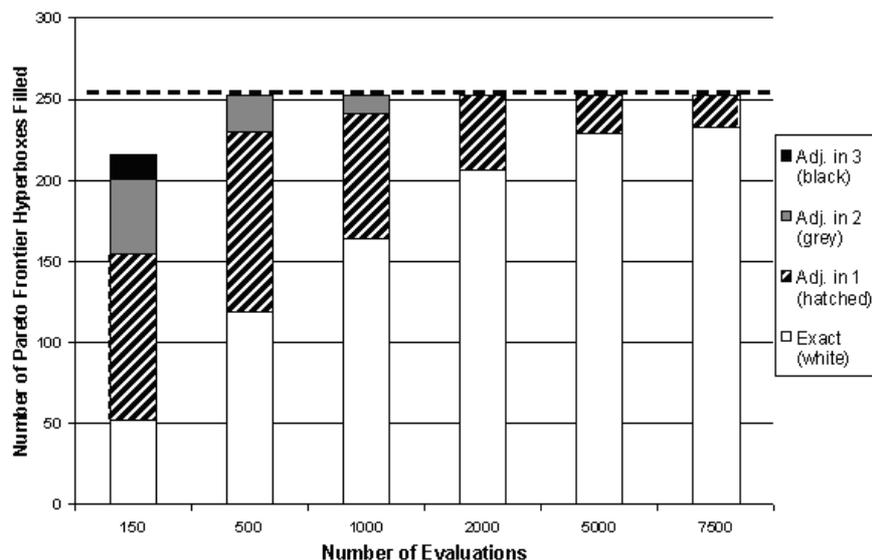
$$L = \sum_{i=1}^n |x_i - y_i| \quad (3)$$

iff  $|x_i - y_i| \leq 1$  for  $i = 1, n$ .

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 10.

From Figure 10, 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 1,000 evaluation cases, allowing for one level of adjacency results in a 200% and 150% increase, respectively, in captured frontier hyperboxes. Allowing for two levels of adjacency, both the 500 and 1,000 evaluation cases fully capture the behaviour of the Pareto frontier. Allowing any level of adjacency significantly increases even the 150 evaluation cases, demonstrating the degree to which the solutions for that MOGA run are scattered over the frontier.

**Figure 10** Convergence data



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 behaviour 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, 2,000 evaluations with an analysis of one level of adjacency are nearly as effective in capturing the behaviour of the Pareto frontier as the exhaustive MOGA case of 10,000 evaluations. This type of large-scale reduction in the number of evaluations required to represent the Pareto frontier would likely prove invaluable as computational time and expense for system analyses increases.

The ability to determine the true Pareto frontier of a problem allows for the generation of a TFM. 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 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 with respect to the selected performance measures. 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. This may be accomplished by determining the 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. Section 4.1 discusses how design variable information can be obtained by mapping the performance space to the design space. The results of the mapping study as performed on the test problem are presented in Section 4.2.

##### *4.1 Performance to design space mapping*

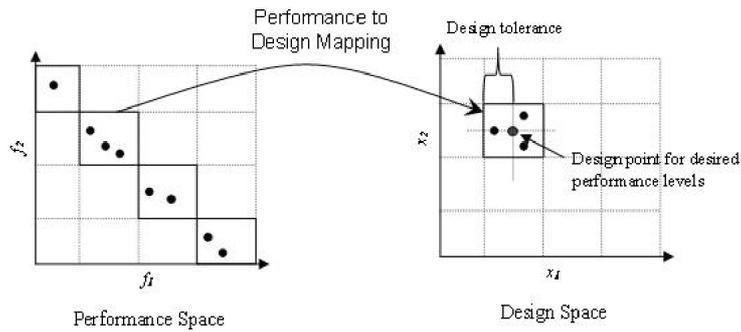
For the purpose of developing a map between the performance and design spaces, indifference thresholds that have been defined earlier are used for the performance space. Using the indifference thresholds to establish the discretisation sizes for each objective, the performance space is divided into a collection of hyperboxes, as discussed earlier, for the convergence study. Each performance space hyperbox maps to some region of the design space that is also discretised 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 and design space hyperboxes can be of three types as discussed below.

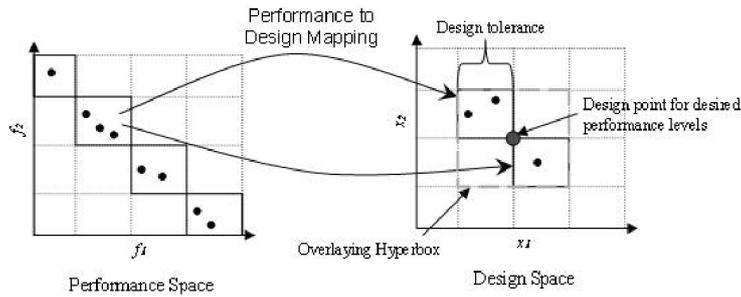
- *Type 1.* Individual performance space hyperbox maps to one design space hyperbox. In this case, the centroid of the design space hyperbox is the design variable vector and the tolerance is half the discretisation range. This is shown in Figure 11.
- *Type 2.* Performance space hyperbox maps to multiple, *adjacent* design space hyperboxes. In this case, a larger hyperbox can encompass all the mapped adjacent hyperboxes. The design variable values correspond to the centroid of this overlaying box with the tolerance values determined from the extremes of the enveloping hyperbox. This is shown in Figure 12.

- Type 3.* Performance space hyperbox maps to multiple, non-adjacent design space hyperboxes. As shown in Figure 13, one performance space hyperbox maps to different design space hyperboxes that are spaced out in the design space. It is hypothesised 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.

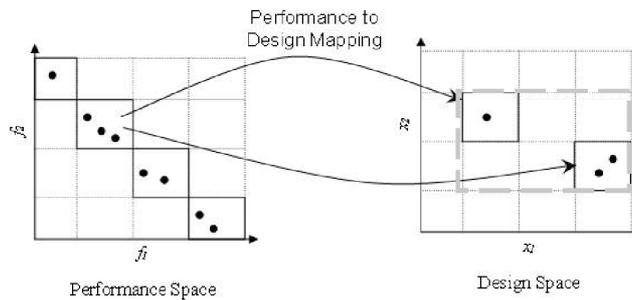
**Figure 11** One performance space hyperbox mapped to one design space hyperbox



**Figure 12** One performance space hyperbox mapped to multiple adjacent design space hyperboxes



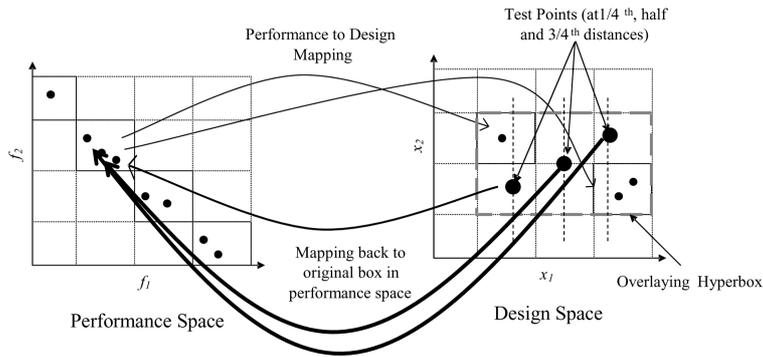
**Figure 13** One performance space hyperbox mapped to multiple, non-adjacent design space hyperboxes



For a set of non-adjacent, mapped design space hyperboxes, a larger hyperbox is defined 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 are originally mapped, the hypothesis is considered valid. The three points

selected are the centre 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 14.

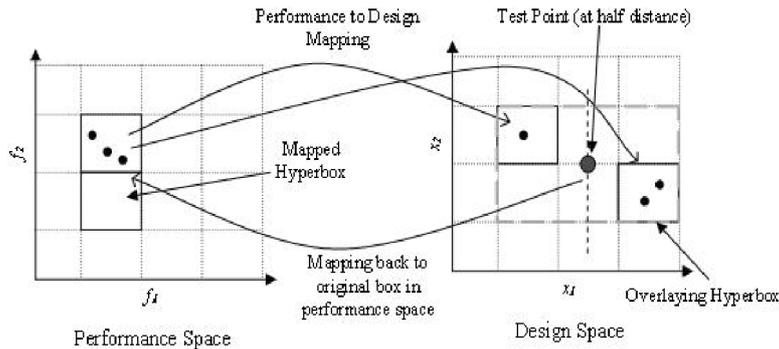
**Figure 14** Determining performance space hyperbox for intermediate test points in overlaying design space hyperbox



These three points generated in Figure 14 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 15.

**Figure 15** Test point mapping to performance space hyperbox adjacent to original



Thus, for *Type 3* mapping, 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 centre 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 discretisation 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. Because the given performance-measure values are known to maximise 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 mapping information for this paper has been presented, this work is applied to the case study problem involving two design variables and three objective functions introduced in Section 3.2. Mapping studies are carried out for this problem and the results are presented in Section 4.2. Detailed results of this study for the vehicle design problem are presented in Section 5.

#### 4.2 Mapping results for working example problem

The process of mapping the performance space to the design space is illustrated using the example problem represented by equation (2). For the mapping study, the 3-D performance space is also discretised into equal sized hyperboxes, each side having a length of 0.2 units. Using this discretisation size and the bounds listed in Table 1, a total of 4,851 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 2.

**Table 2** Measures determined from mapping study for 3-objective problem

<i>Mapping type</i>	<i>Mapping description</i>	<i>No. of pts</i>
1	Number of performance space hyperboxes containing one design mapping to ONE design variable hyperbox (contains only one design point)	21
1	Number of performance space hyperboxes containing more than one design mapping to ONE design variable hyperbox	70
2	Number of performance space hyperboxes containing more than one design mapping to MULTIPLE, adjacent design variable hyperboxes	143
3	Number of performance space hyperboxes containing more than one design mapping to MULTIPLE, non-adjacent hyperboxes.	18
	Total number of performance space hyperboxes	252

The results presented in Table 2 indicate the different mapping types discussed earlier. The first two entries in Table 2 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 2 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 2, 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 equation (4).

$$(f_1, f_2, f_3) = (0.8, 2.75, 2.33). \quad (4)$$

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 found from the analytical expressions given in equation (2). However, the methods presented in this paper are generalisable to non-explicit 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 discretisations to determine indices of performance space hyperbox:* using the bounds of Table 1, and discretisation 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, the given performance values lie in hyperbox number 71
- *determine the design variable information:* reading off the design variable values from the data obtained from mapping, the design variable values are given in equation (5)

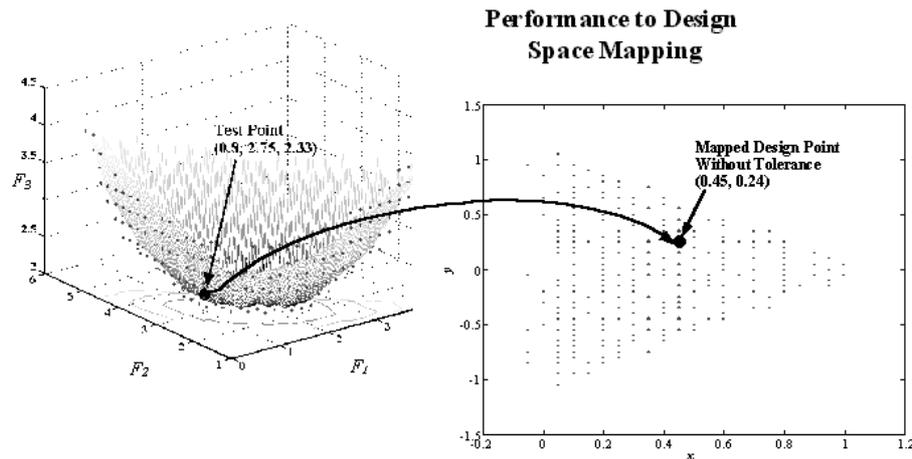
$$(x, y) = (0.45 \pm 0.05, 0.25 \pm 0.05). \quad (5)$$

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 equation (6).

$$(x, y) = (0.4675, 0.2375). \quad (6)$$

As seen from equation (6), 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 16.

**Figure 16** Performance to design mapping of test point



In Figure 16, the graph on the left is the 3-D performance space with the *centroid* of populated performance space hyperboxes shown along with the test point. The graph on the right is the 2-D design space again with the design variable values for the Pareto points along with the mapped test point. All information for Figure 16 is derived from the data obtained in the mapping study.

Thus, it is seen that within the given tolerance, the mapping study provides 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 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.

In this section, the methodology behind performance to design mapping is discussed and, as an illustration, is applied to a simple three objective problem introduced in Section 3.2. The convergence and mapping studies are developed for and applied in preliminary vehicle design frameworks. The results of these studies for the vehicle design problem are presented in Section 5.

## 5 Case study results

The methods presented in the previous sections are developed for application to large, complex, multi-objective optimisation problems. As part of this work, these studies

are conducted during the development of a Technical Feasibility Model (TFM) to test the feasibility of preliminary vehicle designs. The vehicle design problem involves five objectives, 11 design variables, and three constraints. The design variables used in this problem are a combination of eleven high-level vehicle dimensions – such as the vehicle’s overall length, width, and height – and discrete design configuration choices including specification of the vehicle’s powertrain and tire size. The objectives from this analysis are five vehicle attribute measures taken from several engineering disciplines such as energy management and occupant packaging. The constraints are developed to ensure that the candidate vehicle designs generated by the MOGA are realistically proportioned vehicles.

As seen in the working example problem in previous sections, an exhaustive MOGA is needed for both the convergence study and for mapping from performance to design space. For this exhaustive case, the MOGA is run for 20 generations, resulting in over 80,000 unique design evaluations. Also, the final population of this MOGA consists of thousands of unique non-dominated designs. To protect the proprietary data used in this study, all information relating to the performance and design spaces has been normalised. First, it is necessary to discretise the performance and design spaces using defined indifference thresholds. For the purposes of this study, the discretisations seen in Tables 3 and 4 correspond to the performance and the design space respectively. This information is used in the next section to compare performance space hyperboxes in the convergence study.

**Table 3** Upper and lower bounds on objective functions for the vehicle design problem

<i>Objective</i>	<i>Discretisation size</i>	<i>Number of discretisations</i>
Acceleration – $F_1$	0.1	10
Fuel economy – $F_2$	0.058	17
Cargo volume – $F_3$	0.13	8
Front headroom – $F_4$	0.165	5
Shoulder room – $F_5$	0.136	7

**Table 4** Upper and lower bounds on design variables for the vehicle design problem

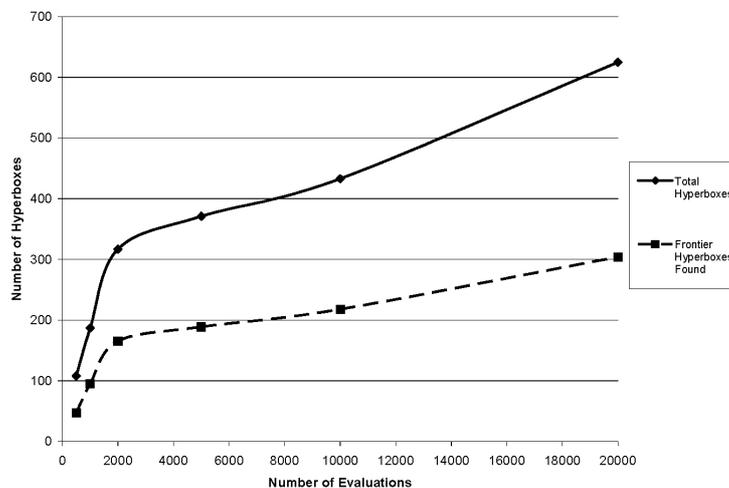
<i>Design variable</i>	<i>Discretisation size</i>	<i>Number of discretisations</i>
Overall vehicle dimension – $x_1$	0.145	8
Overall vehicle dimension – $x_2$	0.084	13
Wheel position – $x_3$	0.5	2
Wheel position – $x_4$	0.435	4
Occupant position – $x_5$	0.168	7
Occupant position – $x_6$	0.333	4
Occupant position – $x_7$	0.252	5
Occupant position – $x_8$	0.333	4
Overall vehicle dimension – $x_9$	0.210	6
Wheel position – $x_{10}$	0.167	7
Powertrain configuration – $x_{11}$	0.2	5

### 5.1 Convergence study for vehicle design problem

Using the discretisation sizes for the performance space in Table 3, the performance space is divided into a collection of hyperboxes. While it may seem that the discretisation sizes selected for the model are large, this combination divides the performance space into 4,76,000 total hyperboxes. Thus, there is an obvious trade-off between the discretisation size and the total number of hyperboxes that are created in the performance space. From the 80,000 total evaluations completed in the exhaustive sampling, 11,885 unique configurations are identified as non-dominated designs. These 11,885 designs are placed into 1,400 unique hyperboxes in the performance space.

To study the convergence of the algorithm six different MOGA cases are generated and compared to the exhaustive Pareto frontier. These cases are developed to analyse the trade-off of maximum allowed evaluations and the ability of the algorithm to accurately capture the Pareto frontier. The different cases allowed for 500, 1000, 2000, 5000, 10,000, and 20,000 evaluations to be completed. The goal of these studies is to evaluate the quality of the frontier generated relative to the exhaustive case, while limiting the number of evaluations to contain computational expense. After the maximum number of allowed evaluations has been reached, the non-dominated designs are placed into the appropriate hyperboxes and compared to the exhaustive MOGA. To start, the 500 evaluation cases found 360 non-dominated designs that mapped to 108 unique performance space hyperboxes. Comparing these 108 hyperboxes to those of the exhaustive MOGA, 47 hyperboxes are found to be identical. The results of the remaining cases are shown in Figure 17.

**Figure 17** Number of hyperboxes filled as evaluations increase for the vehicle design problem

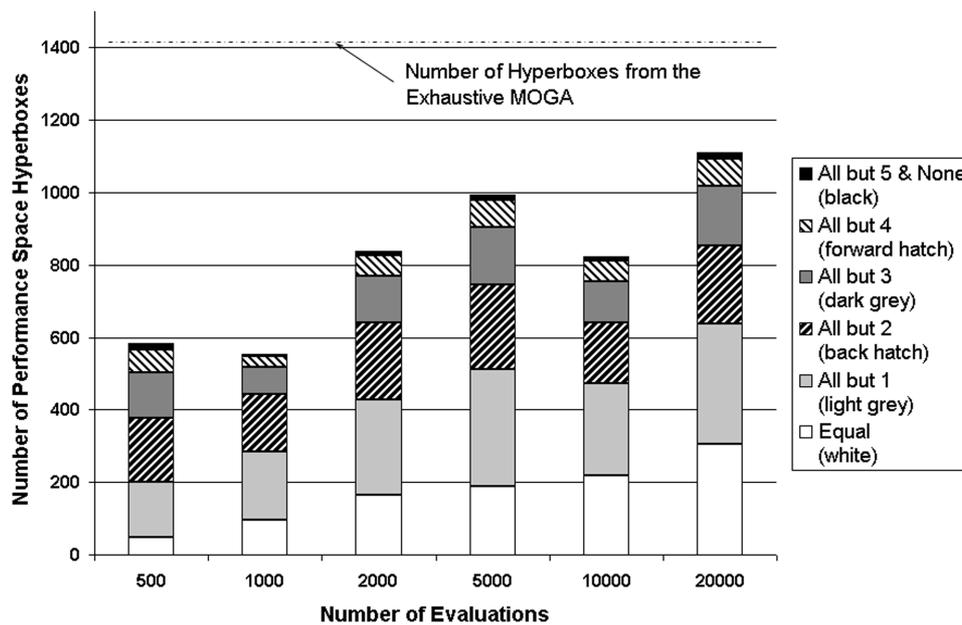


The data in Figure 17 demonstrates how capturing the frontier becomes increasingly difficult as the number of objectives studied increases. Completing 20,000 evaluations yields only 21% of the exhaustive MOGA frontier exactly. It can also be seen that the increase from 5,000 to 20,000 evaluations identifies an extra 100 hyperboxes, or an extra 7% of the total frontier. There is a critical point in the graph located at 1,000 evaluations where the relative change in identified frontier hyperboxes greatly decreases.

This indicates that a large number of evaluations would be needed to accurately capture the Pareto frontier identified by the exhaustive MOGA.

The advantage of adjacency can clearly be seen for a higher dimensional problem, as shown in Figure 18. When using exact comparisons, only 21% of the entire frontier could be identified. Allowing for the first level of adjacency, the different cases capture at least double the original number of frontier hyperboxes. Increasing the levels of adjacency allows for a significant number of hyperboxes to be captured, reducing the need for an exhaustive number of evaluations and returning a suitable frontier representation.

**Figure 18** Convergence data for the vehicle design problem



The study of convergence in this section is aided and completed with the application of discretising 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 using the discretisations within the design space.

### 5.2 Mapping results for vehicle design problem

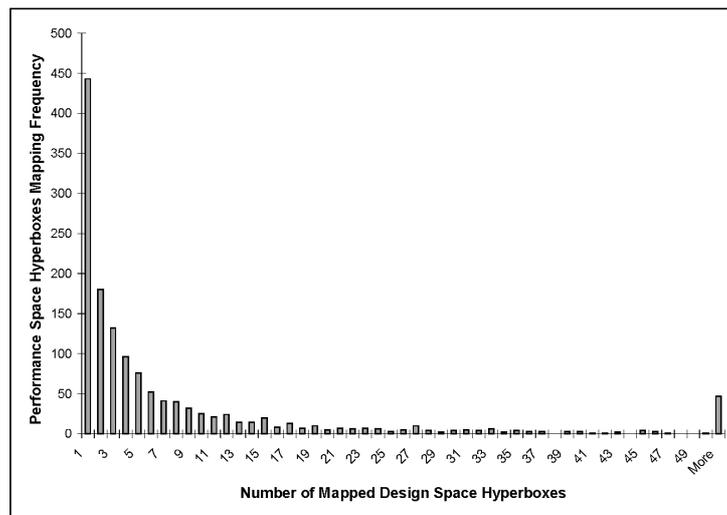
The objective of this study is to develop tools that enable a designer to determine design variable values given desired performance levels. The Pareto set of points is determined by running the MOGA for 20 generations. For this work, it is assumed that the MOGA converges to the Pareto frontier in 20 generations. The number of hyperboxes that are determined are assumed to span the entire Pareto frontier. Given the discretisation sizes for the objective functions in Table 3, the following results are obtained:

Number of Pareto points in original data (obtained from MOGA) = 11,885

Number of hyperboxes needed to span the entire Pareto surface = 1,400.

Using the discretisations of Table 4, the design variable values of each performance space hyperbox are converted to indices representing the design space hyperbox they belong to. Using the mapping information of performance space hyperboxes to design space hyperboxes, the number of design space hyperboxes mapped by each performance space hyperbox is determined. These results are provided in the histogram of Figure 19.

**Figure 19** Histogram of number of design space hyperboxes mapped by each performance space hyperbox



As seen in Figure 19, 443 out of 1,400 performance space hyperboxes are mapped to one design variable hyperbox. More detailed results of the mapping study are provided in Table 5.

**Table 5** Detailed mapping results for the vehicle design problem

Number of performance space boxes mapping to one design space box	443
Number of performance space hyperboxes mapping to one design space hyperbox with a single design point	438
Number of performance space hyperboxes mapping to multiple adjacent design space hyperboxes	40
Number of performance space hyperboxes mapping to multiple, non adjacent design space hyperboxes	917

From Table 5, it is shown that 443 out of 1,400 performance space hyperboxes fall under *Type 1* mapping, while 40 out of 1,400 performance space hyperboxes exhibit *Type 2* mapping. Additionally, as seen in Table 5, there exist 917 out of 1,400 performance space that have *Type 3* mapping. For these 917 performance space hyperboxes, an overlaying hyperbox is placed and 3 intermediate design points generated as described in Section 4. The results for the 917 *midpoints* generated are given below followed by a discussion of the results.

Number of designs that map to the original hyperbox = 172

Number of designs that map to a hyperbox adjacent to the original hyperbox = 690

Number of designs that are infeasible = 176.

From the above results, it is seen that 862 (690 + 172) of the 917 designs map to at least a hyperbox adjacent to the original hyperbox. Based on the earlier discussion, this validates placing an overlaying hyperbox over mapped design space hyperboxes that are not adjacent to each other since 94% of the points map back to the original or adjacent to original performance space hyperbox. The centre of the hyperbox determines the design variable values for the given performance inputs with distances to the extremes of the hyperbox forming the tolerances of the design range.

It is important to note that 176 of the 917 designs are evaluated as infeasible, which includes vehicle designs mapped to the original and adjacent performance hyperboxes. It is determined that these designs violated the upper bound of one of the geometric constraints. The largest infeasibility value for the geometric constraint is only 0.8 mm and is considered negligible in this study. Therefore, for this problem, the overlaying hyperbox is used to determine design variable information.

In this section, the preliminary vehicle design problem is introduced and results from the convergence and mapping studies are presented for this problem. It is seen from the convergence study results that a smaller number of system analyses can be used to obtain a representation of the Pareto set and the corresponding vehicle design configuration can be obtained using the results from the performance to design mapping study. In Section 6, concluding remarks and sources of future work are cited.

## **6 Conclusions and future work**

In this paper, studies are presented to analyse the convergence behaviour of a Multi-Objective Genetic Algorithm by reducing the number of function evaluations that can be performed. Though the process is application dependent, 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 applying the MOGA in two steps, where the first step of the MOGA uses some of the available function evaluations to cluster around one region of the Pareto set, and in the second step the MOGA uses the rest of the evaluations to populate the remaining regions 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 repeating the analyses. This is done by dividing the two spaces into discrete regions of indifference and then studying the relationships between the discrete regions in the performance and design spaces.

Future work in this area includes developing methods for studying MOGA convergence without explicitly running the algorithm exhaustively to generate a representation of the 'true' Pareto set. This would include the development and application of metrics to assess the goodness of a set of non-dominated designs obtained from a smaller number of function evaluations. Additionally, the performance to design mapping could also be expanded to include performance space points within the feasible

region so that the performance of any combination of design variable settings could be assessed whether or not its performance is Pareto optimal.

### **Acknowledgements**

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

### **References**

- Deb, K. and Jain, S. (2002) *Running Performance Metrics for Evolutionary Multiobjective Optimization*, KanGAL Report No. 2002004, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India.
- Eddy, J. and Lewis, K. (2001) 'Effective generation of pareto sets using genetic programming', *ASME Design Technical Conferences, Design Automation Conference*, DETC2001/DAC-21093.
- Eddy, J. and Lewis, K. (2002) 'Visualization of multi-dimensional design and optimization data using cloud visualization', *ASME Design Technical Conferences, Design Automation Conference*, DETC02/DAC-34130.
- Eschenauer, C.M., Koski, J. and Osyczka, A. (1990) *Multicriteria Design Optimization*, Springer-Verlag, New York.
- Fenyves, P.A., Donndelinger, J.A. and Bourassa, J-F. (2002) 'A new system for multidisciplinary design and optimization of vehicle architectures', *9th AIAA Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, AIAA-2002-5509.
- Ferguson, S. and Lewis, K. (2004) 'Effective development of flexible systems in multidisciplinary optimization', *10th AIAA Symposium on Multi-disciplinary Analysis and Optimization*, Albany NY, AIAA-2004-4309.
- Gu, X. and Fenyves, P.A. (2004) 'Application of the integrated framework for architecture development (IFAD) for conceptual vehicle design', *10th AIAA Symposium on Multidisciplinary Analysis and Optimization*, Albany NY, AIAA-2004-4315.
- Gurnani, A., Ferguson, S., Donndelinger, J. and Lewis, K. (2005) 'An approach to feasibility assessment in preliminary design', *ASME International Design Engineering Technical Conferences*, Design Automation Conference, Long Beach, CA, DETC2005-84853.
- Kasprzak, E. and Lewis, K. (2001) 'Pareto analysis in multiobjective optimization using the colinearity theorem and scaling method', *Structural and Multidisciplinary Optimization*, Vol. 22 No. 3, pp.208–218.
- Lee, Y.E. and Black, D.R. (2004) 'Improved RFQ process by optimization technique', *SAE Motorsports Engineering Conference and Exposition*, Dearborn, MI, 2004-01-1155.
- Pareto, V. (1906) *Manuale di Econòmica Polittica*, Società Editrice Libràia, Milan, Italy, translated into English by Schwier, A.S. as *Manual of Political Economy*, Macmillan, New York, 1971.
- van Veldhuizen, D.A. (1999) *Multiobjective Evolutionary Algorithm: Classifications, Analyses, and New Innovations*, PhD Dissertation, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.
- Wu, J. and Azarm, S. (2001) 'Metrics for quality assessment of a multiobjective design optimization solution set', *Journal of Mechanical Design*, Vol. 123, No. 1, pp.18–25.