Chapter 4
Many-Objective Evolutionary Optimisation and Visual Analytics for Product Family Design

Ruchit A. Shah, Patrick M. Reed and Timothy W. Simpson

Abstract Product family design involves the development of multiple products that share common components, modules and subsystems, yet target different market segments and groups of customers. The key to a successful product family is the product platform—the common components, modules and subsystems—around which the family is derived. The fundamental challenge when designing a family of products is resolving the inherent trade-off between commonality and performance. If there is too much commonality, then individual products may not meet their performance targets; however, too little sharing restricts the economies of scale that can be achieved during manufacturing and production. Multi-objective evolutionary optimisation algorithms have been used extensively to address this trade-off and determine which variables should be common (i.e., part of the platform) and which should be unique in a product family. In this chapter, we present a novel approach based on many-objective evolutionary optimisation and visual analytics to resolve trade-offs between commonality and many performance objectives. We provide a detailed example involving a family of aircraft that demonstrates the challenges of solving a 10-objective trade-off between commonality and the nine performance objectives in the family. Future research

R. A. Shah · T. W. Simpson (✉)
Industrial & Manufacturing Engineering, Pennsylvania State University,
University Park, USA
e-mail: tws8@psu.edu

R. A. Shah
e-mail: ruchit@psu.edu

P. M. Reed
Civil & Environmental Engineering, Pennsylvania State University,
University Park, USA
e-mail: preed@engr.psu.edu

L. Wang et al. (eds.), *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, DOI: 10.1007/978-0-85729-652-8_4,

directions involving the use of multi-objective optimisation and visual analytics for product family design are also discussed.

4.1 Balancing Commonality and Performance During Product Family Design

For most companies, product variety is a key to maintaining their market share. Today there are wide arrays of choices available for nearly all consumer products and services; thus, for a company to create a niche for itself its product offerings must be diverse enough to appeal to multiple market segments. However, offering a wide variety of products has its downsides as proliferation of product variety may incur substantial costs to the company [1–6] and reduce its profitability. Many companies struggle to provide variety in their product offerings while maintaining reasonably low costs. This often results from a company's failure to embrace commonality, compatibility, standardisation and modularisation across the product lines [7], which degrades a company's ability to achieve economies of scale across their production/manufacturing process.

Unique product offerings are advantageous to customers but expensive for companies to achieve. High product variety offers customers options customised to their specific needs and preferences but reduces the margins for the company as the increased price might not be proportional to the perceived value estimated by the customer. Commonality on the other hand is cost-effective for the company, but it can compromise customer needs and requirements. Increased commonality allows the company to share resources across products, decrease inventory and take advantage of economies of scale to reduce procurement costs [8]. However, if the products are too common, then they can lose their distinctiveness [9]. In a customer-driven and highly competitive marketplace a company must effectively balance customer preferences against its profitability and economic stability. Thus the challenge is to meet the individual customer's wants and needs while keeping overall costs low.

Developing product platforms and designing families of products based on these platforms is one way to address the challenge associated with sharing assets across the products [7]. Product family design involves concurrent design of multiple products that share common features, components and subsystems based on a common product platform [10]. Optimising the design of product families is the key to resolving the trade-off between the conflicting objectives of commonality and individual product performance. A successful design of a product family maximises the commonality as much as possible without sacrificing the distinctiveness of the individual products in the family.

Many researchers have focussed on multi-objective optimisation approaches for balancing the conflicting objectives of commonality and performance. Simpson [11] reviews and categorises over 40 such approaches. For instance, Nelson et al. [12]

use multi-objective optimisation to analyse the Pareto sets of two derivative products to find a suitable product platform for a family of nail guns. Fellini et al. [13, 14] use a similar approach to study a family of three automobiles with varying levels of commonality in the powertrain. Fujita et al. [15] perform a similar analysis for a family of two aircrafts. Fellini et al. [13, 14] introduce a shared penalty vector and performance loss constraints to study the Pareto sets of automotive bodies. Gonzalez-Zugasti et al. [16] use real options concepts to help select the most appropriate product family design from a set of alternatives; they also investigated the use of multi-objective optimisation to design modular product platforms [17, 18]. Allada and Rai [19] introduce an agent-based multi-objective optimisation framework to capture the Pareto frontier for module-based product families of power screwdrivers and electric knives. Simpson et al. [20] examine the trade-off between different levels of platform commonality within a family of three aircraft. Tseng and Jiao [21] use optimisation techniques to facilitate design for mass customisation, and Chidambaram and Agogino [22] present a catalogue-based optimisation strategy for customising goods. Finally, Nayak et al. [23] and Messac et al. [24] have proposed methods for using commonality indices as part of multi-objective optimisation for product family design.

Resolving the commonality–performance trade-off inherent in product family design yields a set of efficient or Pareto solutions where each solution is better than the other solutions in at least one other objective. Based on the size of the product family and number of decision variables, single-stage and multi-stage optimisation approaches exist to help determine the best design variable settings for the product family and individual variants within the family [11]. Single-stage approaches optimise the product platform and the family simultaneously whereas multi-stage approaches initially optimise the product platform followed by optimisation of the individual products in the family [25]. Single-stage approaches have been shown to yield the best overall performance for product family design problems [26]; however, the high dimensionality of single-stage optimisation problems poses computational challenges to many traditional methods. The curse of dimensionality and limitations of traditional methods have motivated researchers to approach these problems with multi-objective evolutionary algorithms (MOEAs). MOEAs evolve solutions through a process analogous to Darwinian selection [27] with search operators that mimic selection, mating and mutation. Over the past few decades, evolutionary algorithms have been extensively used to address a broad range of single- and multi-objective problems. MOEAs have been shown capable of approximating solution sets that compose the trade-offs for highly nonlinear, discrete and non-convex objective space landscapes [28–30].

This chapter presents a MOEA-based many-objective analysis of product family design to help resolve the trade-off between commonality and individual product performance in a product family. In this chapter the phrase "many-objective" refers generally to problems with four or more objectives and is an area of increasing interest in a range of applications [31, 32]. We also demonstrate the benefits of visual-analytic techniques [31–33] to analyse the high-dimensional trade-offs evolved by MOEAs and guide designers in identifying the best possible compromise solution based on their needs. In short, this chapter introduces a decision-making method for product family design based on many-objective MOEA search and visual analytics. Section 4.2 presents an overview of the process and introduces an example involving the design of a family of aircraft. We describe the problem parameters and state the problem formulation and constraints used for optimisation. Section 4.3 provides a brief description of the MOEA used in the study. Section 4.4 provides a detailed description of the computational experiment required to evolve high quality approximate solutions from the algorithm. Section 4.5 discusses the results and describes the use of visual analytics in the decision-making process. Finally, Sect. 4.6 provides key findings and recommendations for future work.

4.2 Method for Product Family Optimisation

4.2.1 Overview

As discussed earlier, performance and commonality are inherently conflicting objectives during the process of developing product family design. Many-objective optimisation provides a mechanism for discovering high-dimensional multivariate dependencies between performance objectives and exploiting these dependencies in negotiated trade-offs. This method requires both the identification of many-objective Pareto approximate solutions and interactive visual exploration. This chapter uses a many-objective product family design problem to demonstrate how designers can navigate the high-dimensional trade-off surfaces and make well-informed decisions.

Figure 4.1 provides an overview of our method which seeks to facilitate design insights and negotiated solution selection. Overall our many-objective visual design analytics work bridges the historical work in joint cognitive systems [34] and visual analytics [35]. Our framework in Fig. 4.1 approaches many-objective design as a “top-down sense-making” exercise [36, 37] by progressively increasing the complexity of trade-off representations presented to decision-makers while utilising solution filtering to focus their attention on key design discoveries and potential compromises. Initially, the problem formulation should be carefully constructed to reflect the key decisions and performance measures that will strongly shape designer preferences and potential conflicts. The formulation should be flexible enough such that if the designer’s requirements change then a new design can be created without reformulating and resolving the problem. Problem formulation dictates subsequent analysis, and thus considerable time should be spent to ensure its correctness and appropriateness for the product family at hand.

The next step after the problem formulation is the selection of an optimisation algorithm. The designer has to identify an optimisation algorithm that would provide a sufficient approximation for a given problem’s Pareto optimal set. It should be ensured that the results from the heuristic are repeatable and the results

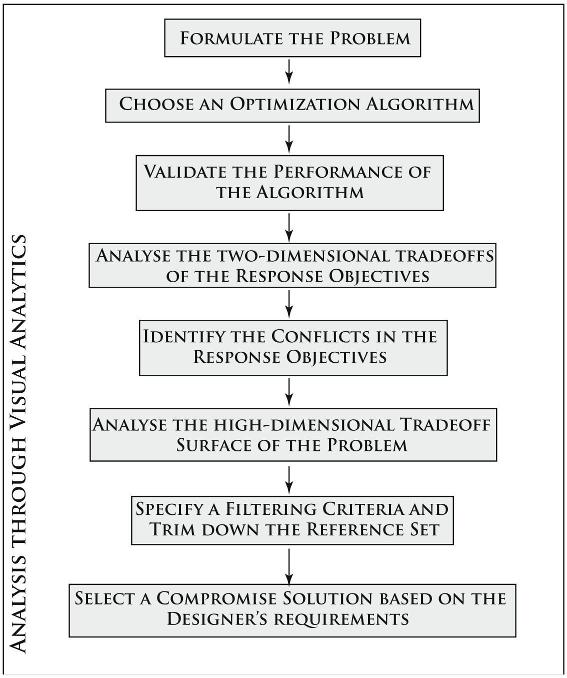

Fig. 4.1 Overview of suggested steps for exploiting many-objective optimisation and visualisation for improving product family designs

should be validated to measure the performance of the algorithm. The next stage is the most critical part of the method: analyses of the solution set. Decisions made after the analyses of the solution set are transformed into products. Success or failure of the product family can have a significant impact on the reputation and profitability of the company. Thus extreme care and diligence should be taken to fully exploit the information captured in a many-objective solution set to guide design decisions.

Extracting meaningful and relevant information from a high-dimensional solution set is a challenging task. The visual analytics method discussed in this chapter presents an organised and structured approach to filter out the relevant and useful information from the solution set. It gradually guides the designer from simple two-dimensional tradeoffs to high-resolution high-dimensional trade-off surfaces. It informs the designer on how each objective contributes to the problem and thus guides the designer to select his/her solution filtering criteria. The designer eventually thins out the solution set and selects the best compromise solution that fits his/her requirements. An important aspect of the method is that none of the decisions require perfect *a priori* knowledge of design preferences, variable interactions and constraints. These problem properties emerge with exploration of the solution set, which itself has the potential to reshape problem conceptions, expert decision rules/heuristics and designer preferences.

The rest of the chapter walks the reader through the method with the help of a real-world product family design problem. It highlights the challenges encountered during the problem formulation and the corresponding analyses. The example also shows how the proposed method can aid designers in discovering product family designs that offer the best compromises between their objectives.

4.2.2 Test Case Development

The General Aviation Aircraft (GAA) problem was introduced by Simpson et al. [38] as an example problem focussing on the design of a family of aircraft based on three different seating configurations. The term "General Aviation" encompasses all flights except military operations and commercial carriers. Potential buyers form a diverse group that includes weekend and recreational pilots, training pilots and instructors, travelling business executives and even small commercial operators. Satisfying a group with such diverse needs and economic potential poses a constant challenge for the General Aviation industry because a single aircraft cannot meet all of the market needs. Hence, the example seeks to design a family of three aircraft to accommodate two, four, or six people that can easily be adapted to satisfy distinct groups of customer demands. For this example, the configuration for the GAA is a fixed-wing, single engine, single-pilot, propeller-driven aircraft. The challenge is to determine the best values of top-level design specifications for the fuselage, wing and engine to satisfy a variety of performance and economic requirements. The problem parameters are described in the next section, which provides a detailed discussion of the problem formulation used in the study.

4.2.3 Problem Parameters

For this example the baseline configuration has been derived from a Beechcraft Bonanza B36TC, a four-to-six seat, single-engine business-and-utility aircraft, which is one of the most popular GAA sold. The general aircraft configuration has been fixed at three propeller blades, high wing position and retractable landing gear based on prior studies [38]. The design variables used in this study and the corresponding ranges of interest are mentioned in Table 4.1.

The General Aviation Synthesis Program (GASP) [39] is used to determine the aircraft sizing and performance estimates. Input variables for GASP are general descriptors of aircraft type, size and missing requirements. The numerical output from GASP includes various performance characteristics of aircraft such as empty weight (WEMP), fuel weight (WFUEL), direct operating cost (DOC) and maximum flight range (RANGE). To reduce the computational expense of performing these calculations, statistical approximations (i.e., response surface models) are employed to provide simplified, yet accurate, approximations of each performance parameter as a function of the input design variables [38].

Table 4.1 Design parameters and their respective ranges

S. No.	Design variable	Name	Units	Min	Max
1	Cruise speed	CSPD	Mach	0.24	0.48
2	Aspect ratio	AR	–	7	11
3	Sweep angle	SWEEP	degrees	0	6
4	Propeller diameter	DPROP	ft	5.5	5.968
5	Wing loading	WINGLD	lb/ft^2	19	25
6	Engine activity factor	AF	–	85	110
7	Seat width	SEATW	inch	14	20
8	Tail length/diameter ratio	ELODT	–	3	3.75
9	Taper ratio	TAPER	–	0.46	1

Table 4.2 Constraints and preferences for the performance parameters

S. No.	Performance parameters	Name	Units	Preference	Performance limits		
					2-seater	4-seater	6-seater
1	Takeoff noise	NOISE	dB	Min	75	75	75
2	Empty weight	WEMP	lb	Min	2200	2200	2200
3	Direct operating cost	DOC	$/h	Min	80	80	80
4	Ride roughness	ROUGH	–	Min	2	2	2
5	Fuel weight	WFUEL	lb	Min	450	475	500
6	Purchase price	PURCH	1970$	Min	–	–	–
7	Flight range	RANGE	nm	Max	2000	2000	2000
8	Max life/drag ratio	LDMAX	–	Max	–	–	–
9	Max cruise speed	VCMAX	kts	Max	–	–	–

There are a total of nine responses that are of interest for each aircraft: takeoff noise (NOISE), DOC, ride roughness (ROUGH), WEMP, WFUEL, purchase price (PURCH), maximum cruise speed (VCMAX), RANGE and lift/drag ratio (LDMAX). The constraint values and the min/max preferences for the performance variables are summarised in Table 4.2. Overall a product family design that satisfies the constraints; minimises the NOISE, WEMP, DOC, ROUGH, WFUEL, PURCH and maximises the RANGE, LDMAX and VCMAX is preferred.

4.2.4 Objective Formulation and Constraints

The problem was first solved using robust design methods embodied in the Robust Concept Exploration Method [38, 40]. Product variety trade-off studies were later performed using the compromise Decision Support Problem (DSP) for the family of aircraft [20]. In the prior work of the GAA problem, the values of the response variables were consolidated into one function to reflect the product-performance. A deviation function was adapted from goal programming to measure product performance, with lower deviations being preferred [41]. This approach requires the designer to specify target values for the response variables *a priori* to optimisation. With very little information available about the problem's objective space, specifying reasonable target values might be difficult, and the specified target values might not reflect the true requirements of the designer.

This chapter presents a novel approach to the problem formulation. Ideally, a designer would like to have the knowledge of the interaction between the various performance parameters and their respective contributions towards commonality and overall product performance. This information would enable the designer to design and introduce products that cater to specific performance parameters without sacrificing the commonality across the families. For instance, DOC (a response variable) might have a greater contribution to the market success of an aircraft as compared with the contributions of other response variables. Thus, a designer might seek to balance DOC and commonality to introduce products that have both economic and market viability. In other words, a designer would like to balance each of the performance parameter independently with the commonality objective to design products based on their requirements. However, such an approach requires solving a high-dimensional and far more complicated problem which may be too computationally expensive or intractable. Toward that end, this chapter introduces a novel approach to the problem formulation which seeks to resolve the trade-off between commonality and individual performance parameter using visual analytics.

During the optimisation process, we seek to find values of the design variables that optimise the performance parameters while maintaining high commonality. Ideally, one would like to optimise the performance parameters for each of the three (2-seater, 4-seater and 6-seater) aircraft in the family. With nine performance parameters per aircraft and three aircraft per family, it would lead to optimising 27 ($= 9 \times 3$) objectives in addition to the objective of maximising commonality within the family. It can be immensely challenging and overwhelming to analyse trade-offs for such high-dimensional problem; thus, we adopt a min–max/max–min optimisation approach and try to optimise the worst-case performance measure across the three product families. If a performance parameter has to be minimised on all the three product families, then a min–max criterion aims at constructing solutions that minimise the maximum performance value across the three aircraft. This formulation minimises the maximum deviation and ensures the best possible performance in the worst case. Similar explanation holds true for the max–min optimisation approach on the response metric that has to be maximised. A significant advantage of the min–max/max–min optimisation approach is that it ensures that variation of design variables does not degrade the performance of a specific aircraft significantly. Another advantage in terms of problem formulation is that the problem reduces from 27 independent performance objectives to nine robust performance objectives. A 10-objective problem (nine robust performance objectives plus the commonality objective) is relatively more tractable and easier to solve as compared with the original 28-objective problem.

To measure the commonality across the product families we use the Product Family Penalty Function (PFPF) developed by Messac et al. [24]. PFPF penalises the uniqueness within the product family by measuring the percentage variation of

Table 4.3 Objectives used in the General Aviation aircraft formulation

S. No.	Objectives	Value	Preference
1	Maximum NOISE	Max ($NOISE_2$, $NOISE_4$, $NOISE_6$)	Minimise
2	Maximum WEMP	Max ($WEMP_2$, $WEMP_4$, $WEMP_6$)	Minimise
3	Maximum DOC	Max (DOC_2, DOC_4, DOC_6)	Minimise
4	Maximum ROUGH	Max ($ROUGH_2$, $ROUGH_4$, $ROUGH_6$)	Minimise
5	Maximum WFUEL	Max ($WFUEL_2$, $WFUEL_4$, $WFUEL_6$)	Minimise
6	Maximum PURCH	Max ($PURCH_2$, $PURCH_4$, $PURCH_6$)	Minimise
7	Minimum RANGE	Min ($RANGE_2$, $RANGE_4$, $RANGE_6$)	Maximise
8	Minimum max LDMAX	Min ($LDMAX_2$, $LDMAX_4$, $LDMAX_6$)	Maximise
9	Minimum max VCMAX	Min ($VCMAX_2$, $VCMAX_4$, $VCMAX_6$)	Maximise
10	PFPF	–	Minimise

the design variables within the product family. The percentage variation of design variables is measured as follows:

$$\mathrm{pvar}_j = \frac{\mathrm{var_j}}{\bar{x}_i} \tag{4.1}$$

where,

$$\mathrm{var}_j = \sqrt{\frac{\sum_{(i=1)}^{p} (x_{ij} - \bar{x}_j)^2}{(p-1)}} \quad \text{and} \quad \bar{x}_j = \frac{\sum_{(i=1)}^{p} x_{ij}}{p} \tag{4.2}$$

x_{ij} is the value of the jth design variable for the ith product, $i = 1, 2, \ldots, p$ and $j = 1, 2, .., n$. PFPF is computed by summing the percentage variations of all n design variables across all p products:

$$\mathrm{PFPF} = \sum_{j=1}^{n} \mathrm{pvar}_j \tag{4.3}$$

Unlike many commonality indices available in the literature [42–44], the PFPF compares commonality not only on how many variables are common but also on how similar the values of the unique variables are to one other (i.e., parametric variation). Product families with high variation in design parameters (i.e., distinct inputs) have higher values of PFPF while product families with low variation in design parameters (i.e., common inputs) have lower values of PFPF. As high commonality is desired, a lower PFPF value is preferred. A summary of the problem objectives used in this study is given in Table 4.3.

As mentioned in Table 4.2, each aircraft is associated with certain performance limits. These are rigid constraints that establish the feasibility of each product. The violation of a constraint can be measured as follows:

$$c_{\mathrm{in}} = \begin{cases} \dfrac{(\text{value} - \text{limit})}{\text{limit}}, & \text{if value} > \text{limit} \\ 0, & \text{if value} \leq \text{limit} \end{cases}, \quad n \in (1,2,3), \quad i \in (1,2,\ldots,9) \tag{4.4}$$

The total constraint violation for the product family is computed by summing the violation of each constraint for each aircraft.

$$\mathrm{CV} = \sum_{i=1}^{3} (c_{1i} + c_{2i} + c_{3i} + c_{4i} + c_{5i} + c_{6i}) \tag{4.5}$$

In summary, the size of the product family optimisation problem in this study is: 27 design variables, 10-objectives and 1 constraint ($\mathrm{CV} < 0$).

4.3 Optimisation Algorithm Selection

The Epsilon-dominance Nondominated Sorted Genetic Algorithm-II (ε-NSGAII) is based on the NSGAII [45] an elitist MOEA. NSGAII uses a non-domination sorting approach to classify solutions according to the level of non-domination and a crowding distance operator to maintain solution diversity across approximation solution sets. The ε-NSGAII developed by Kollat and Reed [46, 47] reduces the extensive parameter calibration by using the concepts of ε-dominance archiving [48, 49], adaptive population sizing [50] and self-termination. The ε-NSGAII has been validated extensively across a suite of test problems and applications [46] and has been shown to perform as well or better than state-of-the-art MOEAs [47, 51].

The ε-NSGAII algorithm generates an initial small random population and uses non-domination and crowding distance to assign fitness to each individual. A non-domination sort is performed across all the solutions, and individuals are classified into fronts based on their ranks, with rank 1 assigned to the solutions that are non-dominated. Additionally, crowding distance is calculated for all individuals based on the average Euclidean distance between an individual and the individuals within the population which are assigned the same rank. Selection is done using binary tournaments and is based on the rankings and crowding distances of the individuals with a preference given to larger crowding distance. Individuals with larger crowding distance add to the diversity of the population and help to ensure that the ε-NSGAII explores the entire trade-off landscape. Selected individuals now become parents of the next generation, and the evolution process is repeated. These individuals are also eligible to enter an offline archive that stores the best solutions throughout the run. To achieve entry into the archive, individuals should be ε-non-dominated with respect to solutions in the archive.

The ε-NSGAII thereafter uses a series of "connected runs" to inject the archive solutions into the population of the next run using a 25% injection scheme. The injection scheme requires that the present archive forms 25% of the next population and the remaining 75% is filled with randomly generated individuals. This assists the performance of the ε-NSGAII by directing the search towards previously known good solutions; however the 75% random solutions help to ensure that the algorithm does not pre-converge while encouraging the exploration of new regions in the objective space. The algorithm can increase or decrease its population size as the search progresses and adapts its population based on the solutions obtained.

The ε-dominance archive allows the user to control the computational costs of evolution by specifying their precision requirements for each of the objectives. Based on the user's preferences, the algorithm applies a grid to the search space of the problem that can significantly reduce its computational costs when solving multi-objective problems by avoiding unnecessary precision in calculations [52] Larger ε values result in a coarser grid (and ultimately fewer trade-off solutions) while smaller ε values produce a finer grid. The fitness of each solution is then mapped to a box fitness based on the specified ε values. Non-domination sorting is then conducted using each solution's box fitness, and solutions with identical box fitness (i.e., solutions that occur in the same grid block) are compared, and those that are dominated within the grid block are eliminated. Only a single non-dominated solution is permitted in any one grid block, preventing clustering of solutions and promoting a more diverse search of the objective space. We refer the reader to Laumanns et al. [48] and Deb et al. [49] for additional details. Meanwhile, dynamic population sizing allows ε-NSGAII to start with a small initial population to pre-condition the search at a low computational cost in terms of the number of function evaluations. When the size of the ε-dominated archive stabilises, the connected runs are equivalent to a diversity-based EA search enhancement recommended by Goldberg [53] termed *time continuation*, where diverse search is sustained as long as it is required or feasible. Prior work using the ε-NSGAII by Kollat and Reed [46, 47] can be referenced for more details on the algorithm and its dynamic search features.

4.4 Computational Experiment

The ε-NSGAII was used to approximate the trade-offs in the family of aircraft and its evolutionary operators were parameterised as follows: probability of cross-over—pc = 1.0, probability of mutation—pm = 0.04, cross-over distribution index—gc = 15 and the mutation distribution index—gm = 20. The ε-NSGAII's adaptive population sizing was initialised using 152 individuals, and maximum number of function evaluations per trial was set at 500,000. Epsilon resolution settings (ε) for the 10 objectives are given in Table 4.4. These values represent the precision with which each objective is quantified and were chosen to represent the full precision Pareto-optimal set. Since MOEA search is initialised with randomly generated populations and as evolutionary operators are probabilistic, the process can yield high variability in search efficiency and reliability. It is standard practice to overcome this variability by running MOEA for a distribution of ''seeds'' for the random number generator which is used to initialise and guide their probabilistic search. In this study, our analysis across the 10-objective GAA problem was characterised using 50 random seed trial runs.

Table 4.4 Epsilon settings and ranges of the objectives

S. No.	Objectives	Name	ε	Range	
				Min	Max
1	Maximum NOISE	MAX_NOISE	0.05	73.25	74.46
2	Maximum WEMP	MAX_WEMP	10	1879.20	2032.91
3	Maximum DOC	MAX_DOC	2	58.67	80.00
4	Maximum ROUGH	MAX_ROUGH	0.01	1.81	2.00
5	Maximum WFUEL	MAX_WFUEL	10	367.87	500.00
6	Maximum PURCH	MAX_PURCH	1000	41901.85	44925.33
7	Minimum RANGE	MAX_RANGE	50	2000.00	2496.87
8	Minimum max LDMAX	MAX_LDMAX	0.1	14.20	16.00
9	Minimum max VCMAX	MAX_VCMAX	1	185.33	200.17
10	PFPF	PFPF	0.1	0.07	2.50

4.5 Results and Discussions

This chapter aims to evolve the non-dominated trade-off for a 10-objective problem and demonstrate the value of visual analytics in understanding key trade-offs. Solving such high-dimensional problems is a challenging proposition for most of the domination-based MOEAs. It becomes extremely difficult to effectively parameterise the algorithm and effectively guide the evolution process. Many authors have highlighted that on high-dimensional problems (objectives more than five or six) that some MOEAs may struggle in evolving high quality approximation sets and in some cases devolve into a “random walk” [31, 54–56]. As a test of the value and quality of the ε-NSGAII attained results, we have utilised a Monte-Carlo analysis to establish a pure random search baseline for the GAA problem where any selected trial solution is fully independent of any previous choice and its outcome [30]. If the results of the Monte-Carlo simulations are comparable with those obtained from the optimisation algorithm, then it negates the value of the optimisation algorithm and may also highlight the ease of solving a product family design problem such as this.

To validate the performance of the ε-NSGAII and test the quality of solutions generated by it, the results obtained from the optimisation algorithm were compared against the results obtained from Monte-Carlo simulation. The optimisation algorithm had 50 random trials with 500,000 function evaluations per trial. Thus, the algorithm used a total of 25 million (=50 $\times$ 500,000) function evaluations to generate a non-dominated set for the 10-objective GAA problem. The comparison was biased towards the Monte-Carlo simulation as it was allowed to generate 50 million samples (twice the number of function evaluations used by ε-NSGAII) to identify the non-dominated set for the problem. Table 4.5 presents a summary of run results from both approaches.

Comparative analysis of the Monte-Carlo simulation and ε-NSGAII revealed some interesting insights about the objective space of the problem. Of the 50 million random samples generated by the Monte-Carlo simulation study, only four solutions

Table 4.5 Number of solutions by Monte-Carlo simulation and the optimisation algorithm

Method	Total number of non-dominated solutions generated	Contribution to the reference set
Monte-Carlo simulation	4	0
Optimisation algorithm	16900	16900

were found feasible. The identification of only four feasible solutions from a set of 50 million solutions shows that the GAA problem is heavily constrained with respect to its performance parameters and a challenging overall search space. In the GAA's 27-dimensional decision (input) space it is almost impossible to randomly pick a point that would be feasible in the objective space. On the other hand ε-NSAGII generated a non-dominated set of 16,900 solutions from the 25 million function evaluations. On each run of the algorithm, ε-NSGAII found its first feasible solutions after only 1000–2000 function evaluations. The algorithm struggled for brief duration during the onset of a run; however, once it identified a feasible solution it quickly adapted and redirected its search to the favourable region of the objective space to generate more feasible solutions. Non-domination sorting of results from the Monte-Carlo simulation study and the ε-NSGAII indicated that the four solutions generated by the Monte-Carlo simulation were dominated by the solutions generated by the ε-NSGAII.

Superior performance of ε-NSGAII can be attributed to the use of ε-dominance archiving and adaptive population sizing. The combination of adaptive population sizing and epsilon archiving represents a diversity enhancement that also ensures stable and bounded archiving of high-dimensional approximation sets. As the dimensionality of a problem's objective space increases, generally the size of their Pareto-optimal solution sets grows rapidly yielding an impediment to search that Purshouse and Fleming [54] termed "dominance resistance". Dominance resistance represents the increasing difficulty of converging a high-dimensional set towards Pareto-optimality. In ε-NSGAII, the population size grows commensurate with the ε-dominance archive. In this strategy it controls the dominance resistance by setting epsilons [48] and uses archive size as a proxy for problem difficulty that triggers increases in the population size. Increased population sizes serve to both add diversity and selective pressure due to the truncation selection used in the ε-NSGAII algorithm framework. Moreover, ε-dominance archiving provides a theoretical bound to the approximation set size and population size [57].

The comparative analyses justified the need of an optimisation algorithm like ε-NSGAII to solve the problem. Thus a reference set was generated by pooling the non-dominated solutions across the 50 runs of the optimisation algorithm. The reference set consisted of 16,900 solutions. Analysing the high-resolution trade-off solutions on a 10-dimensional objective space can be difficult and overwhelming for a designer. Thus before analysing the high-dimensional trade-off we analyse relatively simple and easier to understand two-dimensional trade-offs of the performance parameters. Two-dimensional trade-offs provide valuable insights about the performance parameters and their mutual interactions and are much simpler than analysing 10-objectives at a time. Having some prior knowledge about the mutual interactions of performance parameters assists the designer in analysing the high-dimensional trade-off surface by eliminating the redundant information content.

A 10-objective problem yields 45 two-dimensional trade-offs. Figure 4.2 highlights some of the interesting two-dimensional trade-offs identified by the algorithm. The colour in the subplots indicates the performance on the commonality objective. Blue solutions indicate high commonality, and green solutions indicate low commonality. In each subplot the solutions highlighted with a red-coloured outline represent the trade-off for the corresponding set of objectives.

Figure 4.2a shows the interactions between PURCH and WEMP, both of which are to be minimised. The plot clearly indicates that there is a strong positive correlation between the PURCH and WEMP: an increase in WEMP results in an increase in PURCH. As the two objectives are positively correlated the trade-off solution set for this sub-problem essentially reduces to one solution. Figure 4.2b represents the interactions between the WFUEL and WEMP, where both the objectives are to be minimised. The plot shows there is a strong negative correlation between the two objectives, indicating a strong conflict between the two objectives. Thus a design with low WEMP results in higher WFUEL and vice versa. Figure 4.2c and d represent the interactions between LDMAX and ROUGH, and RANGE and ROUGH ,respectively. RANGE and LDMAX are to be maximised and ROUGH has to be minimised. The plots indicate that as the RANGE and LDMAX increases, there is proportional increase in ROUGH. Figure 4.2e and f represent the interactions between the DOC and NOISE, and RANGE and NOISE, respectively. DOC and NOISE are to be minimised and VCMAX is to be maximised. An interesting aspect of the trade-off seen here is that there is a steep drop in DOC (and a steep rise in VCMAX) for a relatively small increase in NOISE. However, beyond a threshold (65 for DOC and 198 for VCMAX) any further decrease in DOC (or increase in VCMAX) requires a significant increase in NOISE.

While Fig. 4.2 presents only a small subset of the 45 two-dimensional plots, it presents valuable information to the designer. It highlights the facts that while the designer is optimising WEMP, the PURCH is also being optimised; meanwhile, the designer cannot optimise WFUEL and WEMP at the same time—he/she will need to prioritise one over the other. Furthermore, the designer cannot target extreme performances for DOC and VCMAX as they might result in unacceptable values for NOISE which is a constraint. The rest of the two-dimensional plots can be analysed to extract further information about the performance parameters and their behaviour. In summary, a few other strong relationships observed in other plots were: (1) decreases in WEMP results in increases in VCMAX and (2) low PURCH results in high WFUEL. Having some prior information about the interaction between the performance objectives the designer is better placed to analyse the complete reference set.

The two-dimensional analysis in Fig. 4.2 informs the designer about the strong trade-offs for the problem, and this information can be used to when visualising the

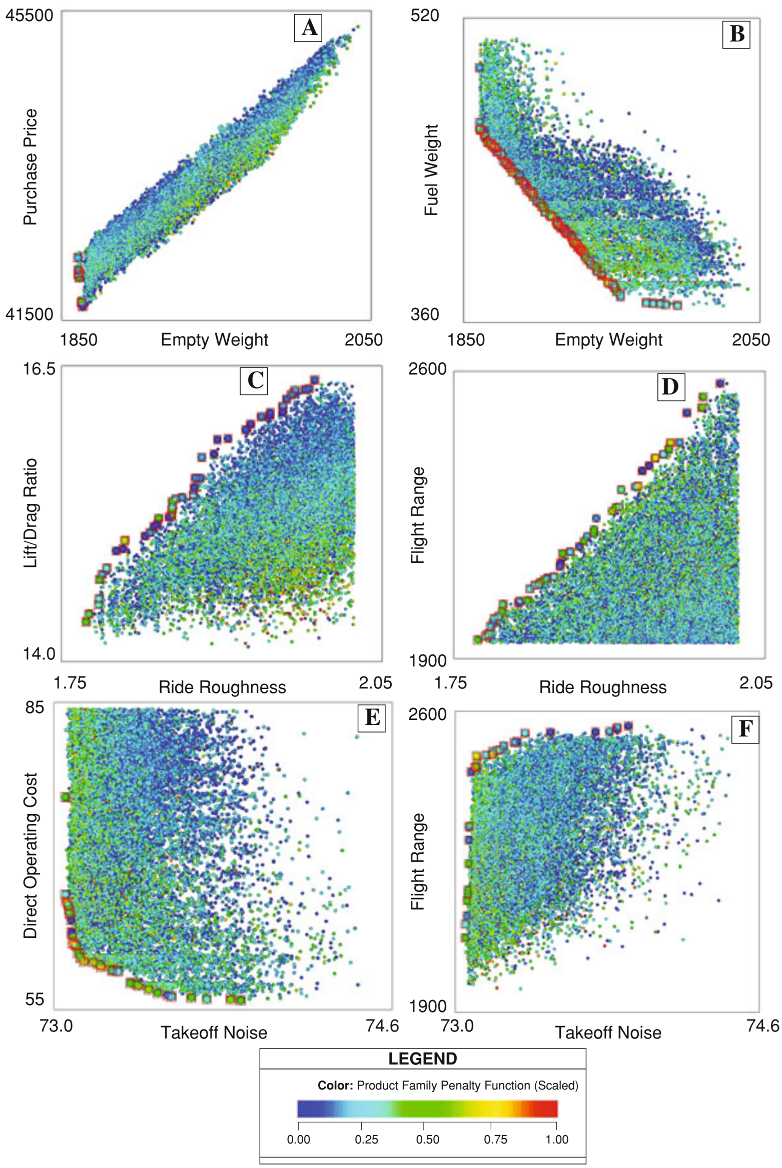

Fig. 4.2 Scatter plot analysis of the two objective interactions as subspaces of the overall 10-objective formulation

higher-dimensional reference set. Figure 4.3 represents the reference set for the 10-objective GAA problem. It uses the information captured in Fig. 4.2 to organise the response objectives into corresponding axes such that it highlights the conflicts at the higher-dimension. The WEMP, DOC and WFUEL objectives have been

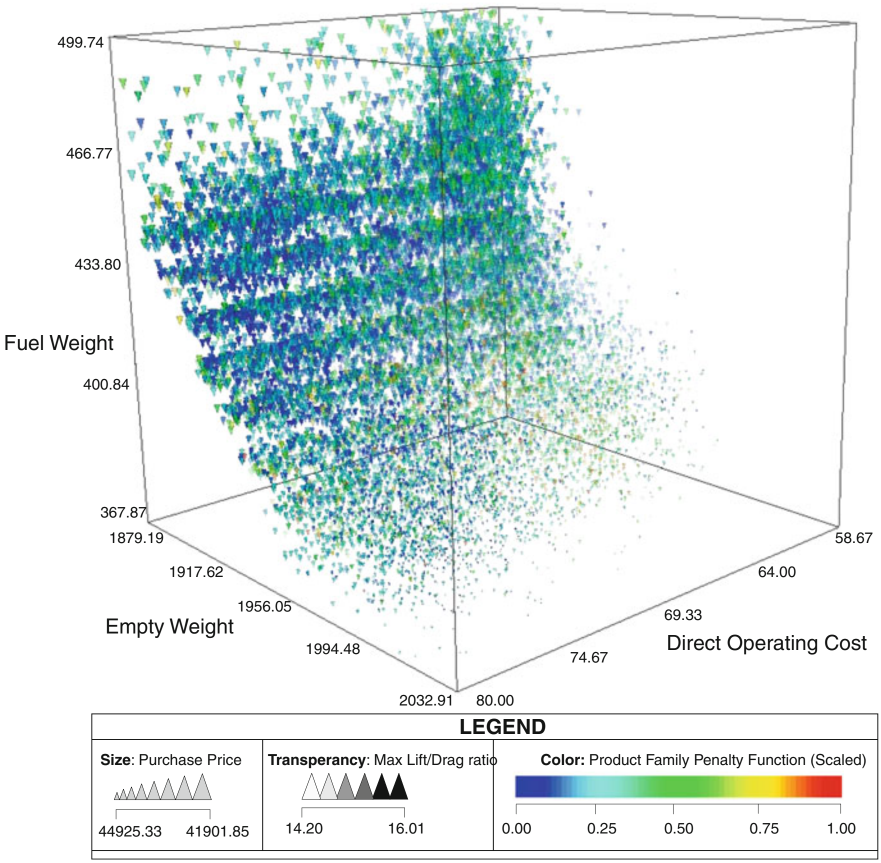

Fig. 4.3 Reference set

plotted on X-, Y- and Z-axes respectively. PURCH is represented by the size of the cones, which is scaled so that smaller cones represent higher PURCH while larger cones represent lower PURCH. Transparency is used to represent the LDMAX with lighter cones representing lower values of LDMAX and darker cones representing higher values of LDMAX. The colours of the cones represent the scaled values of the PFPF. Blue cones represent low PFPF values (high commonality) and red cones represent high PFPF values (low commonality). The values of PFPF range from 0.065 to 2.5.

Objective values for this non-dominated set range from 367.87 to 500lbs on WFUEL, 1879–2032lbs on WEMP, 58–80\$/h on DOC, 41901.85–44925.33 (1970\$) on PURCH, 14.20–16 on LDMAX and 0.065–2.5 on PFPF. For the ease of understanding and analysis the PFPF values have been scaled to vary between 0 and 1, with 0 representing high commonality and 1 representing no commonality. PFPF values have been scaled using the following equation:

$$PFPF_{Scaled} = \frac{PFPF - 0.065}{2.5 - 0.065} \quad (4.6)$$

WFUEL, WEMP, DOC, PURCH and PFPF (scaled) are to be minimised whereas LDMAX has to be maximised. As discussed earlier, the figure shows a clear conflict between the WFUEL and WEMP. LDMAX is in conflict with DOC as the solutions on the farther end of DOC axis yield more favourable performance on LDMAX. To optimise commonality in the family, a designer should focus on dark, large, blue solutions.

Ideally, a designer would like to optimise the product performance while maintaining high commonality (low PFPF). In other words, the designer's focus would be to maximise the commonality (low PFPF) and then subsequently get the best performance measure available from the performance parameters. Thus, we would like to concentrate only on the solutions with high commonality and eliminate the low commonality solutions.

Figure 4.4 filters out the low commonality from the set and displays only the top 5% ($PFPF_{Scaled}$ values from 0 to 0.05) of the high commonality solutions. The figure clearly shows there are two conflicting regions of the objective space: (1) Regions A and B, and (2) Region C. Region A (marked in blue box) offers high WFUEL, low PURCH, low WEMP, low DOC and low LDMAX. The PFPF values for the solutions in Region A are in the range of 0.04–0.05. Region B (marked in green box) offers low WFUEL, high WEMP, high PURCH, high DOC and high LDMAX. The PFPF values for the solutions in Region B are in the range of 0.04–0.05. Thus, Regions A and B offer designers a few conflicting design options while maintaining relatively high commonality. Based on his/her priorities she/he can focus on a specific region of the objective space. However, if a designer is interested in extremely high commonality ($PFPF_{Scaled}$ values from 0–0.01) there is a small trade-off region available in terms of Region C. Region C offers solutions that compromise on the values of the performance metrics to yield extremely low PFPF values. Thus if a designer is willing to sacrifice performance and is willing to accept a compromise value then s/he can design products with extremely high commonality.

Figure 4.5 shows solutions highlighted in Fig. 4.4 on a parallel coordinate plot. Parallel coordinate plots help visualise the performance across many-objectives simultaneously [58]. The vertical axes represent the individual objectives. The lower and higher end values of the vertical axes represent the ranges of the objective values. Each coloured line represents a solution with its intersection point on the vertical axes representing the performance on the corresponding objective. Colour coding of the boxes in Fig. 4.4 correspond to the colour coding of solutions in Fig. 4.5. Blue solutions represent solutions from Region A in Fig. 4.4, green solutions represent the solutions from Region B and red solutions represent the solutions from Region C. As discussed earlier, the blue set performs favourably on WEMP, DOC, PURCH, RANGE, VCMAX and performs fairly well on PFPF. The green set on the other hand performs favourably on WFUEL, LDMAX and shows similar performance as the blue set on the PFPF objective. The red set offers the compromise performance values on most of the objectives and the best performance value for ROUGH and PFPF. The

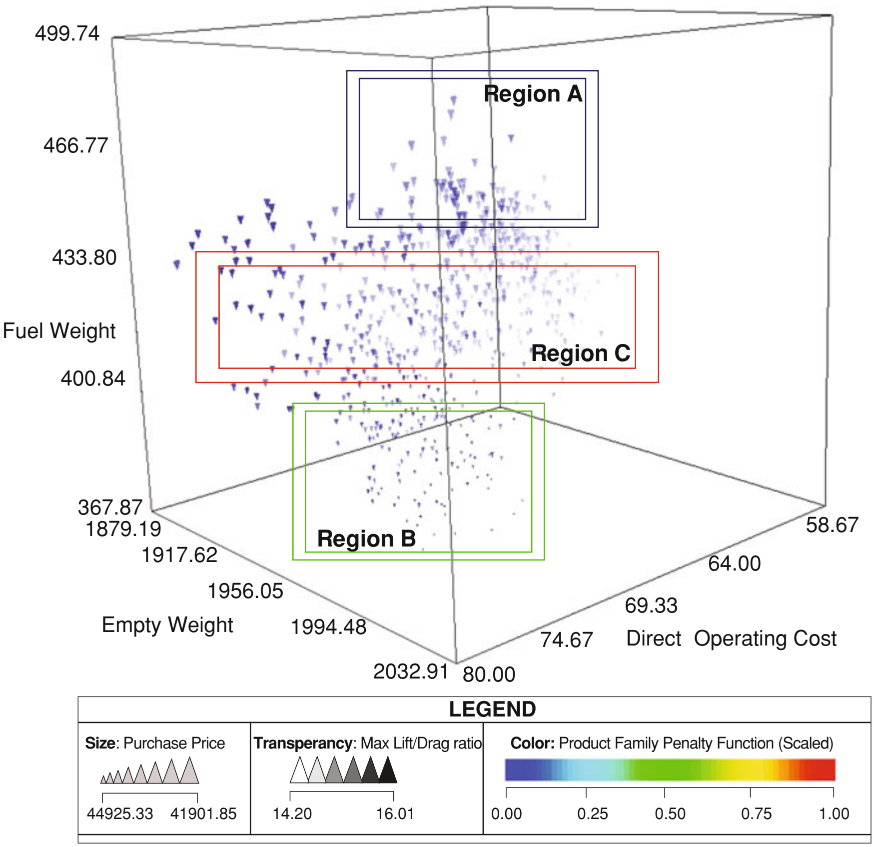

Fig. 4.4 Brushed reference set

dotted solutions represent the best possible compromise in their respective region. The blue dotted line represents the best possible compromise solution among all the solutions of Region A. It represents the best performance for DOC, ROUGH, RANGE, LDMAX and VCMAX while maintaining a fairly decent level of commonality. The green dotted line represents the compromise solution in the Region B with favourable performances on WFUEL, PURCH and LDMAX. Finally, the red dotted line represents a high commonality (3 inputs common across all three aircraft) solution with acceptable compromise values on the rest of the performance objectives.

Depending on the market demands and economic viability the designer can select a suitable product family design from either of the regions. This visual analytics-based approach gives the designer the flexibility in terms of focussing on specific performance metrics while maintaining high commonality. Thus a designer can create dedicated and customised products for various market segments without compromising on the commonality of the products.

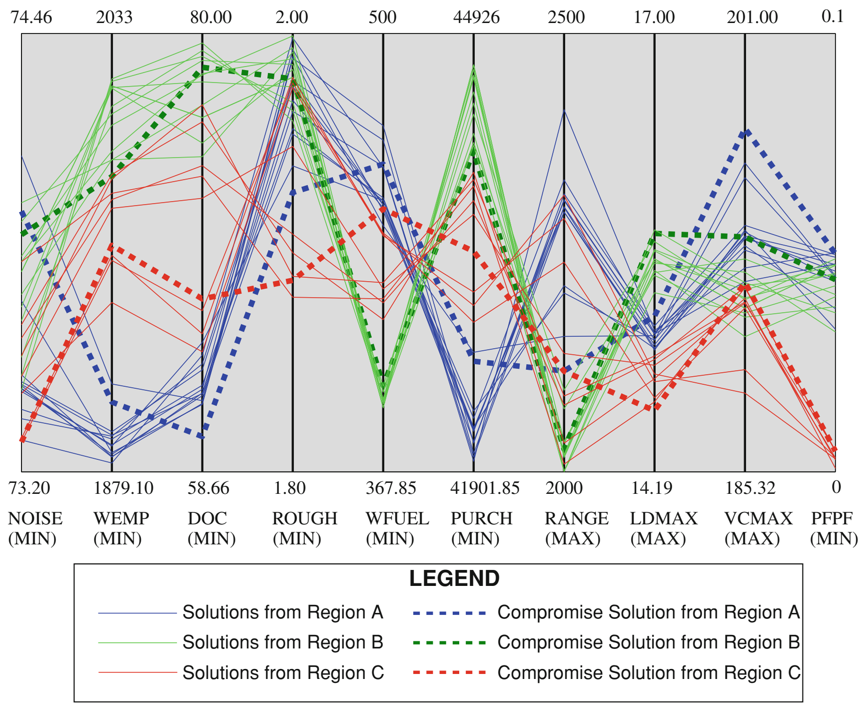

Fig. 4.5 Parallel coordinate plot for solutions displayed in Fig. 4.4

4.6 Conclusions

Multi-objective optimisation provides a useful tool for resolving the trade-offs between commonality and individual product performance (i.e., distinctiveness) within a family of products. This chapter presents a novel method to the product family optimisation based on MOEA and visual analytics. It uses a General Aviation Aircraft (GAA) example to demonstrate the relative merits of the proposed method to optimise a family of products for specific market needs without sacrificing commonality across the family. It introduces a 10-objective robust problem formulation where each objective represents a different performance parameter in the family. This formulation expands the dimensionality of the problem and seeks to resolve the trade-off between commonality and individual performance parameters. MOEAs are known to struggle on such high-dimensional problems, and selecting an algorithm that effectively solves the problem was a challenge. The ε-NSGAII has been shown to perform reasonably well on high-dimensional real-world problems. It uses the concepts of ε-dominance archiving and adaptive population sizing to balance the dominance resistance associated with high-dimensional problems. Thus, ε-NSGAII was used to resolve the trade-off for the 10-objective problem and the results were benchmarked against a Monte-Carlo simulation. The results indicate the GAA is a highly constrained problem, thus making it virtually impossible to generate a feasible solution through random search. The ε-NSGAII on the other hand performs well, navigating the search space to identify feasible solutions.

The proposed method integrates the use of visual analytical-techniques to gain insight into the high-dimensional trade-off surfaces generated by the algorithm. We illustrate some of the tools that are available for designers to identify strong conflicts between the performance parameters. These tools inform designers about the trade-offs for the problem, and this information is further used to effectively visualise the higher-dimensional reference set. The method allows designers to reduce the full-resolution set to a tractable set focussing on the most relevant and useful information. Aware of the interactions between the performance parameters, the designer can select the best compromise solution from the reduced set to satisfy his/her requirements. The key aspect of this method is that it does not require *a priori* knowledge of the problem, and it provides designers with a plethora of solutions from which to choose. Designers can enjoy the flexibility of creating a wide variety of products customised to different market segments without re-solving the problem.

The proposed method provides an efficient way to analyse the high-dimensional trade-offs for many-objective problems. The method explains through an example, on how best compromise solutions can be identified from a high-resolution high-dimensional trade-off surface. Future work can be based on improving the method by including the designer's requirements as an integral part of the method while allowing for interactivity between designers and the optimisation algorithms as solutions evolve. This also includes developing effective strategies for dealing with high-dimensional input spaces and handling product family problems with many product variants. Finally, extending the visual analytics techniques to help identify platform variables within a product family would be beneficial to designers.

Acknowledgments The first and second authors were partially supported by the National Science Foundation (NSF) under CAREER Grant No. CBET-0640443, and the third author acknowledges support from NSF Grant No. CMMI-0620948. The computational experiments in this work were supported in part through instrumentation funded by NSF Grant No. OCI-0821527. Any opinions, findings and conclusions or recommendations in this chapter are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Anderson, D.M. (1997). Agile product development for mass customization: How to develop and deliver products for mass customization, Niche Markets, JIT, Build-to-Order and Flexible Manufacturing. Chicago, IL: Irwin.
2. Galsworth, G. D. (1994). *Smart, simple design: Using variety effectiveness to reduce total cost and maximize customer selection*. Essex Junction, VT: Omneo.
3. Ho, T. H., & Tang, C. S. (1998). *Product variety management: Research advances*. Boston, MA: Kluwer Academic Publishers.

4. Child, P., Diederichs, R., Sanders, F.-H., & Wisniowski, S. (1991). The management of complexity. *Sloan Management Review, 33*(1), 73–80.
5. Ishii, K., Juengel, C, & Eubanks, C.F. (1995). Design for product variety: Key to product line structuring. *Proceedings of the ASME Design Engineering Technical Conferences—Design Theory and Methodology* (pp. 499–506). Boston, MA 83(2)
6. Lancaster, K. (1990). The economics of product variety. *Marketing Science, 9*(3), 189–206.
7. Meyer, M. H., & Lehnerd, A. P. (1997). *The power of product platforms: Building value and cost leadership, Free Press*. NY: New York.
8. Thevenot, H. J., & Simpson, T. W. (2007). A comprehensive metric for evaluating commonality in a product family. *Journal of Engineering Design, 18*(6), 577–598.
9. Robertson, D., & Ulrich, K. (1998). Planning product platforms. *Sloan Management Review, 39*(4), 19–31.
10. Simpson, T. W., Siddique, Z., & Jiao, J. (Eds.). (2005). *Product platform and product family design: methods and applications*. New York: Springer.
11. Simpson, T. W. (2005). Methods for optimizing product platforms and product families: Overview and classification. In T. W. Simpson, Z. Siddique, & J. Jiao (Eds.), *Product platform and product family design: Methods and applications* (pp. 133–156). New York: Springer.
12. Nelson, S. A., I. I., Parkinson, M. B., & Papalambros, P. Y. (2001). Multicriteria optimization in product platform design. *ASME Journal of Mechanical Design, 123*(2), 199–204.
13. Fellini, R., Kokkolaras, M., Michelena, N., Papalambros, P., Saitou, K., Ferez-Duarte, A., & Fenyes, P.A. (2002). A sensitivity-based commonality strategy for family products of mild variation, with application to automotive body structures. *Proceedings of the 9th AIAA/ ISSMO Symposium on Multidisciphnary Analysis and Optimization, Atlanta*, GA, AIAA, AIAA-2002-5610.
14. Fellini, R., Kokkolaras, M., Papalambros, P., & Perez-Duarte, A. (2002). Platform selection under performance loss constraints in optimal design of product families. *Proceedings of the ASME Design Engineering Technical Conferences—Design*
15. Fujita, K., Akagi, S., Yoneda, T., & Ishikawa, M. (1998). Simultaneous optimization of product family sharing system structure and configuration. *Proceedings of the ASME Design Engineering Technical Conferences, Atlanta*, GA, ASME, Paper No. DETC98/DFM-5722.
16. Gonzalez-Zugasti, J. P., Otto, K. N., & Baker, J. D. (1999). 12–15 September. *Assessing value for product family design and selection*. Advances in Design Automation, Las Vegas, NV, ASME, Paper No. DETC99/DAC-8613.
17. Gonzalez-Zugasti, J.P., & Otto, K.N. (2000). Modular platform-based product family design. *Proceedings of the ASME Design Engineering Technical Conferences—Design Automation Conference, Baltimore*, MD, ASME, Paper No. DETC-2000/DAC-14238.
18. Gonzalez-Zugasti, J. P., Otto, K. N., & Baker, J. D. (2000). A method for architecting product platforms. *Research in Engineering Design, 12*(2), 61–72.
19. Allada, V., & Rai, R. (2002). *Module-based multiple product design*, IIE Annual Conference 2002, Orlando, FL, IIE.
20. Simpson, T. W., Seepersad, C. C., & Mistree, F. (2001). Balancing commonality and performance within the concurrent design of multiple products in a product family. *Concurrent Engineering: Research and Applications, 9*(3), 177–190.
21. Tseng, M.M., & Jiao, J. (1998). Design for mass customization by developing product family architecture. *ASME Design Engineering Technical Conferences—Design Theory and Methodology, Atlanta*, GA, ASME, Paper No. DETC98/DTM-5717.
22. Chidambaram, B., & Agogino, A. M. (1999). 12–15 September. Catalog-based customization. *Advances in Design Automation*, Las Vegas, NV, ASME, Paper No. DETC99/DAC-8675.
23. Nayak, R. U., Chen, W., & Simpson, T. W. (2002). 10–13 September. A variation-based method for product family design. *Engineering Optimization*, 34(1), 65–81.
24. Messac, A., Martinez, M. P., & Simpson, T. W. (2000). Introduction of a product family penalty function using physical programming. 8th AIAA/NASA/USAF/ISSMO *Symposium*

on Multidisciplinary Analysis and Optimization, Long Beach, CA, AIAA, AIAA-2000-4838, to appear in ASME Journal of Mechanical Design.

25. Khajavirad, A., Michalak, J. J., & Simpson, T. W. (2009). An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality. *Structural and Multidiscipilnary Optimization, 39*(2), 187–201.
26. Messac, A., Martinez, M. P., & Simpson, T. W. (2002). Effective product family design using physical programming. *Engineering Optimization, 34*(3), 245–261.
27. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. New York, NY: Addison-Wesley Publishing.
28. Back, T., Fogel, D., & Michalewicz, Z. (2000). *Handbook of evolutionary computation*. Bristol, UK: Oxford University Press.
29. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. New York: John Wiley & Sons LTD.
30. Coello, C. C., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*. New York, NY: Kluwer Academic Publishers.
31. Fleming, P. J., Purshouse, R. C., & R. J. Lygoe (2005), Many-objective optimization: An engineering design perspective, in Evolutionary Multi—Criterion Optimization, ser. Lecture Notes in Computer Science. Springer: Berlin, Heidelberg pp. 14–32.
32. Kasprzyk, J. R., Reed, P. M., Kirsch, B., & Characklis, G. *Managing population and 761 drought risks using many-objective water portfolio planning under uncertainty*, Water Resources 762 Research, doi:10.1029/2009WR008121.
33. Kollat, J. B., & Reed, P. M. (2007). A framework for visually interactive decisionmaking and design using evolutionary multiobjective optimization (VIDEO). *Environmental Modelling and Software, 22*(12), 1691–1704.
34. Woods, D. (1986). “Paradigms for intelligent decision support,” Intelligent Decision Support in Process Environments, Springer, New York, NY
35. Keim, D. A., Mansmann, F., Schneidewind, J., & Ziegler, H. (2006). Challenges in visual data analysis. *Proceedings of Information Visualization, IEEE Computer Society* 9–16, London, UK.
36. Russell, D. M., Stefik, M. J., Pirolli, P., & Card, S. K. (1993). The cost structure of sensemaking. *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, Amsterdam, The Netherlands, April 24–29.
37. Qu, Y., & Furnas, G. W. (2005). Sources of structure in sensemaking. *Proceedings of the SIGCHI ‘05 Conference on Human Factors in Computing Systems*, ASM Press: Portland, OR, April 2–7.
38. Simpson, T. W., Chen, W., Allen, J. K., & Mistree, F. (1996). 4–6 September. Conceptual design of a family of products through the use of the robust concept exploration method. 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, AIAA, Vol. 2, pp. 1535–1545. AIAA-96-4161-CP.
39. NASA. (1978). GASP—General aviation synthesis program, NASA CR-152303, Contract NAS 2-9352, NASA Ames Research Center, Moffett Field, CA.
40. Simpson, T. W., Chen, W., Allen, J. K., & Mistree, F. (1999). Use of the robust concept exploration method to facilitate the design of a family of products. In U. Roy, J. M. Usher, et al. (Eds.), *Simultaneous engineering: Methodologies and applications* (pp. 247–278). Amsterdam, The Netherlands: Gordon and Breach Science Publishers.
41. Mistree, F., Hughes, O.F., & Bras, B.A. (1993). The compromise decision support problem and the adaptive linear programming algorithm, In: M. P. Kamat (ed.), *Structural optimization: Status and promise*, Washington
42. Jiao, J., & Tseng, M. M. (2000). Understanding product family for mass customization by developing commonality indices. *Journal of Engineering Design, 11*(3), 225–243.
43. Kota, S., Sethuraman, K., & Miller, R. (2000). A metric for evaluating design commonality in product families. *ASME Journal of Mechanical Design, 122*(4), 403–410.

44. Siddique, Z., Rosen, D.W., & Wang, N. (1998). On the applicability of product variety design concepts to automotive platform commonality, *Design Theory and Methodology—DTM'98*, Atlanta, GA, ASME, Paper No. DETC98/DTM-5661.
45. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.
46. Kollat, J. B., & Reed, P. M. (2005). The value of online adaptive search: A performance comparison of NSGA-II, ε-NSGAII, and εMOEA. In C. C. Coello, A. H. Aguirre, & E. Zitzler (Eds.), *The Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science 3410* (pp. 386–398). Guanajuato, Mexico: Springer.
47. Kollat, J. B., & Reed, P. M. (2006). Comparing state-of-the-art evolutionary multiobjective algorithms for long-term groundwater monitoring design. *Advances in Water Resources, 29*(6), 792–807.
48. Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation, 10*(3), 263–282.
49. Deb, K., Mohan, M., Mishra, S. (2003). A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. Tech. Rep. KanGAL 2003002, Indian Institute of Technology Kanpur.
50. Harik, G. R., Lobo, F. G. (1999). A parameter-less genetic algorithm. Tech. Rep. IlliGAL 99009, University of Illinois at Urbana-Champaign.
51. Tang, Y., Reed, P., Wagener, T. (2006). How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? Hydrology and earth system sciences 10 (2).
52. Kollat, J., & Reed, P. (2007). A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. *Advances in Water Resources, 30*(3), 408–419.
53. Goldberg, D. E. (2002). *The design of innovation: lessons from and for competent genetic algorithms*. Norwell, MA: Kluwer Academic Publishers.
54. Purshouse, R. C., & Fleming, P. J. (2007). On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation, 11*(6), 770–784.
55. Farina, M., & Amato, P. (2004). A fuzzy definition of "optimality" for manycriteria optimization problems. Systems, man and cybernetics, part A: Systems and humans, IEEE Transactions on, vol. 34, no. 3, pp. 315 – 326, May 2004.
56. Teytaud, O. (2006). How entropy-theorems can show that on-line approximating high-dim pareto-fronts is too hard. in PPSN BTP Workshop, 2006.
57. Laumanns, M., L. Thiele, K. Deb, & Zitzler, E. (2001), On the convergence and diversity-preservation properties of multi-objective evolutionary algorithms.
58. Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer, 1*(1), 69–91.