Facilitating Meta-design Techniques for Multi-disciplinary Conceptual Design

Alireza Maheri [*†], Askin T. Isikveren[‡]

Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, UK

Abstract

This paper represents a method of producing enhanced alternatives against an initial conceptual design candidate in multi-disciplinary design spaces. Working off the tenet of meta-design, or, the execution of design and development based on concurrent discovery and learning the produced alternatives are assessed at two stages. Primary assessment takes place during the generation process by a software tool developed for this purpose, while the designer carries out the refined assessment with the aid of the interpretation facilities provided by the software. The concept of "unsatisfactoriness" has been introduced and has been employed in assessment of design alternatives. The search method is a genetic algorithm capable of performing both local and global search. In a local modification the negative effect of modification on non-local zone is also taken into account. The developed software operates as a Design Decision Support System for conceptual design candidate evaluation and modification.

Keywords: Design Decision Support System, Educated Search, Conceptual Design, Optimisation, Genetic Algorithm, Evolutionary Computing

1 Introduction

It is well recognised that in conceptual design phase the impact of design decision is very high, while too little software is available to assist the designer [1]. Computer support of conceptual design is lagging behind, though it has significant scope for development [2]. Significant research has been carried out addressing issues such as search, modification and evaluation which are involved in development of design decision support systems for conceptual design phase [3-7]. However, design evaluation methods, which form the basis of most computer design support tools, provide poor support for multiple outcomes [7].

The principle of Meta-design is characterised by objectives, techniques and processes embodied in a Computer-Aided Engineering (CAE) environment that empowers designers to, where appropriate, consider development of innovative systems rather than just continue with the refinement of existing systems. This approach promotes the notion of "designing the design process" through thoughtful presentation of various figures-of-merit (or design quality) commensurate with the "tier" (level) of search the designer is opting to work in.

To achieve the ultimate goal of a modification process, one approach is to employ a refined evaluation function and robust search methods to find the optimum solution with respect to defined objectives. In multidisciplinary design problems, a design candidate is assessed based on a series of qualities which represent the goodness of various aspects of the design candidate. Forming a generalised and sophisticated evaluation function that comprehensively includes every aspects of the design is a difficult task. The challenge is not due to the number of design qualities involved in that function, but the form of the function itself. When constructing such a function, the appearance of vague and uncertain parameters such as weighting factors and tuning exponents is inevitable. These parameters refer to the level of importance of one design quality with respect to the other design qualities. The assigned values to these parameters are based mostly on the experience, knowledge and judgment of the one who defines the function, therefore their existence impacts the generality of the function.

In preference to the above approach, one may employ a relatively simpler, and hence, more generalised evaluation function to find a series of alternatives with enhanced qualities, which have passed a primary assessment stage. The designer can then utilise some interpretation facilities to study and compare the generated alternatives. This stage can be referred as the secondary or refined assessment stage. This approach is called "Educated Search" and is illustrated in Figure (1). Interpretation facilities, which are designed to enhance the designer judgment, comprise of visualisation tools for comparing the initial design candidate and its alternative solutions in the design space.

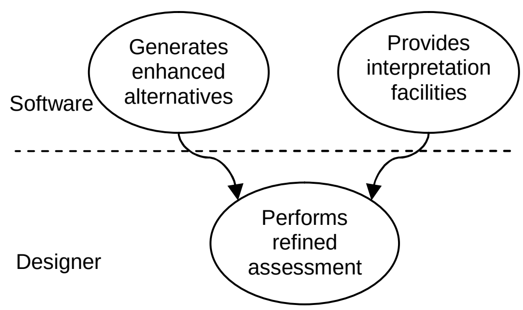

Figure 1 - Educated Search, a designer-in-loop design candidate modification

Serving as a platform that will facilitate meta-design capability in a future CAE design application this paper is mainly focused on the first stage of an Educated

---

[*] Corresponding Author, Alireza.Maheri@bristol.ac.uk
[†] Research Associate
[‡] Senior Lecturer in Engineering Design, Lead, Aerospace Vehicle Architecture & Design Integration (AVADI) Research Theme and Teaching Group

Search which is generation of enhanced alternatives and primary assessment.

2 Producing Enhanced Alternatives

Alternative generation follows a simple procedure. Starting with an initial design candidate, the software evaluates and modifies the design towards possessing enhanced qualities. Employing a global modification in a multi-disciplinary design problem might not be computationally efficient, if practical at all, whereas, a local modification approach, in which only some aspects of a design candidate in turn undergo modification, a smaller number of design variables are involved, leading to an easier and computationally more efficient modification being performed. A local modification, however, has a major drawback. When modifying a design candidate locally the designer needs to be aware of and control the effect of this local modification of design variables on design qualities at a macroscopic level. In this paper, the term “local zone” refers to one, or, a combination of two or more design subspaces while the rest of design space is referred to as “non-local” zone. Each design sub-space covers a subset of design qualities that are exclusive to that sub-space. However, design variables can be shared within different sub-spaces.

Normally, in a multi-disciplinary design problem, relations between design qualities and design variables have a wide range of complexities. Some design qualities are related to design variables through simple explicit functions, while some other cannot be determined unless complicated time consuming numerical methods are employed [8]. Empirical and low fidelity formulae governing the design domain, however, can be adequately suitable to be utilised in defining a low fidelity design space and generating approximated alternatives in conceptual design phase.

The initial stage of Educated Search is driven primarily by a simple iteration convergence algorithm. It is assumed that the design space can be modelled by relating the vector of design qualities, Y, to the vector of design variables, X, by utilising explicitly known and analytical functions, f_j s, as follows:

$$y_j = f_j(x_1, x_2, \dots, x_p) \quad j = 1, \dots, q \tag{1}$$

Where p and q are the number of design variables and design qualities in the local zone respectively. Applying a first order approximation to Equation (1), one obtains a set of linearised relations between Δx_i s and Δy_j s in a neighbourhood of the initial design candidate, X_0, as follows:

$$[\Delta y]_{q\times 1} = \left([C]_{q\times p}\right)_{X=X_0} [\Delta x]_{p\times 1} \tag{2}$$

Selection of any $m\times m$ sub-matrix from the matrix of coefficients $\left([C]_{q\times p}\right)_{X=X_0}$ and its corresponding Δx s and Δy s from $[\Delta x]_{p\times 1}$ and $[\Delta y]_{q\times 1}$ form a square system of linear equations:

$$[\Delta y]_{m\times 1} = \left([C]_{m\times m}\right)_{X=X_0} [\Delta x]_{m\times 1} \tag{3}$$

in which $m \in \{1, 2, ..., m_{\max}\}$ and $m_{\max} = \min\{p, q\}$ is the maximum number of available equations in the local zone. Replacing elements of $[\Delta y]_{m\times 1}$ by:

$$\Delta y_j = y_{j,m} - y_{j,0} \tag{4}$$

where $y_{j,m}$ is the (optimal) target value of design quality y_j, and solving for $[\Delta x]_{m\times 1}$ gives a new alternative that theoretically has m satisfactory qualities. Since a linearised approximated model has been used to obtain $[\Delta x]_{m\times 1}$, the m qualities are not exact target values, but will be very close to those values if $[\Delta x]_{m\times 1}$ is small enough. Moreover, if all of the y s involved in Equation (3) are already satisfactory this algorithm will reproduce the initial design candidate.

Having determined $[\Delta x]_{m\times 1}$, X_{new} can be calculated. Substitution of X_{new} back into functions f_j gives Y_{new} which needs to be assessed. If the overall quality of the generated alternative is satisfactory it will be passed into the set of feasible solution for the next stage of Educated Search, which is the refined assessment.

3 Primary Assessment of Generated Alternatives

An enhanced alternative has less “unsatisfactoriness” compared to an initial design candidate, and can be produced by applying a small modification to the initial design candidate (referred to as the “modification cost”). In the case of performing a local modification, an enhanced alternative has also small effects on the design qualities from non-local zone (referred to as the “redesigning cost”).

“Unsatisfactoriness” of a Design Candidate

Unsatisfactoriness of a design candidate can be measured by a combination of: (i) unsatisfactoriness of design qualities; (ii) number of unsatisfactory qualities; and (iii) maximum unsatisfactoriness of qualities.

To keep the objective of primary assessment simple the number of unsatisfactory design qualities and the maximum unsatisfactoriness of design qualities are not considered directly in the calculation for the unsatisfactoriness of the design candidate. However, these parameters will affect designer decision making when studying different alternatives at the refined assessment stage. Therefore, unsatisfactoriness of a design candidate, ε , is simply defined as the norm of the unsatisfactoriness of its qualities, namely,

$$\varepsilon = \sqrt{\sum \left(u_j\right)^2} \qquad (5)$$

$\forall$ unsatisfactory $y_j \in$ local zone.

where u_j represents the unsatisfactoriness of a typical design quality, y_j is defined by using its satisfactory interval $[y_{j,SL} \quad y_{j,SU}]$ for lower (*SL*) and upper (*SU*) bounds for which a target optimal value $y_{j,m} \in (y_{j,SL} \quad y_{j,SU})$ based on m equations is defined. In quantifying the extent of unsatisfactoriness the following heuristic applies:

$$u_j = \frac{y_j - y_{j,SU}}{y_{j,SU} - y_{j,m}} \qquad \text{if } y_j > y_{j,m} \qquad (5.a)$$

$$u_j = \frac{y_j - y_{j,SL}}{y_{j,SL} - y_{j,m}} \qquad \text{if } y_j < y_{j,m} \qquad (5.b)$$

Unsatisfactoriness defined by Equations (5) lies between -1 and $+\infty$. The condition $u = -1$ represents a fully satisfactory quality.

Redesigning Cost

Design variables are not necessarily exclusive to a local zone; therefore if their modification affects design qualities from other sub-spaces, the already designed and satisfactory sub-spaces must go through a redesigning process. One can reasonably assume that redesigning cost is proportional to the level of unsatisfactoriness produced in the non-local zone as a result of a local modification process. Let the Redesigning Cost, δ , be defined as:

$$\delta = \sqrt{\sum \left(u_j\right)^2} \ , \qquad (6)$$

$\forall$ already satisfactory $y_j \in$ non-local zone.

Modification Cost

Modification Cost represents how close the generated alternative is to the initial design candidate. A major modification in a design candidate due to significant changes in design variables may cause problems that affect the efficiency of the search process. For example the same method of analysis in assessment may no longer be valid if new design variable $x_{i,new}$ is too far from its initial value $x_{i,0}$. Modification Cost is about the performance of the modification process while the other two objectives concern the quality of the produced alternative. Let the Modification Cost, γ , be defined as:

$$\gamma = \sqrt{\sum \left(\Delta x_i\right)^2} \qquad (7)$$

$$\Delta x_i = x_{i,new} - x_{i,0} \qquad (7.a)$$

$\forall x_i$ selected for modification $\in$ local zone.

Evaluation Function

The level of local unsatisfactoriness, ε , and the associated Redesigning Cost of non-local zone, δ , both denote a deviation of design qualities from their satisfactory values, and hence, are considered equally important in the evaluation function. On the other hand since Modification Cost, γ , is not as important as the other two objectives, then it must be either weighted in the evaluation function, or eliminated from the evaluation function by the virtue of adding some constraints on x_is $\in$ local zone. Taking the latter option to avoid using a weighting system, let the evaluation function be defined as:

$$h = \sqrt{\delta^2 + \varepsilon^2} \qquad (8)$$

A generated alternative is declared as being feasible if $h_{new} < h_0$ and $x_{i,CL} \le x_i \le x_{i,CU}$. The parameter h_0 is the initial design objective, where $x_{i,CL}$ and $x_{i,CU}$ are the lower and upper limits of constraints imposed on the x_is due to elimination of the modification cost operation from the evaluation function.

4 Search Algorithm

In contrast to typical search and optimisation problems where the individuals with maximum fitness are chosen, here, at the end of a search process the designer studies and compares alternatives which have greatest fitness. Therefore, both the maximum fitness and the fitness distribution over the population are important. Moreover, since it is only some alternatives that the designer studies, rather than the entire population, manipulating large population sizes, if not influencing the quality of the search, is not judicious. Ideally, on one hand, population size should not be too large making the search process computationally expensive. On the other hand, a sufficient number of individuals that possess a high level of fitness should be present within the population in order to give the designer enough options to study and select from. The following sections elaborate on how, in the proposed genetic search algorithm definition of chromosomes, generation of initial population, utilisation of a variable population size and reproduction operations can be orientated in order to produce a set of enhanced alternatives efficiently.

Utilising the set of expressions found in Equation (3) as the basis of alternative generation, chromosomes (C_i) in the genetic algorithm can be defined by:

$$C_i = [\varphi_i \quad \psi_i] \tag{9}$$

in which φ_i is a vector of m design variables and ψ_i is a vector of m design qualities. Chromosomes have a variable length of $2m$, ($m \in \{1,2,...,m_{\max}\}$). To increase the efficiency of the algorithm, genetic operations are applied on only the φ-part of each chromosome, while the ψ-part is formed heuristically. That is for a φ-part of a chromosome with m elements, the first m elements of a prescribed vector of y s, called y_{order}, is selected to form the ψ-part of the chromosome. One y_{order} used in this study, is a sorted vector of local y s in which the first element is the most unsatisfactory local y, with the highest u obtainable from Equations (5.a) and (5.b). Figure (2) shows how forming ψ-part of the chromosomes heuristically improves the efficiency of search algorithm in finding alternatives with greatest fitness. All the results shown in the following Figure (2) refer to the average values of five runs of the algorithm for a generic design space summarised in the appendix.

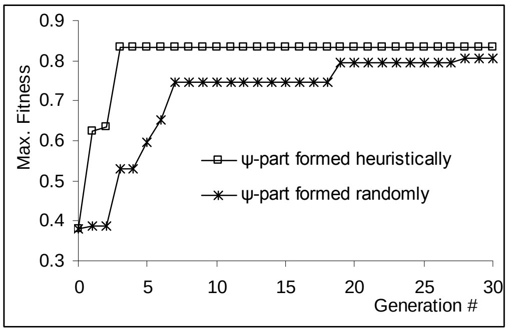

Figure 2 - Improving search efficiency by forming the ψ-part of the chromosomes heuristically

The level of fitness is defined by Equation (10),

$$fitness = \frac{1 - h/h_0}{1 + h/h_0} \tag{10}$$

where h and h_0 are objective and initial objective respectively. The fitness defined above is bounded between 0 and 1. The condition $fitness = 0$ represents the initial design candidate, while the condition $fitness = 1$ represents the mathematically best solution, with zero deviation of local y s from their satisfactory values ($\varepsilon = 0$) and zero Redesigning Cost ($\delta = 0$).

In another approach, y_{order} is a sorted vector of y s, where the first element is the most sensitive y to the variations of x s in the φ-part. Sensitivity of a design quality y_j to a φ-part can be represented by S_j, which is defined as the absolute variation in y_j for a unit variation of all design variables in that φ-part. Algebraically, this is given as

$$S_j = \left(\sum \left| \frac{\partial y_j}{\partial x_i} \right| \right)_{X = X_0} \quad \forall x_i \in \varphi\text{-part} \tag{11}$$

It can be observed that y_{order} could be generated based on most unsatisfactory y s first where $y_{order,u}$ is fixed for all possible φ-part. Alternatively, y_{order} based on most sensitive y s first, namely, $y_{order,s}$, depends on the present x s in the φ-part. Therefore, it is expected that the algorithm generates more alternatives when using $y_{order,s}$. Figure (3) compares the number of generated feasible alternatives using different definitions that constitute a y_{order} vector.

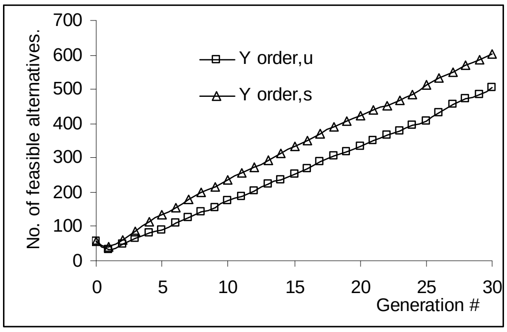

Figure 3 - Effect of implementing different y_{order} vectors in the algorithm on the number of generated feasible alternatives

Using $y_{order,u}$ in the algorithm is more likely to produce alternatives with smaller deviations in local y s from their satisfactory values, consequently smaller unsatisfactoriness while using $y_{order,s}$ is more likely to produce alternatives with lower Modification and Redesigning Costs. Figure (4) compares two ordering methods regarding the average (*av*) and minimum (*min*) unsatisfactoriness in local y s and redesigning cost of generated alternatives (ε_{av}, $\varepsilon_{\min}$ and δ_{av}).

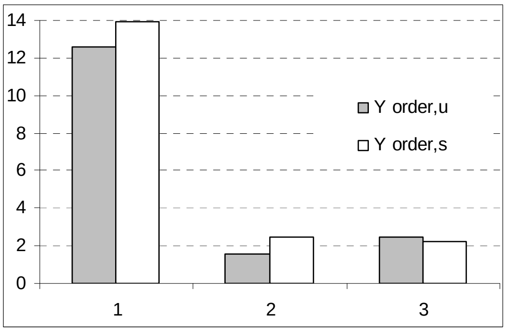

Figure 4 - Effect of implementing different y_{order} vectors in the algorithm on ε_{av} (1), $\varepsilon_{\min}$ (2) and δ_{av} (3)

Two sets of individuals form the initial population. In order to have all design variables in the initial population, the first set of individuals is made of all possible chromosomes with their φ-part in relation to the $m=1$ element. The second set of chromosomes in the initial population are made from some x-combinations with $m = m_{\max}$ in their φ-parts. Adding this set of the most populated chromosomes to the initial population accelerates the algorithm convergence by performing more effective cross-over operations in early generations.

Reproduction operations (cross-over and mutation) generate both feasible and unfeasible solutions while only feasible solutions are passed to the next generation. Adding the new feasible solutions obtained from the reproduction operators to the already existed feasible solutions from the current generation creates a new generation with a larger population size.

Mask vectors have been used for both cross-over and mutation operations. In cross-over, the number and locations of the genes in the mask vector are random numbers independent of generation number, population size and fitness of the selected individual. In mutation the fitness of an individual affects the number of genes in the mask vector. This is aimed at having refinement effect on individuals with higher fitness and significant mutation effects on individuals with lower fitness. In mutation the number of genes, n_g, in the mask vector is defined as:

$$n_g = \max\left\{1, \min\left\{[rp], \left[r\frac{h}{h_0}p\right]\right\}\right\} \qquad (12)$$

where $0 < r < 1$ is a random number and the bracket notation is used to denote the integer part function.

Cross-over and mutation probabilities, P_c and P_m respectively, are dynamic. As the population size increases, the value of P_c and P_m decrease in order to have a reasonable amount of cross-over and mutation operations in each generation. Cross-over and mutation probabilities are defined as:

$$P_c = P_{c0} \times \min\left\{1\,,\ \frac{N_0}{Popsize}\right\} \qquad (13)$$

$$P_m = P_{m0} \times \min\left\{1\,,\ \frac{N_0}{Popsize}\right\} \qquad (14)$$

in which P_{c0}, P_{m0} and N_0 are the reference cross-over probability, mutation probability, and population size.

Selection of parents for cross-over is based on a hybrid method. In early generations, when the population size is relatively small, each individual has the same chance of being selected, while in later generations with larger population the chance of selection of a chromosome is proportional to its fitness. Using a proportional selection approach from the beginning leads to smaller population sizes and possibly a premature convergence, while using a uniform selection generates large population sizes which do not necessarily contain fitter alternatives. Employing a hybrid approach is aimed at allowing the algorithm to generate more alternatives at the early stages and to prevent over-sizing the population thereafter. Figures (5) and (6) compare three parent selection methods regarding the number of generated feasible alternatives and fitness distribution over the first few alternatives.

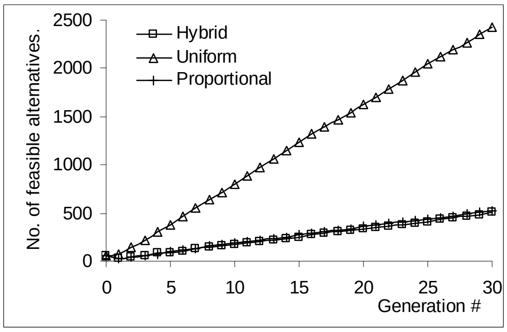

Figure 5 - Population size

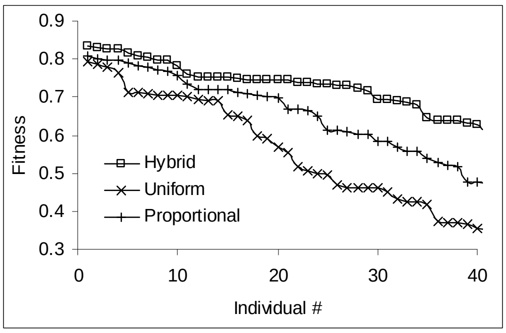

Figure 6 - Fitness distribution over the first 40 design alternatives after 30 generations

5 Search Performance

Provided they exist a fully successful local search finds design alternatives with $h = 0$ ($fitness = 1$). In the event of not having a fully successful search, enlarging or re-shaping of the local zone as well as re-modifying a generated alternative can be used as the two possible means of finding fitter alternatives in a second run of the search.

Figure (7) is an exemplar visualisation means for detailed study of alternatives at the stage of secondary assessment. It shows the unsatisfactoriness distribution over all design qualities of the initial design candidate, and two generated alternatives which are selected by the designer to study. Since the number of unsatisfactory design qualities and the maximum unsatisfactoriness of design qualities are not considered in the primary assessment, the best alternative according to the primary assessment criterion (the one with minimum h), might not pass the refined assessment stage.

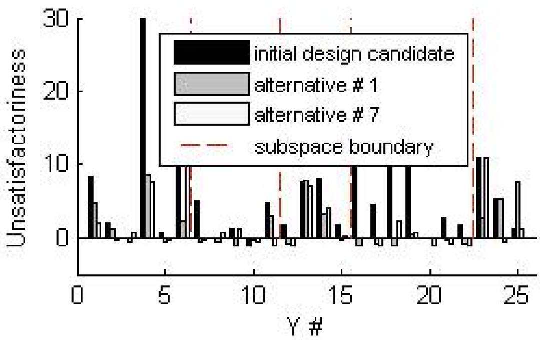

Figure 7 - Comparing design alternatives with respect to their qualities' unsatisfactoriness

6 Conclusion

To perform an effective and a meta-design based design candidate modification a designer-in-loop approach has been adapted. The robustness of the method is due to localisation of the search area, performing a directed search towards enhancing design qualities, performing a two stage assessment aimed at increasing the reliability of design candidate evaluation and exploiting the designer's knowledge, and efficiency of the search algorithm. The concept of unsatisfactoriness introduced in this paper has been employed to carry out a reliable assessment of design alternatives.

7 Acknowledgement

The research reported in this paper was supported by the EU FP6 funded project, SimSAC (Simulating Aircraft Stability and Control Characteristics for Use in Conceptual Design).

8 References

[1] L Wang, W Shen, H Xie, J Neelamkavil, A. Pardasani, "Collaborative conceptual design—state of the art and future trends", Computer Aided Design, 34 (13), 2002, pp 981-996

[2] W Hsu and B Liu, "Conceptual design: issues and challenges", Computer Aided Design, 32(14), 2000, pp 849-850

[3] D R Wallace, M J Jakiela, W C Flowers, "Design search under probabilistic specifications using genetic algorithms", Computer-Aided Design, 28(5), 1996, pp 405-421

[4] A Gelsey, M Schwabacher, D Smith, "Using modeling knowledge to guide design space search", Artificial Intelligence, 101(1-2), 1998, pp 35-62

[5] I Parmee, D Cvetkovic, C Bonham, I Packham, "Introducing prototype interactive evolutionary systems for ill-defined, multi-objective design environments", Advances in Engineering Software, 32(6), 2001, pp 429-441

[6] S Gentil, J Montmain, "Hierarchical representation of complex systems for supporting human decision making" Advanced Engineering Informatics, 18(3), 2004, pp 143-159

[7] PC Matthews, "A Bayesian support tool for morphological design", Advanced Engineering Informatics, 2007, doi:10.1016/j.aei.2007.05.001

[8] A Chattopadhyay, N Pagaldipti, "A multidisciplinary optimization using semi-analytical sensitivity analysis procedure and multilevel decomposition", Computers & Mathematics with Applications, 29(7), 1995, pp55-66

9 Appendix

The design space used to study the performance of the search algorithm includes 25 design qualities and 40 design variables distributed over five highly non-linear design subspaces. All design subspaces have common design variables with other subspaces. The table below summarises the design space and the initial design candidate.

Subspace i	No. of design variables p_i	No. of design qualities q_i	No. of unsat. qualities n_u	Max quality unsat. $\max\{u_j\}$
1	16	6	5	36.5
2	11	5	3	5.1
3	14	4	4	8.0
4	24	7	6	14.3
5	11	3	3	10.8

Table 1 - Design space properties

The local zone used in producing Figures (2) to (7) is the union of subspaces 2 and 4 and includes 12 design qualities, of which 9 initially unsatisfactory, and 29 local design variables.