9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization
4-6 September 2002, Atlanta, Georgia

AIAA 2002-5524

Optimization Methodologies In Conceptual Design

Carren M.E. Holden
BAE SYSTEMS,
ATC Sowerby Building,
FPC 267, PO Box 5,
Filton, Bristol, BS34 7QW,
UNITED KINGDOM.
Carren.Holden@
baesystems.com

Rob Davies
Future Projects Office
Airbus U.K.
No. 7 D.O., Filton,
Bristol, BS99 7AR,
UNITED KINGDOM.
Rob.Davies@
baesystems.com

Andy J. Keane
Professor of
Computational Engineering
University of Southampton
Room 2009, Building 25,
Main Campus, Highfield,
Southampton, SO17 1BJ,
UNITED KINGDOM.
Andy.Keane@soton.ac.uk

Abstract

There are still considerable cost benefits to be derived from improving aircraft designs both by better searches in current design spaces and by improved design space definition. The fastest way to perform a constrained optimization is probably using a classical quadratic programming (QP) method. However, it has been found difficult to meet a large number of constraints in a future concepts design program which consists of a combination of engineering modules and using a gradient based QP method. It was thought that a genetic algorithm might ameliorate these problems. Additionally, it is of interest to compare the QP method's performance with another gradient based algorithm, with different constraint handling. This paper therefore, compares QP, simplex ($SIMPLEX$) and genetic algorithm (GA) search methods in a future concepts design environment. The aim is to compare both the quality of feasible solutions and the search for new feasible parts of the design space prior to running gradient search algorithms. Reasons for the method choices and details of the particular implementations used are given.

Four optimization methods (including a hypersurface fitting technique) were tested on two sample problems. These are a 4–dimensional and 34–dimensional conceptual design for a high capacity, long range transport aircraft, with 52 constraints in both cases. Objective function evaluation time is rapid for all the test cases used.

As preliminary visualization showed that the design space was not fundamentally multi–modal, the primary reason for resorting to use of the genetic algorithm was to overcome non–linearities in the data, the superposition of which resulted in small local optima. Therefore a radial basis function (RBF) hypersurface fitting algorithm was applied to each of the constraints and objective function and the $SIMPLEX$ method was used to optimise the resulting design space. Although this worked well in 4–dimensions, getting the right combination of an accurate curve fit while still smoothing out the local minima was too difficult to achieve in 34–dimensions.

Comparison with the QP for known feasible and design of experiment (DoE) starts on a 34–dimensional problem shows that QP is quicker and achieves a better optimum when a feasible point is known. However, QP appears to have difficulty in obtaining a feasible point when starting outside the constraint boundaries. The hybrid GA demonstrates potential for tackling this problem, although a heavily constrained 34–dimensional space is a difficult problem, requiring a large number of evaluations. Such searches may still be the quickest way of generating a feasible design, however.

A deeper understanding of the relationship between the objective function, penalty function and design variables in the problem would enable better solving of the problem constraints. Extensive testing of different weightings within the penalty terms was not undertaken here. At present design space interior terms (i.e. terms involving satisfied constraints) have only been tested in the hybrid GA and were not utilized in the $SIMPLEX$ method. In the first instance it is thought that further experimentation should take place with the constraint weightings by modifying the scaling in the optimization problems and by modifying the form of the penalty functions used. Further experimentation with the curve fitting algorithms may also be productive.

1 Introduction

In a conceptual design, in which an iterative search is performed to find optimal aircraft layouts, there are a large number of optimization methods (or algorithms) that could be adopted. Roughly speaking these methods can be divided into two. Those which undertake a *local* search of the design space and methods that use a *global* search. Typically the local methods rely upon obtaining the gradient of the design space about a particular point and then iterating from this point using the gradient information to find the nearest local optima. These methods are limited. In particular:

- the methods are usually unable to escape local sub–optima in the design space.
- depending upon the nature of the problem, the result obtained from these methods may be very dependent on the start point of the optimization process (i.e. the initial design).

Global methods on the other hand are not as susceptible to these problems. These methods may be further divided into two classes.

- Those which make approximations to the design space, as, for instance, in the DACE kriging literature, see for example, Jones *et al* [7] and [8] and the Space mapping literature, for example Bandler *et al* [1] and combined approximation and optimization, see for example the work of Toropov *et al* [19, 20, 21].
- Those which probabilistically search the design space. These are the stochastic methods such as Simulated Annealing [12] and Genetic Algorithms [3, 24].

The methodologies to be employed in a future concepts design scenario need to robustly find optima in a highly constrained multi–dimensional design space. Recent work by Ong and Keane at Southampton University has considered the application of a number of optimization strategies [15, 16] from Keane's *OPTIONS* suite [9]. Both the Genetic Algorithm and Simplex method score highly in terms of robustness and neither use gradients *per se*, although the Simplex method is a local search. Cases for which the evaluation calculation fails before completion can have a poor* value set for the *fitness*, and either of these methods is robust to encountering such a point.

*Poor here refers to a low value for a maximizing or hill climbing optimization method and a high value for a minimizing or downhill optimization strategy.

2 Methods Utilized

Four optimization procedures were used:

- The simplex algorithm (*SIMPLEX*) used was the minimizing method of Nelder and Mead [14] and [23] with restarts in conjunction with decreasing penalty function relaxation parameter.

Var. No.	Prob. 1	Prob. 2	Starting value	Lower bound	Upper bound
1	√	√	0.6158320×10^6	0.5200000×10^6	0.7000000×10^6
2	√	√	0.8535750×10^3	0.7500000×10^3	0.1000000×10^4
3	×	√	7.718800	7.000000	0.1100000×10^2
4	×	√	0.6420070	0.4000000	0.8000000
5	×	√	0.3094997	0.2000000	0.5000000
6	×	√	0.1505300	0.1100000	0.1600000
7	×	√	0.1022190	0.9000000×10^{-1}	0.1200000
8	×	√	0.9390000×10^{-1}	0.9000000×10^{-1}	0.1200000
9	×	√	0.3850677×10^2	0.2500000	0.4500000×10^2
10	×	√	0.3850677×10^2	0.2500000	0.4500000×10^2
11	×	√	5.525351	-0.1000000×10^2	0.1500000×10^2
12	×	√	0.1634801	0.0000000	0.2000000
13	×	√	0.2577800	0.2000000	0.4000000
14	×	√	0.4130500	0.2000000	0.4500000
15	×	√	0.3299698	0.1000000	0.3500000
16	×	×	0.7650000	0.6500000	0.8500001
17	×	×	0.7500000	0.4000000	1.000000
18	×	√	0.2300000×10^2	0.0000000	0.2300000×10^2
19	×	√	0.3200000×10^2	0.0000000	0.3200000×10^2
20	×	×	1.000000	0.0000000	1.000000
21	×	√	4.017500	1.000000	6.000000
22	×	√	0.2542000	0.2500000	0.5000000
23	×	√	0.1204770×10^3	0.3000000×10^2	0.3000000×10^3
24	×	×	3.901000	0.0000000	5.000000
25	×	√	0.1986480×10^3	0.6000000×10^2	0.3000000×10^3
26	×	×	2.832000	0.0000000	5.000000
27	×	×	0.0000000	−.1000000	1.000000
28	×	×	0.0000000	−.1000000	1.000000
29	×	×	0.1852000×10^3	0.9000000×10^2	0.2000000×10^3
30	×	×	0.8900000	0.3000000	0.9000000
31	×	×	0.8397100×10^4	0.5000000×10^4	0.1000000×10^5
32	×	×	0.8500000	0.7200000	0.8500000
33	×	√	0.3716750×10^6	0.2500000×10^6	0.5000000×10^6
34	×	×	0.1371600×10^5	0.0000000	0.1400000×10^5
35	×	×	0.1543333×10^3	0.5000000×10^2	0.2000000×10^3
36	×	×	0.8400000	0.2000000	0.8700000
37	×	×	0.1543333×10^3	0.5000000×10^2	0.2000000×10^3
38	×	×	0.8500000	0.2000000	0.8700000
39	√	√	0.3850400×10^6	0.3200000×10^6	0.4200000×10^6
40	×	×	0.1066800×10^5	0.1000000×10^4	0.1200000×10^5
41	×	×	0.8500000	0.8000000	0.9000000
42	×	√	0.5752550	0.5000000×10^{-1}	1.000000
43	×	×	0.1188720×10^5	0.9450000×10^4	0.1200000×10^5
44	×	×	0.8500000	0.8000000	0.9000000
45	×	×	0.9496400	0.8000000	0.9500000
46	×	×	0.1188720×10^5	0.9450000×10^4	0.1400000×10^5
47	×	×	0.8500000	0.8000000	0.9000000
48	√	√	0.5290600×10^6	0.4000000×10^6	0.6000000×10^6
49	×	√	0.6524800	0.1000000	0.9000000
50	×	×	0.8980800	0.1000000	0.9000000
51	×	√	0.3915600×10^6	0.3000000×10^6	0.4500000×10^6
52	×	√	0.2300000×10^2	0.0000000	0.2300000×10^2
53	×	√	0.2007360×10^2	0.0000000	0.3200000×10^2
54	×	√	0.2300000×10^2	0.0000000	0.2300000×10^2
55	×	√	0.2535807×10^2	0.0000000	0.3200000×10^2
56	×	√	0.1220400×10^3	0.0000000	0.3000000×10^3
57	×	√	0.1205400×10^3	0.0000000	0.2000000×10^3
58	×	×	0.6096000×10^4	0.8000000×10^2	0.7000000×10^4
59	×	√	0.1830030×10^3	0.8000000×10^2	0.2500000×10^3

Table 1: Starting values and bounds for *Problems 1 & 2*. Variable scaling was by reference to the upper and lower design variable bounds. A tick (√) indicates the design variables in the specified problem.

- A classical quadratic programming (*QP*) method, which utilizes both *Gauss–Newton* and *quasi–Newton* optimization, the latter for the case when the optimization is at a point distant from the feasible set ([18]).

- A Genetic Algorithm (GA). The features of this approach are detailed in Appendix B and its performance was tested on a well understood analytical problem. In general it performs as expected and well, although (of course) is not guaranteed to find the global optimum. As the number of dimensions increases the harder it becomes to find the global optimum with *any* optimization method.
- A linear radial basis function (RBF) (see for example [8]) was used to fit a design of experiments (DoE) of the objective function and constraints separately. Then the $SIMPLEX$ method was used to optimize the model. A final evaluation of the objective function and constraints is then required to be sure that the curve fit is accurate in the region of the optimum. If the result is inaccurate a new model may be created including the previously found optimum point.

3 The Conceptual Design Problem

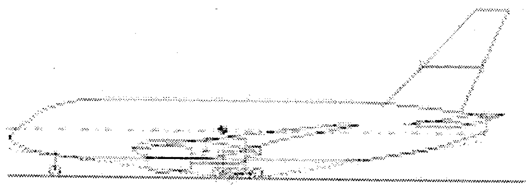

Figure 1: *Side view of a typical aircraft design.*

The following test cases were utilized:

1. A conceptual design problem in 4 dimensions.
2. A conceptual design problem in 34 dimensions.

The measure of efficiency utilized is the number of objective function and constraint evaluations as the cost of the rest of the optimization is relatively small compared to these.

The conceptual design of a high capacity aircraft considered here uses up to a total of 59 possible design variables. Many conceptual design optimizations are difficult, non–linear, optimization problems because the large number of constraints in the problem give a heavily constrained design space, with small pockets of feasibility. The optimizations were performed using QP, the $SIMPLEX$ method alone and a hybrid *SIMPLEX/GA* on subsets of these. The design variables used for both the 4 and 34 dimensional problems were as shown in Table 1. The problem has 52 active constraints in all cases, including equalities and inequalities. From a large number of possible alternatives, the objective function chosen for these test cases was direct operating cost (DOC). The objective function calculation is invoked in the conceptual design program by calling the subroutine *design*. A sketch of a side view of a typical aircraft design is shown in Figure 1.

In all cases the hybrid GA was used with $SIMPLEX$ evaluations at the first, 100^{th}, 200^{th} and 400^{th} evaluations, see Appendix B. There are 200 members in the population at each generation unless otherwise stated. Where the number of members in the population is reduced, the convergence criteria of the $SIMPLEX$ method is increased so that fewer evaluations take place when running the $SIMPLEX$ method part of the hybrid GA.

3.1 The conceptual design problem in four dimensions. (*Test problem 1.*)

Prob. No.	Var. No.	Starting value	Final value	Movement
1a	1	615832.0000	615485.4375	- 0.1 %
1a	2	853.5750	852.8229	- 0.1 %
1a	39	385040.0000	384897.9375	- 0.0 %
1a	48	529060.0000	528791.6875	- 0.1 %
1b	1	615832.0000	615804.9375	- 0.0 %
1b	2	853.5750	852.7921	- 0.1 %
1b	39	385040.0000	384980.3438	- 0.0 %
1b	48	529060.0000	529049.9375	- 0.0 %
1c	1	615832.0000	615483.5625	- 0.1 %
1c	2	853.5750	852.7855	- 0.1 %
1c	39	385040.0000	384895.1562	- 0.0 %
1c	48	529060.0000	528789.8125	- 0.1 %
1d	1	615832.0000	615434.5000	- 0.1 %
1d	2	853.5750	853.1151	- 0.1 %
1d	39	385040.0000	384919.8750	- 0.0 %
1d	48	529060.0000	528845.0625	- 0.0 %

Table 2: *Test result 1a:* GA Results after 104657 calls to *design* from a DoE of 200 points *including* a specified start, which meets all 52 constraints. *Test result 1b:* QP Results after 54 calls to *design* from the same specified start, which meets all 52 constraints. *Test result 1c:* GA Results after 114078 calls to *design* starting from a DoE of 200 points. Here, the starting point shown is for comparison purposes and was *not* used to obtain the solution. *Test result 1d: RBF & QP* Results after 10 calls to *design* from the specified start, which meets all 52 constraints.

Four results were obtained for the 4 design variable problem. The first, *result 1a*, from the hybrid GA

with a start from a *DoE* of 199 points[†] and the point which satisfies the constraints given in Table 1. The second, *result 1b*, from *QP* with a start from the point which satisfies the constraints given in Table 1. The third, *result 1c*, from the hybrid *GA* from a *DoE* of 200 points *excluding* the point which satisfies the constraints. The fourth result was obtained by fitting the objective function and constraints with an *RBF* and using the *SIMPLEX* method to optimise the derived design space. The final values of the design variables and constraints for *problem 1* are given in Tables 2 and 3 respectively and a comparison between the achieved values of the objective function, *DOC* is given in Table 4.

Constraint No. & Type	Constraint value -1a	Constraint value -1b	Constraint value -1c	Constraint value -1d
Initial 1#	0.3360	0.3359	0.3359	0.3362
Initial 4#	1.3604	1.3604	1.3604	1.3608
Initial 6#	1.4072	1.4072	1.4072	1.4073
Initial 7#	0.2676	0.2676	0.2676	0.2676
Initial 9#	0.1946	0.1946	0.1946	0.1947
Initial 20=	0.0000	0.0000	0.0000	0.0000
Initial 27#	0.0068	0.0068	0.0068	0.0068
Initial 29#	4.5762	4.5762	4.5762	4.5762
Initial 32#	-38.4687	0.0000	-38.4062	-28.3437
Initial 33#	0.0006	0.0006	0.0006	0.0006
Initial 34#	0.0984	0.0984	0.0984	0.0984
Initial 36#	0.0077	0.0077	0.0077	0.0077
Commonality 45#	0.0549	0.0549	0.0549	0.0550
Commonality 46#	0.6876	0.6875	0.6875	0.6879
Longit. S&C 64#	0.0000	0.0000	0.0000	0.0002
Longit. S&C 67#	0.2077	0.2077	0.2077	0.2078
Longit. S&C 71#	0.0300	0.0300	0.0300	0.0302
Longit. S&C 73#	0.1934	0.1935	0.1935	0.1932
Longit. S&C 74#	0.2077	0.2077	0.2077	0.2078
Longit. S&C 76#	0.0008	0.0008	0.0008	0.0007
Longit. S&C 78#	0.0671	0.0671	0.0670	0.0673
Longit. S&C 79#	0.0164	0.0163	0.0164	0.0162
Longit. S&C 80#	0.1934	0.1935	0.1935	0.1932
Longit. S&C 81#	0.2077	0.2077	0.2077	0.2078
Longit. S&C 82#	0.0427	0.0427	0.0427	0.0427
Longit. S&C 84#	0.0929	0.0929	0.0930	0.0927
Longit. S&C 85#	0.0591	0.0591	0.0590	0.0593
Longit. S&C 87#	0.2680	0.2680	0.2680	0.2678
Longit. S&C 88#	0.2827	0.2827	0.2827	0.2828
Longit. S&C 89#	0.0306	0.0306	0.0306	0.0306
Longit. S&C 90#	0.0088	0.0308	0.0308	0.0307
Lateral S & C 94#	0.2141	0.2218	0.2234	0.1418
Engine Out 112#	0.0165	0.0165	0.0165	0.0165
Takeoff 113#	0.0001	0.0000	0.0001	0.0000
Takeoff 114#	210.0	206.9	209.9	210.7
Takeoff 116=	-61.2	0.0	-61.2	-165.6
Climb 118#	4.234	4.209	4.234	4.231
Climb 119#	1.1604	1.1511	1.1602	1.1604
Climb 127#	2.0558	2.0485	2.0558	2.0548
Cruise 121#	0.7708	0.7613	0.7706	0.7714
Cruise 129#	2.0955	2.0883	2.0954	2.0947
Hold 150#	0.01	0.01	0.01	0.01
Overall Mission 151#	10820.2	10679.5	10798.6	10970.1
Overall Mission 152#	-51.4	0.0	-52.4	-24.0
Climb 159#	6.2746	6.2746	6.2746	6.2747
Cruise 161#	5.6330	5.6330	5.6330	4.4326
Climb 167#	4.4325	4.4324	4.4324	0.2195
Cruise 169#	4.5556	4.5556	4.5555	4.5559
Climb 176#	0.2193	0.2193	0.2193	5.63331
Cruise 190#	0.00	0.00	0.00	0.00
Overall Mission 192#	148.5	66.1	151.3	126.9
Low speed Perf. 193#	0.0293	0.0203	0.0280	0.0396

Table 3: Results for *Test Problem 1.* # - inequality constraint, = - equality constraint, S&C - stability and control, *Longit.* – *Longitudinal* and *Perf.* – *Performance.*

The two results from the *GA* were very similar, both in terms of modification to design variables, values of the constraints and the final value of *DOC* obtained. This indicates that the *GA* is searching well in a design space for which prior knowledge is unavailable. *QP* converged, after just 54 evaluations having a smaller reduction in *DOC*. In fact the reduction in *DOC* was small for all three cases, indicating that the design space is tightly confined by the constraints in the problem and that the initial design was of high quality.

[†]See McKay [11] for a description of the DoE used.

Figure 2 shows a contour plot of the sum of the constraint functions for design variables 1 and 39 for a reduced portion of the design space. Although the individual constraint functions are linear with some noise, the space is complicated as there are 52 design variables in total. The plane of symmetry in the plot is caused by the equality constraints in the problem. Figure 3 shows a Hierarchical Axes Technique (HAT) plot [13] of a reduced portion of the 4–dimensional design space. Viewing this domain more closely and in just one dimension shows that local minima can be created due to superposition of constraints when slight deviations from the linear occur in some individual constraint evaluations. The problem with these is that they generally stop the gradient search method before it can reach the true optimum.

RBF's can be used to perform regression on the conceptual design space to try to eliminate these local minima. However, a balance is required between smoothing the data and not being so far away from the original data that the design point located no longer satisfies the constraints. This is easily possible in four dimensional space, as found in *result 1d* shown in Tables 2 to 4, but results satisfying all the constraints were not achieved in higher dimensions.

Start	Result	no. of evals	Start	Final	Movement
specified (GA)	1a	104657	119784.22	119768.36	-0.013%
specified (QP)	1b	54	119784.22	119779.52	-0.004%
from DoE (GA)	1c	114078	-	119768.12	-0.013%
specified (RBF&QP)	1d	10	119784.22	119769.24	-0.013%

Table 4: *GA* and *QP* Results for *test problem 1.*

3.2 The conceptual design problem in 34–dimensions. (*Test problem 2.*)

Five sets of results are presented for this problem. The first (*result 2a*) is for a specified start, using the hybrid *GA*. These results are compared to those from *QP* (*result 2b*). Then a *DoE* is used to provide all the starting points for both the hybrid *GA* (*result 2c*) and *QP* (*result 2d*). Finally, a larger run of the hybrid *GA* using the *DoE* to provide all the starting points is performed (*result 2e*).

result 2a in which the starting point specified in Table 5 and a *DoE* of 99 points is optimized using the *GA* and gives just a 0.1% improvement in *DOC*

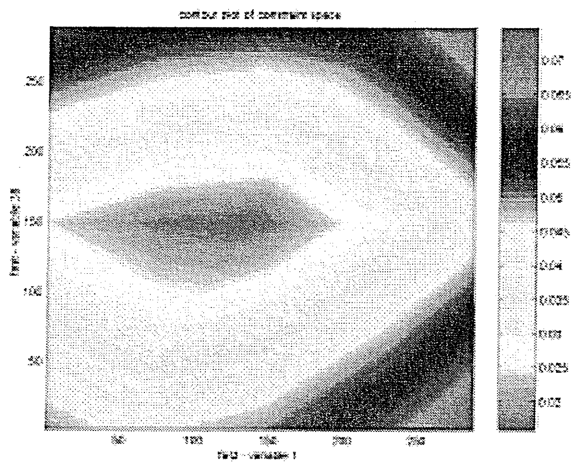

Figure 2: *Contour plot of two design variables in the Four Dimensional Design Space. Variables 2 and 48 are held fixed. (The numbers on the axes denote pixel number (289 in total)).*

No.	Scaling factor	Starting value	Lower bound	Upper bound
1	0.5000000×10^6	0.6158320×10^6	0.520000×10^6	0.700000×10^6
2	0.800000×10^3	0.8535750×10^3	0.750000×10^3	0.100000×10^4
3	0.100000×10^2	0.7718800×10	0.700000×10	0.110000×10^2
4	0.600000	0.6420070	0.400000	0.800000
5	0.300000	0.3094997	0.200000	0.500000
6	0.100000	0.1505300	0.110000	0.160000
7	0.100000	0.1022190	0.900000×10^{-1}	0.120000
8	0.100000	0.939000×10^{-1}	0.900000×10^{-1}	0.120000
9	0.300000×10^2	0.3850677×10^2	0.250000	0.450000×10^2
10	0.300000×10^2	0.3850677×10^2	0.250000	0.450000×10^2
11	5.00000	0.5525351×10	-0.100000×10^2	-0.150000×10^2
12	0.150000	0.1634801	0.000000	0.200000
13	0.300000	0.2577800	0.200000	0.400000
14	0.400000	0.4130500	0.200000	0.450000
15	0.200000	0.3299698	0.100000	0.350000
18	0.200000×10^2	0.230000×10^2	0.000000	0.230000×10^2
19	0.300000×10^2	0.320000×10^2	0.000000	0.320000×10^2
21	0.500000	0.4017500×10	0.100000×10	0.600000×10
22	0.300000	0.2542000	0.250000	0.500000
23	0.200000×10^3	0.1204770×10^3	0.300000×10^2	0.300000×10^3
25	0.200000×10^3	0.1986480×10^3	0.600000×10^2	0.300000×10^3
33	0.300000×10^6	0.3716750×10^6	0.250000×10^6	0.500000×10^6
39	0.200000×10^6	0.3850400×10^6	0.320000×10^6	0.420000×10^6
42	0.500000	0.5752550	0.500000×10^{-1}	0.100000×10
48	0.100000×10^6	0.5290600×10^6	0.400000×10^6	0.600000×10^6
49	0.100000	0.6524800	0.100000	0.900000
51	0.100000×10^6	0.3915600×10^6	0.300000×10^6	0.450000×10^6
52	0.200000×10^2	0.230000×10^2	0.000000	0.230000×10^2
53	0.200000×10^2	0.2007360×10^2	0.000000	0.320000×10^2
54	0.200000×10^2	0.230000×10^2	0.000000	0.230000×10^2
55	0.200000×10^2	0.2535807×10^2	0.000000	0.320000×10^2
56	0.200000×10^2	0.1220400×10^3	0.000000	0.300000×10^3
57	0.200000×10^2	0.1205400×10^3	0.000000	0.200000×10^3
59	0.100000×10^3	0.1830030×10^3	0.800000×10^2	0.250000×10^3

Table 5: Starting values, bounds and description for *results 2a and 2b*. Variable scaling by reference to the upper and lower design variable bounds for the hybrid *GA result 2a*. The scaling values shown were used for *QP result 2b*.

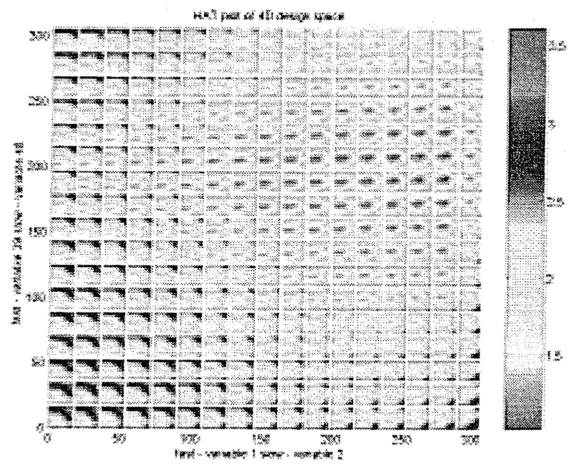

Figure 3: *HAT plot of the sum of the constraints in the Four Dimensional Design Space. The tile one in from the right and six tiles from the top is detailed in Figure 2. In an individual tile the slow variables (2 and 48) remain fixed. Going from tile to tile in the x–direction represents a change in design variable 2 and in the y–direction a change in design variable 48.*

from the specified start in 71747 evaluations, with a population size of 100 per generation. However, no constraints were violated in this result. This can be compared directly to *result 2b*, in which a run with *QP* having 39878 calls to *design* gives a 3.941% improvement in *DOC* from the specified start, again with no violations. Clearly the *GA* explored the wider design space to no effect while a concentrated search by *QP* produced better results.

In *result 2c*, which uses a Latin hypercube *DoE* and hybrid *GA* without a specified start, after 71039 evaluations the optimum still violates *constraint 32* significantly. The final *DOC* is 113811.23, which is 4.986% better than the specified start used in *results 2a and 2b*.

An alternative to this strategy, for comparison purposes is performed in *result 2d*. Here, a *DoE* with 200 points and *QP* was started from the best 5 points in the *DoE*. In all cases two or more constraints were violated. The *GA* software was easily adapted to perform the *DoE* and select the 5 best solutions ready for input to *QP*. This complete process uses a similar number of evaluations to running the hybrid *GA*. These results are compared in Table 6.

As a final comparison a larger *GA* run was under-

Res-ult	Start point	start DOC	final DOC	% imp-roved	no. of viols	violated constraints	no. of extra evals
2c	GA	-	113811.23	4.986%	1	32	71039
2d	1	117436.58	115288.80	1.829%	3	32, 94, 113	3157
2d	2	117440.81	111277.48	5.248%	4	27, 32, 94, 114	25520
2d	3	117445.05	113885.16	3.031%	4	27, 32, 64, 113	9907
2d	4	117449.00	111412.09	5.140%	9	27,32,64,76,94,113,114	20764
2d	5	117453.53	113245.39	3.583%	2	32,76	9121

Table 6: Test *result 2d* used a DoE with 200 points and started QP from the best 5 of these. The total number of evaluations for *result 2d* was 68669. *result 2c* is also shown, in which a DoE with 200 points was used to start a hybrid GA.

taken (*result 2e*). This used a starting population consisting of a DoE of 200 points. The results are given in Tables 7 and 8. This optimization used a total of 161094 evaluations in all. Only constraint 152 is outside the tolerance for this result and this has a scaled value of 2.0×10^{-3} compared to the value of 1.0×10^{-4} which was the set tolerance. It is likely that this value would be improved by changing the relative scaling of this parameter in the problem and tuning the other parameters. (Extensive testing of relative scalings and parameter settings lies outside the scope of this work.)

Variable No.	Starting value	Final value	Movement
1	615832.0000	599900.6250	−2.6%
2	853.5750	922.8808	+8.1%
3	7.7188	7.2042	−6.7%
4	0.6420	0.6105	−4.9%
5	0.3095	0.3821	+23.5%
6	0.1505	0.1347	−10.5%
7	0.1022	0.0910	−10.9%
8	0.0939	0.0909	−3.2%
9	38.5068	38.9042	+1.0%
10	38.5068	38.9002	+1.0%
11	5.5254	5.2000	−5.9%
12	0.1635	0.1719	+5.1%
13	0.2578	0.3046	+18.1%
14	0.4130	0.3455	−16.3%
15	0.3300	0.3409	+3.3%
18	23.0000	21.3569	−7.1%
19	32.0000	29.4349	−8.0%
21	4.0175	3.9786	−1.0%
22	0.2542	0.2530	−0.5%
23	120.4770	134.4109	+11.6%
25	198.6480	259.1774	+30.5%
33	371675.0000	350576.0312	−5.7%
39	385040.0000	386229.0938	+0.3%
42	0.5753	0.4474	−22.2%
48	529060.0000	513187.2812	−3.0%
49	0.6525	0.6899	+5.7%
51	391560.0000	403604.5000	+3.1%
52	23.0000	21.5397	−6.3%
53	20.0736	6.0470	−69.9%
54	23.0000	20.6249	−10.3%
55	25.3581	6.3736	−74.9%
56	122.0400	118.1193	−3.2%
57	120.5400	192.0125	+59.3%
59	183.0030	157.5098	−13.9%

Table 7: GA Results after 161094 calls to *design* from a DoE of 200 points, *excluding the specified start* (shown here for comparison purposes). Only design variables which are varied are shown in this table. *Test result 2c.*

It seems that the best application for the GA here is in the search for feasible points and not to optimize once in the vicinity of an optimum, a task that gradient methods, such as QP seem better suited for (they require fewer evaluations to reach a better optimum). However, QP has significant difficulty in finding solutions where all the constraints are satisfied; this is typical of gradient based optimizations in general. However, the number of evaluations using the GA to perform this task are significantly more than the designers have been using to date. Even so such searches may still be quicker than manual methods, in which solutions from QP are adjusted manually until the constraints are solved and then QP is used to improve the result. The known starting point that yields *result 2a* is such a case and the final answer from QP remains the best result seen for this problem. Without this information, QP alone is unable to yield a feasible solution.

4 Comparison between the Simplex and QP Methods

Constraint No. & Type	Constraint value
Initial 1#	0.1652
Initial 4#	0.5358
Initial 6#	1.3595
Initial 7#	0.2057
Initial 9#	0.1469
Initial 20=	-0.0040
Initial 27#	0.2412
Initial 29#	0.1366
Initial 32#	-38.4687
Initial 33#	0.0077
Initial 34#	0.0114
Initial 36#	0.0017
Commonality 45#	0.1342
Commonality 46#	1.0015
Longit. S&C 79#	0.0235
Longit. S&C 73# 80# 87#	0.2129 0.2129 0.2874
Longit. S&C 67# 74# 81# 88#	0.2009 0.2009 0.2009 0.2759
Longit. S&C 82# 89#	0.0334 0.0267
Longit. S&C 76# 83* 90#	0.0002 0.0082 0.0302
Longit. S&C 84#	0.0802
Longit. S&C 64# 71# 78# 85#	0.0004 0.0304 0.0798 0.0718
Longit. S&C 94#	0.1386
Engine Out 112#	0.0141
Takeoff 113#	0.0135
Takeoff 114#	1.9
Takeoff 116=	-41.6
Climb 118#	2.221
Climb 119#127#135*	1.0988 0.5535 -1.5240
Cruise 121#129#	1.3241 1.0323
Hold 150#	0.00
Overall Mission 151#	28515.4
Overall Mission 152#	-1240.7
Climb 159#167#	4.5315 3.0266
Cruise 161#169#	4.2343 3.3488
Climb 176#	0.2845
Hold 190#	0.45
Overall Mission 192#	5796.6
Low Speed Perf. 193#	1.1930

Objective	Start	Final	Movement
DOC	119784.22	118253.27	-1.278%

Table 8: GA Results after 161094 calls to *design*. # - inequality constraint, = - equality constraint, S&C - stability and control, *Longit.* – *Longitudinal* and *Perf.* – *Performance*. *Test result 2e.*

A test case was also chosen to compare the relative performance of $SIMPLEX$ and QP. The input for both the $SIMPLEX$ and QP methods in this test case is given in Table 9. In each case the input deck has been adjusted by the designers to best suit the individual optimization technique, as would be done ordinarily during their optimization procedure.

The final values of the design variables and constraints are given in Tables 10 and 12 for the methods and a comparison between the final values of DOC achieved are given in Table 13. These results show that the two methods probably give an equally good optimum, but one that is different in each case. However, the $SIMPLEX$ method requires 2.5 times the number of evaluations as QP and therefore takes 2.5 times as long to run. These numbers of evaluations may be decreased by judicious reduction of the number of evaluations in the individual $SIMPLEX$ method iterations and the number of $SIMPLEX$ iterations used.

It is interesting to note that none of the $SIMPLEX$ method resulting design variables sit on the boundary, whereas the QP results do. The $SIMPLEX$ method penalty function currently does not drive the optimizer to the boundary, whereas the QP algorithm solves the problem at the boundary (although it too does not contain penalty function terms for satisfied constraints).

Var. No.	Scaling factor	Starting value (QP)	Starting value (SIMPLEX)	Lower bound	Upper bound
1	0.5000×10^6	0.6148240×10^6	0.6158320×10^6	0.52×10^6	0.70×10^6
2	0.8000×10^3	0.8600500×10^3	0.8535750×10^3	0.75×10^3	0.10×10^4
3	0.1000×10^2	7.718800	7.718800	7.000	0.11×10^2
4	0.6000	0.6420070	0.6420070	0.4000	0.8000
5	0.3000	0.3094997	0.3094997	0.2000	0.5000
6	0.1000	0.1505300	0.1505300	0.11000	0.16000
7	0.1000	0.1022190	0.1022190	0.90×10^{-1}	0.12000
8	0.1000	0.939000×10^{-1}	0.939000×10^{-1}	0.90×10^{-1}	0.12000
9	0.3000×10^2	0.3850677×10^2	0.3850677×10^2	0.25000	0.45×10^2
10	0.3000×10^2	0.3850677×10^2	0.3850677×10^2	0.25000	0.45×10^2
11	5.000	5.525351	5.525351	-0.10×10^2	0.15×10^2
12	0.15000	0.1634801	0.1634801	0.0000	0.2000
13	0.3000	0.2577800	0.2577800	0.2000	0.4000
14	0.4000	0.4130500	0.4130500	0.2000	0.45000
15	0.2000	0.3299698	0.3299698	0.1000	0.35000
18	0.2000×10^2	0.23000×10^2	0.23000×10^2	0.0000	0.23×10^2
19	0.3000×10^2	0.32000×10^2	0.32000×10^2	0.0000	0.32×10^2
21	0.5000	4.017500	4.017500	1.000	6.000
22	0.3000	0.2527000	0.2542000	0.25000	0.5000
23	0.2000×10^3	0.1217730×10^3	0.1204770×10^3	0.30×10^2	0.30×10^3
25	0.2000×10^3	0.1992000×10^3	0.1986480×10^3	0.60×10^2	0.30×10^3
33	0.3000×10^6	0.3707500×10^6	0.3716750×10^6	0.25×10^6	0.50×10^6
39	0.2000×10^6	0.3852800×10^6	0.3850400×10^6	0.32×10^6	0.42×10^6
42	0.5000	0.6053700	0.5752550	0.50×10^{-1}	1.000
48	0.1000×10^6	0.5280600×10^6	0.5290600×10^6	0.40×10^6	0.60×10^6
49	0.1000	0.6524800	0.6524800	0.1000	0.9000
51	0.1000×10^6	0.3917250×10^6	0.3915600×10^6	0.30×10^6	0.45×10^6
52	0.2000×10^2	0.23000×10^2	0.23000×10^2	0.0000	0.23×10^2
53	0.2000×10^2	0.2007360×10^2	0.2007360×10^2	0.0000	0.32×10^2
54	0.2000×10^2	0.23000×10^2	0.23000×10^2	0.0000	0.23×10^2
55	0.2000×10^2	0.2535807×10^2	0.2535807×10^2	0.0000	0.32×10^2
56	0.2000×10^2	0.1220400×10^3	0.1220400×10^3	0.0000	0.30×10^3
57	0.2000×10^2	0.1205400×10^3	0.1205400×10^3	0.0000	0.20×10^3
59	0.1000×10^3	0.1830030×10^3	0.1830030×10^3	0.8000×10^2	0.25×10^3

Table 9: QP and $SIMPLEX$ starting values and bounds. Variable scaling utilized in the $SIMPLEX$ was determined by reference to the upper and lower design variable bounds and for the QP method by use of the values shown.

5 Conclusions

A comparison has been made between a quadratic programming method, a stand alone $SIMPLEX$ method and a hybrid $SIMPLEX/GA$ *with constraint handling capabilities and clustering.* The hybrid GA includes a Latin hypercube DoE to obtain an initial starting population rather than the random population that is used more usually. It is thought that this DoE gives better coverage through the design space. The hybrid GA also uses a $SIMPLEX$ method to converge to local peaks in the population at generations specified by the user.

The slight non–linearities in some of the constraints in the problem and interaction between these constraints causes the design space to be too complicated to be solved directly using gradient based methods without a starting solution which satisfies the constraints. The actual design space seems to be small as are the numbers of solutions obtained which satisfied all the constraints. It is clear that real problems contain more difficulties than may be imagined at the outset.

Var. No.	Starting value	Final value	Move-ment
1	615832.0	578306.5625	- 6.1 %
2	853.5750	838.6968	- 1.7 %
3	7.7188	7.5173	- 2.6 %
4	0.6420	0.6047	- 5.8 %
5	0.3095	0.3770	+21.8 %
6	0.1505	0.1447	- 3.9 %
7	0.1022	0.0981	- 4.1 %
8	0.0939	0.0991	+ 5.5 %
9	38.5068	38.8596	+ 0.9 %
10	38.5068	38.8533	+ 0.9 %
11	5.5254	5.4566	- 1.2 %
12	0.1635	0.1971	+20.5 %
13	0.2578	0.3209	+24.5 %
14	0.4130	0.3455	-16.4 %
15	0.3300	0.3483	+ 5.6 %
18	23.0000	22.8644	- 0.6 %
19	32.0000	20.6856	-35.4 %
21	4.0175	3.7738	- 6.1 %
22	0.2542	0.2609	+ 2.6 %
23	120.4770	117.7440	- 2.3 %
25	198.6480	207.8923	+ 4.7 %
33	371675.0	326765.8125	-12.1 %
39	385040.0	371247.1250	- 3.6 %
42	0.5753	0.5960	+ 3.6 %
48	529060.0	491671.0312	- 7.1 %
49	0.6525	0.8862	+35.8 %
51	391560.0	387539.2812	- 1.0 %
52	23.0000	22.9185	- 0.4 %
53	20.0736	9.3408	-53.5 %
54	23.0000	0.1569	-99.3 %
55	25.3581	2.7812	-89.0 %
56	122.04	119.6032	- 2.0 %
57	120.54	166.4054	+38.0 %
59	183.00	154.5911	-15.5 %

Table 10: $SIMPLEX$ method results after 5 iterations and 43436 calls to *design*.

Comparison with the gradient based method, QP, on the 4– dimensional conceptual design problem demonstrates that the hybrid GA can perform better than QP and that performance does not necessarily depend on starting point. However, a very large number of evaluations are required by the hybrid GA for a small improvement in DOC during local searches.

Comparison with the QP for known feasible and DoE starts in 34–dimensions shows that QP is quicker and achieves a better optimum for a feasible starting point. However, QP appears to have

Var. No.	Starting value	Final value	Movement
1	614824.0000	579744.1250	- 5.7 %
2	860.0500	878.3734	+ 2.1 %
3	7.7188	7.7997	+ 1.0 %
4	0.6420	0.6472	+ 0.8 %
5	0.3095	0.3315	+ 7.1 %
6	0.1505	0.1432	- 4.9 %
7	0.1022	0.0900	*LOWER*
8	0.0939	0.0900	*LOWER*
9	38.5068	38.2133	- 0.8 %
10	38.5068	38.2133	- 0.8 %
11	5.5254	5.2120	- 5.7 %
12	0.1635	0.1438	-12.1 %
13	0.2578	0.2149	-16.6 %
14	0.4130	0.4351	+ 5.3 %
15	0.3300	0.3418	+ 3.6 %
18	23.0000	23.0000	*UPPER*
19	32.0000	31.5136	- 1.5 %
21	4.0175	4.0283	+ 0.3 %
22	0.2527	0.2542	+ 0.6 %
23	121.7730	128.0341	+ 5.1 %
25	199.2000	209.6489	+ 5.2 %
33	370750.0000	327129.6250	-11.8 %
39	385280.0000	375586.4375	- 2.5 %
42	0.6054	0.6188	+ 2.2 %
48	528060.0000	492989.1250	- 6.6 %
49	0.6525	0.6561	+ 0.6 %
51	391725.0000	375523.7500	- 4.1 %
52	23.0000	20.3109	-11.7 %
53	20.0736	18.7838	- 6.4 %
54	23.0000	23.0000	*UPPER*
55	25.3581	25.9294	+ 2.3 %
56	122.0400	122.5491	+ 0.4 %
57	120.5400	121.0918	+ 0.5 %
59	183.0030	185.6261	+ 1.4 %

Table 11: QP Results after 226 iterations and 17118 calls to *design*.

difficulty in obtaining a feasible point when starting outside the constraint boundaries. The hybrid GA demonstrates potential for tackling this problem, although the heavily constrained 34-dimensional design space proves to be a difficult problem, requiring a large number of evaluations. However, such searches may still be the quickest way of generating a feasible design. The DoE part of the hybrid GA was used successfully to provide useful multiple starts for QP.

As visualization showed that the design space was not fundamentally multi–modal, the primary reason for resorting to use of the genetic algorithm was to overcome non–linearities in the data which resulted in small local optima. Therefore an RBF hypersurface fitting algorithm was applied to each of the constraints and objective function and the $SIMPLEX$ method was used to optimise the resulting design space. Although this worked well in four dimensions, the right combination of an accurate curve fit and smoothed local minima was too difficult to achieve in 34–dimensions for this problem with 52 constraints.

Comparison between QP and the $SIMPLEX$ method alone show that QP is faster and is more likely to obtain an answer on the design boundary. However, preliminary speeds of the $SIMPLEX$ method may be significantly improved by judicious trimming of the number of evaluations per iteration and iterations per optimization.

It is thought that the application for the hybrid GA will not be to simply optimize designs, but rather to find feasible starting points for input to a gradient search technique, such as the $SIMPLEX$ method in isolation or QP.

It is clear that the performance of the $SIMPLEX$ method alone and hybrid GA will improve with use and tuning of the multiple input parameters and design variable and constraint scaling factors.

Constraint No. & Type	Constraint Value - *SIMPLEX*	Constraint Value - QP
Initial 1#	0.1461	0.4805
Initial 4#	0.3557	1.8577
Initial 6#	1.2126	1.3894
Initial 7#	0.1791	0.2334
Initial 9#	0.0000	0.1768
Initial 20=	-0.0064	0.0000
Initial 27#	-0.0006	0.0000
Initial 29#	1.1465	6.0820
Initial 32#	-170.5625	-0.0156
Initial 33#	0.0035	0.1192
Initial 34#	-0.0001	0.0098
Initial 36#	0.0007	0.0084
Commonality 45#	0.1331	0.0626
Commonality 46#	0.6077	1.0839
Longit. S&C 64#	0.0028	0.0000
Longit. S&C 67#	0.2437	0.2122
Longit. S&C 71#	0.0328	0.0300
Longit. S&C 73#	0.2601	0.2188
Longit. S&C 74#	0.2437	0.2122
Longit. S&C 76#	0.0001	0.0000
Longit. S&C 78#	0.0829	0.0739
Longit. S&C 79#	0.0366	0.0299
Longit. S&C 80#	0.2601	0.2188
Longit. S&C 81#	0.2437	0.2122
Longit. S&C 82#	0.0083	0.0287
Longit. S&C 84#	0.0771	0.0861
Longit. S&C 85#	0.0749	0.0659
Longit. S&C 87#	0.3347	0.2934
Longit. S&C 88#	0.3187	0.2872
Longit. S&C 89#	0.0004	0.0225
Longit. S&C 90#	0.0301	0.0300
Longit. S&C 94#	0.1647	0.0000
Engine Out 112#	0.0116	0.0132
Take off 113#	0.0000	0.0000
Take off 114#	40.4	0.0
Take off 116=	-119.4	0.0
Climb 118#	0.000	0.855
Climb 119#	0.5011	0.8345
Cruise 121#	0.6373	1.0312
Climb 127#	1.0906	1.3511
Cruise 129#	1.4592	1.7555
Hold 150#	0.00	0.04
Overall Mission 151#	-15.9	37296.7
Overall Mission 152#	-363.6	-0.2
Climb 159#	3.8644	4.5057
Cruise 161#	3.6482	4.1867
Climb 167#	3.1324	2.8977
Cruise 169#	3.4219	3.2038
Climb 176#	0.2153	0.3017
Hold 190#	0.29	0.04
Overall Mission 192#	11055.7	0.0
Approach/Air-drop speed 193#	0.0056	0.5344

Table 12: $SIMPLEX$ method results after 5 iterations and 43436 calls to *design*. QP results after 226 iterations and 17118 calls to *design*. # - inequality constraint, = - equality constraint and S&C - stability and control.

Objective	Start	Final	Movement	no. of evals
SIMPLEX DOC	119784.19	113892.76	-4.918%	43436
QP DOC	119699.53	113613.52	-5.084%	17118

Table 13: Comparison of QP and $SIMPLEX$ method final DOC values.

6 Future Work

This report highlights a number of topics for future work. The first is that understanding of the design space to be optimized is limited due to the large number of design variables and constraints utilized and the complexity of the objective functions to be optimized. This sort of understanding would help the set up of the optimization problem to be solved. A number of visualization techniques, such as hierarchical axes techniques [5] and [13] and Kohonen's Self–Organizing Maps (*SOM's*) [10] have been proposed although they may not extend well into 34 dimensional design spaces. Future work will use techniques such as these to derive an understanding of the interplay between the design variables and constraints. It is expected that this type of visualization would provide insight in addition to the viewing of aircraft general arrangements that currently takes place in aircraft conceptual design teams.

A deeper understanding of the relationship between the objective function, penalty function and design variables in the problem would enable better solving of the problem constraints. Extensive testing of different weightings within the penalty terms was not undertaken here. At present design space interior terms (i.e. terms involving satisfied constraints) have only been tested in the hybrid GA and were not utilized in the stand alone $SIMPLEX$ method. In the first instance it is thought that further experimentation should take place with the constraint weightings by modifying the scaling in the optimization problems and by modifying the form of the penalty functions used. Further experimentation with the curve fitting algorithms may also be productive.

It is also clear that a considerable amount of user knowledge goes into the optimizations which is not being fully brought to bear in the automated process described here. In an improved process the knowledge capture activities being undertaken by the University Technology Partnership (*UTP*, see for instance Wallace *et al* [22]) may also be implemented to good effect.

Finally, it is good optimization practice to remove equality constraints completely from the problem, by elimination of design variables. The conceptual design problem has a number of equality constraints the elimination of which, if possible, should improve the GA results in particular.

7 Acknowledgements

This work was funded by Airbus U.K. Ltd. and the Department of Trade and Industry (DTI) and could not have been performed without the help of Murray Cross of AIRBUS U.K. and Alan Gould of the BAE SYSTEMS, Advanced Technology Centre (ATC).

References

[1] M.H. Bakr, J.W. Bandler, K. Madsen, and J. Sondergaard. Review of the space mapping approach to engineering optimization and modeling. *Optimization and Engineering*, 2, 2001.

[2] L. Davis. *Handbook of Genetic Algorithms*. Number ISBN 0-442-00173-8. Van Nostrand Reinhold, 1991.

[3] Goldberg, D.E. *Genetic Algorithms - In Search, Optimization and Machine Learning*. Number ISBN 0-8247-1633-7. Marcel Dekker Inc., 1982.

[4] W.E. Hart. *Adaptive Global Optimization with Local Search*. PhD thesis, University of California, San Diego, June 1994.

[5] C.M.E. Holden. *Visualisation for Aircraft Design Optimisation*, ATC, Sowerby, JS14571, February 2001. PhD Transfer Thesis, Southampton University.

[6] J.H. Holland. *Adaptation in Natural Artificial Systems*. University of Michigan Press, 1975.

[7] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[8] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[9] Keane, A.J. The OPTIONS design exploration system, reference manual and user guide. Technical Report Version B0.3, Dynamics Modelling Limited, December 1994.

[10] T. Kohonen. *Self–Organizing Maps*. Number ISBN: 3-540-67921-9. Springer–Verlag, 2000.

[11] M.D. McKay, Beckham R.J., and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Techometrics*, 21(2):239–245, 1979.

[12] N. Metropolis, A. Rosenbluth, Rosenbluth M., A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 6:1087, 1953.

[13] T. Mihalalisin, J. Timlin, and J. Schwegler. Visualizing multivariate functions, data and distributions. In S.K. Card, J. Mackinlay, and B. Shneiderman, editors, *Readings in Information Visualization: Using Vision to Think*,

number 1-55860-533-9, pages 115–125. Morgan Kaufmann, 1999.

[14] Nelder, J.A. and Mead, R. A simplex method for function minimisation. *Computer Journal*, 7:308–313, 1965.

[15] Y. S. Ong and A. J. Keane. Knowledge based search advisors for design problem solving environments (part I - knowledge modelling methodologies for optimizer selection). *Engineering Applications of Artificial Intelligence*, 2001.

[16] Y. S. Ong and A. J. Keane. Knowledge based search advisors for design problem solving environments (part II - case studies on existing design problem domains). *Engineering Applications of Artificial Intelligence*, 2001.

[17] Siddall, J.N. *Optimal Engineering Design*. Number ISBN 0-8247-1633-7 in Mechanical Engineering; 14. Marcel Dekker Inc., New York, 1982.

[18] J.J. Skrobanski. *Optimisation Subject to Non-Linear Constraints*. PhD thesis, London University, 1986.

[19] Toropov, V.V. Simulation approach to structural optimization. *Structural Optimization*, (1):37–46, 1989.

[20] Toropov, V.V. and Filatov A.A. and Polynkin, A.A. Multiparameter structural optimization using fem and multipoint explicit approximations. *Structural Optimization*, (6):7–14, 1993.

[21] van Keulen, F. and Toropov, V.V. New developments in structural optimization using adaptive mesh refinement and multipoint approximations. *Eng. Opt.*, (29):217–234, 1997.

[22] K.M. Wallace, C.W. Clegg, and A.J. Keane. Visions for engineering design: A multi-disciplinary perspective. In *International Conference on Engineering Design*, ICED 01, Glasgow, U.K., August 21–23 2001.

[23] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. *Numerical Recipes, The Art of Scientific Computing, Second Edition*. Number ISBN 0-521-43064-X in 1. Numerical Analysis - Computer Programs. Cambridge University Press, 1992.

[24] Yin, X. and Germay, N. *A Fast Genetic Algorithm with Sharing Scheme Using Cluster Methods in Multimodal Function Optimisation Approximation Method, In: Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, ed. R.F. Albrecht, C.R. Reeves and N.C. Steele.* Number pp. 450-457. Springer-Verlag, Innsbruck, 1993.

A Definition of Terms Used

The nature of the terms *penalty function*, *fitness*, *objective*, *constraints* and *penalty* and are all somewhat arbitrarily defined in the optimization literature. Therefore to clarify, for the purposes of this paper the definitions used are as follows:

- The term *penalty function* is applied to the function used to calculate the *fitness*, including terms for the *objective* and *constraints*. The *penalty function* used in the Genetic Algorithm and *SIMPLEX* method used in this paper is given in equation 1, and further description of this and other examples are provided and discussed in Siddall's book, [17].

$$\begin{aligned} p(\mathbf{x}) = Obj(\mathbf{x}) \quad &+ \quad \frac{1}{r}\sum_{j=1}^{L} c_j^V(\mathbf{x})^2 \\ &+ \quad r^2 \sum_{j=1}^{L} \frac{1}{c_j^S(\mathbf{x})} \end{aligned} \tag{1}$$

  The quadratic programming method utilizes:

$$p(\mathbf{x}) = Obj(\mathbf{x}) + \sum_{j=1}^{L} \lambda_j c_j^A(\mathbf{x}) + \nu \sum_{j=1}^{Q} (c_j^A(\mathbf{x}))^2. \tag{2}$$

  where superscript A indicates an "active" constraint as determined by the quadratic programming method.

- The term *fitness* is chosen to denote the single figure of merit for an individual in the population including contributions from the objective and constraints. Any single number numerical result from equation 1 would be a *fitness*.

- The term *objective* and *objective function* is applied to the cost, to be maximized or minimized, excluding the *constraints*. In equation 1, $Obj(\mathbf{x})$ is the *objective*.

- The term *constraints* is applied to those functions which limit the extent or range of the *objective*. These are to be satisfied during the optimisation and may include design variable bounds which may be included in the problem in the same or different ways. The terms $c_j^V(\mathbf{x})$

and $c_j^S(\mathbf{x})$ are the violated and satisfied constraints, respectively, in equation 1.

- The term *penalty* is applied to the terms associated with the *constraints and their weightings* in the *penalty function.* In equation 1 the terms:

$$\frac{1}{r}\sum_{j=1}^{L} c_j^V(\mathbf{x})^2 \ \ and \ r^2 \sum_{j=1}^{L} \frac{1}{c_j^S(\mathbf{x})}$$

are the *penalty* terms.

B The Genetic Algorithm

A more detailed description of the Genetic Algorithm (GA) method can be found in Davis [2] and Holland [6]. The procedure for the GA used in this paper is summarized in Figure 4.

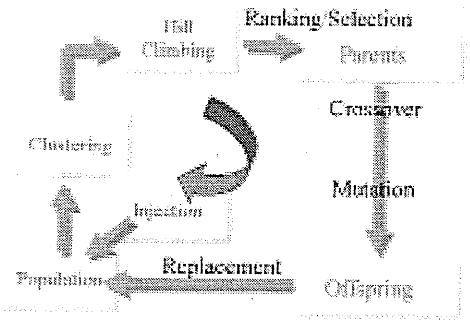

Figure 4: *Schematic to show the constituents of the proposed hybrid GA. The hill climbing does not necessarily take place at every generation.*

Genetic algorithms are usually started using an initial population *randomly* selected from the design space. This population provides an initial sample of the design space. However, the quality of the statistical sample of the data is improved by using a Latin hypercube DoE (from McKay [11]). The design space is evenly divided into a number of levels corresponding to the number of design variables in the problem and on each level of every design variable one point is placed at random.

This Genetic Algorithm provides design variable encoding in terms of a variable number of decimal digits. This is thought to be a compromise between binary coding which is not as efficient or often as accurate (although this depends on the number of digits used) and real number coding which can spend to long searching irrelevant regions of the design space.

The Genetic Algorithm uses a set of operators on the current population of solutions to create a new generation of candidate solutions. At the basic level there are two main operators . These operators are *crossover*, which combines the information between two solution strings, and *mutation*, which is the random perturbation of the digits in the solution string.

The Genetic Algorithm used here also uses a *clustering* or *niche forming* algorithm to delay the convergence of the algorithm, so that more widely spaced global optima are located as opposed to similar local ones [24]. To effect this, the fitness is modified so as to penalize population members in large clusters and those close to the cluster centroid. This encourages the formation of new clusters. The new *fitness* is given by:

$$f_i' = \frac{f_i}{m_i'} \tag{3}$$

where:

m_i'	$= n_c - n_c \times \left(\frac{d_{ic}}{2d_{max}}\right)^\alpha$
α	is a constant
d_{ic}	is the distance between the individual i and its niche's centroid
d_{max}	is the maximal distance that clusters can be apart
n_c	is the number of individuals in the cluster

and the individual x_i belongs to the cluster C.

Constraint penalty functions are used to drive the result of the optimization away from the constraint boundaries. Strings are selected for crossover randomly on a basis of their *fitness.* That is a more fit string is more likely to be selected as a parent, for combination with another string. The intention here is to propagate aspects of the fit solutions into the next generation such that as the algorithm iterates from one generation to the next the *fitness* of the solutions increases, hopefully converging to the optimal solution. So called 'roulette wheel' parental selection is utilized in this algorithm to achieve this.

The literature shows that so called hybrid methods potentially give better solutions than genetic algorithms in isolation, see for instance [4]. In the hybrid method, once promising regions have been located, an efficient local technique (e.g. the $SIMPLEX$ method) can be used to converge on precise minima.

In keeping with the Darwinian idea of survival of the fittest, the GA is a maximizing optimization algorithm. The GA also does not recalculate *objective* or *constraint* values already known from the previous generation, but uses the previously calculated value, the *fitness* itself is though always recalculated.