A Broad and Narrow Approach to Interactive Evolutionary Design – An Aircraft Design Example

Oliver Bandte and Sergey Malinchik

Icosystem Corporation, 10 Fawcett Street, Cambridge, Massachusetts 02138
{oliver, sergey}@icosystem.com
http://www.icosystem.com

Abstract. While Interactive Evolutionary Computation (IEC) is starting to penetrate a larger scientific community, only few researchers have applied IEC to the design of complicated artifacts like machines or transportation systems. The present paper introduces a specific approach to Interactive Evolutionary Computation that breaches the two historical categories of user-defined fitness and selection in each generation (narrow) and occasional user-intervention of an automated evolutionary process to correct the fitness function used for (multi-objective) optimization (broad). To highlight the approach, a real world aircraft design problem is employed that demonstrates the relevance and importance of both features for an effective design process.

1 Introduction

Interactive Evolutionary Computation (IEC) has started to capture the fascination of researches from fields as diverse as art, architecture, data mining, geophysics, medicine, psychology, robotics, and sociology. Takagi outlined many of these applications in his overview paper [1]. However, to this day only very few researchers have applied IEC to the problem of engineering and design of complicated artifacts. While the main reason for the slow pace of adoption in engineering is mostly open for speculation, it is partially a result of the field's reluctance to accept new methods, like Genetic Algorithms, as well as the field's already heavy reliance on automated optimization processes that leave decidedly little room for subjectivity. Among the few that did apply IEC to engineering design, Parmee has to be given the adequate credibility for his research. [2], [3]

However, without notable exception, all research currently done in applying IEC to engineering design seems to belong to the category of broad-IEC. Takagi introduced this term in [1], separating the applications of IEC into two categories, *broad* and *narrow*, based on the type of interaction the human has with the evolutionary process. According to that definition, a narrow-IEC uses the human input as the fitness measure for a population member, while the broad-IEC utilizes a numerical fitness value in conjunction with an interface through which the human can guide or interact with the evolutionary process. That means, the broad-IEC can accommodate systems for which we already have a numerical fitness measure, and take advantage of the interactive capability to benefit the optimization procedure. On the other hand,

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 883–895, 2004.

narrow-IEC can be utilized for systems that do not provide a numerical fitness measure, and rather rely on the (subjective) judgment of a human individual or group.

Accordingly, the aforementioned strong reliance on numerical evaluation in engineering design, see [4], [5], [6], [7], resulted in research efforts focusing on broad-IEC techniques. For example, Parmee introduced an Interactive Evolutionary Design System that enables designer interaction after a certain number of generations, presenting the current population and allowing for a redirection of the search through a change in preferences for multiple objectives. [2], [3] All fitness evaluations are performed internal to the evolutionary process, utilizing an analysis tool that calculates multiple system performance values, i.e. objectives, as well as the preference information supplied (interactively) by the designers.

While the reliance of engineers on analysis tools requires interactive evolutionary techniques to utilizes them in the fitness generation, it is also true that many design decisions in practice are made through gut feel and intuition rather then analysis. Recognizing that fact, this paper identifies an IEC approach to design that can be called broad and narrow, allowing for automatic fitness calculation through analysis as well as selection and fitness assignment by the human designers directly.

2 Proposed Process

The approach proposed in this paper is intended to be comprehensive with regard to the implementation into an actual engineering design process. Hence, a process, outlined in Figure 1, was established first that identifies ten steps from problem set-up to deciding on a final design.

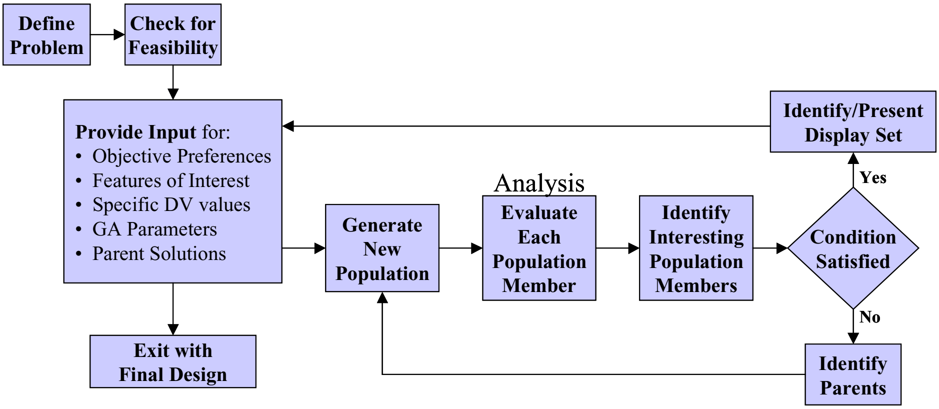


Fig. 1. Integrated Interactive Evolutionary Design Process

Step 1 – Define Problem

As in any design problem, the first step is to define the independent parameters, objectives and constraints, as well as evaluation functions that describe the objectives' dependencies on the independent variables. Additionally, for this interactive evolutionary design environment this step identifies the genotype representation of a design alternative, the fitness evaluation function, influenced by the objectives, and how to handle design alternatives that violate constraints.

For many design problems the fitness of a solution is based on the satisfaction of multiple objectives. In such cases, a wide variety of approaches to fitness assignment and selection can be employed for this interactive evolutionary design process. Naturally, such established MOGA methods as Goldberg's Pareto ranking [8] or the utility function method with weights representing the relative preferences for the objectives [9] lend themselves as immediate choices. However, without presenting proof here, such techniques as Branke's Focused Pareto Frontiers [10], Deb's Goal Programming adaptation [11], Parmee's Fuzzy Preferences [2,3], or even currently un-proposed uses of TOPSIS [12], LINMAP [13], and MinMax/MaxMin could present a useful solution to the well-known shortcomings of the established methods.

Related to the problem of selecting a fitness assignment approach is the issue of proper treatment of problem constraints. While solutions with good fitness values are desired, they cannot be considered a final solution, as long as they violate the identified constraints. However, in order to enhance the speed with which the genetic algorithm can find good feasible solutions, it may be beneficial to allow for a certain level of infeasibility in other population members. Again, without going into details here, such ideas as inclusion of constraints in the fitness function, fitness discounting or repairing of infeasible alternatives, flat out rejecting infeasible solutions, or treatment of the constraints as objectives in one of the Multi Objective Optimization techniques have been proposed in the past. [14]

Step 2 – Check for Feasibility

The next step entails a check for sufficient feasible design space to search for an interesting optimal solution. While different techniques can be employed here, the simplest approach is to test a large random sample of design variable values, generated from uniform distributions, and to count the number of times constraints were violated. If less then 10% of the sample satisfies all constraints, an effort should be made first to relax the constraints or introduce new technologies to the system, altering the dependency of the constraint functions on the design variable settings. If neither is possible, an interactive evolutionary approach might simply be too expensive for evaluating a small set of possible solutions.

Step 3 – Provide Input

This step represents the central interaction point of the designers with the environment. Here they process the information displayed and communicate preferences for objectives, features of interest in particular designs, whether specific design variable values should be held constant in future iterations, what parameter setting the GA should run with in the next iteration (e.g. a condition that identifies the end of the GA iteration), or whether specific design alternatives should serve as parents for the next generation.

The difficulty in implementing this step is identifying *what information* to display and *how*, so that the designers can easily process it. For example, a general area of research that will benefit this step is the display of multi-dimensional data. Since part of the information conveyed to the designers is data for the objective functions and design variables, it is guaranteed to be multi-dimensional (higher than three!), for which data visualization is not straightforward.

After analyzing the displayed information, giving feedback is the next critical element in which the designers exert their influence on the optimization process through parent selection for the GA, objective preferences for the automated alternative evaluation procedure, or guiding the search for a design with specific features. Unfortunately from a scientific perspective, what feedback is deemed relevant is highly project and designer experience specific, so that no further guidelines can be given at this point. Hopefully experience gained from future work will yield some insights into some must-haves and must-avoids.

Finally, most actual engineering design work is done by a team of designers (or experts representing different disciplines) rather then one person, hence the issue of differing or opposing opinions among experts/designers needs to be addressed in order for any interactive process to be implemented. Again without proof, several voting and scoring approaches (see Hwang [15]) can be used to accommodate the different opinions and provide a fair framework to generate a collective group opinion.

Step 4 – Generate New Population

Utilizing the information provided by the designers previously, this step simply generates a new population of alternatives. Here the selection of appropriate variation operators (mutation, crossover, weighted average, permutation, or representation specific) becomes important to guarantee the efficiency of the process.

Step 5 – Evaluate Each Population Member

Next, a numerical analysis is employed to distinguish the different designs in the population by their objective and constraint values. In order to provide direct and immediate information to the designers, an *integrated* analysis tool is critical to the success of any “broad” interactive evolutionary design process. Having to wait for a lengthy period of time, just to see the result of a particular action the designers took, would clearly interrupt the dynamic of an interactive environment. While most complex engineering design projects inherently provide such analysis, some of these tools can be very time consuming (e.g. Finite Element Analysis). In many of these cases metamodel analysis (see Mavris [16]) is recommended to facilitate a speedy evaluation during the interactive process, allowing for detailed optimization with the original analysis tool at a later time.

Step 6 – Identify Interesting Population Members

While guaranteeing an efficient search with a genetic algorithm requires a large population size, it seems impractical and unnecessary to subject the designers to the entire (large) population, including the highly dominated alternatives. It is therefore proposed to down-select the population to a more manageable group of interesting solutions. For example, such set of solutions could be the Pareto frontier itself, or alternatives dominated by at most one other population member.

Step 7 – Condition Satisfied
This step simply tests whether the designer-specified condition for ending an iteration is satisfied. This condition can vary from simply reaching the number of generations an iteration was supposed to entail, to reaching a specified fitness or objective function level, to reaching a specified minimum distance level between design alternatives, that guarantees a diverse population. In some cases the designers may elect to execute an iteration with only a few generations, but in general it is anticipated that the GA is supposed to identify significantly improved design alternatives requiring several generations. However, to guarantee continuous involvement of the designers, each iteration should be limited to a reasonable length. Once the condition is satisfied, the set of interesting solutions is (possibly) further reduced to a display set. Otherwise, new parents are being selected for the GA's next generation.

Step 8 – Identify Parents
In this proposed interactive evolutionary design process this selection step closes the loop of the evolutionary process. Specifically, it determines, based on the fitness of each population member and/or the designers' feedback, which solution participates in the creation of the next generation, and which solution has to leave the pool of useful solutions.

Step 9 – Identify and Present "Display Set" of Solutions
For problems with a large number of objectives, it is possible that the set of interesting solutions, e.g. the set of Pareto optimal solutions, is very large and it becomes infeasible to display all interesting solutions. In such cases a further step of down-selecting the alternatives to be displayed is needed. Promising approaches are cluster analysis, displaying a representative alternative that is central with respect to objective or design variable values, as well as simply selecting a fixed number of alternatives with highest fitness to be presented to the designers.

Step 10 – Exit with Final Design
Finally the process ends with the designers simply selecting the solution that satisfies their design requirements best. Naturally, the point at which this step happens is a function of time constraints the design project is subject to.

3 Implementation Example

To demonstrate the interactive evolutionary design approach proposed here, a supersonic business jet was chosen as a complex aerospace design problem for implementation over a simple benchmark problem. The analysis tool used in this example is a system of Response Surface Equations (see Buonanno [17] for details) that allows for a rapid fitness evaluation.

Problem Definition
For this supersonic business jet example, 35 design variables were chosen, separated into five groups as presented in Figure 2. The first group, General, consists of

variables for the vehicle, some of which could be designated Design Requirements. The other four groups contain geometric parameters for the wing, fuselage, empennage, and engine, while the engine group also entails performance parameters relevant to the design.

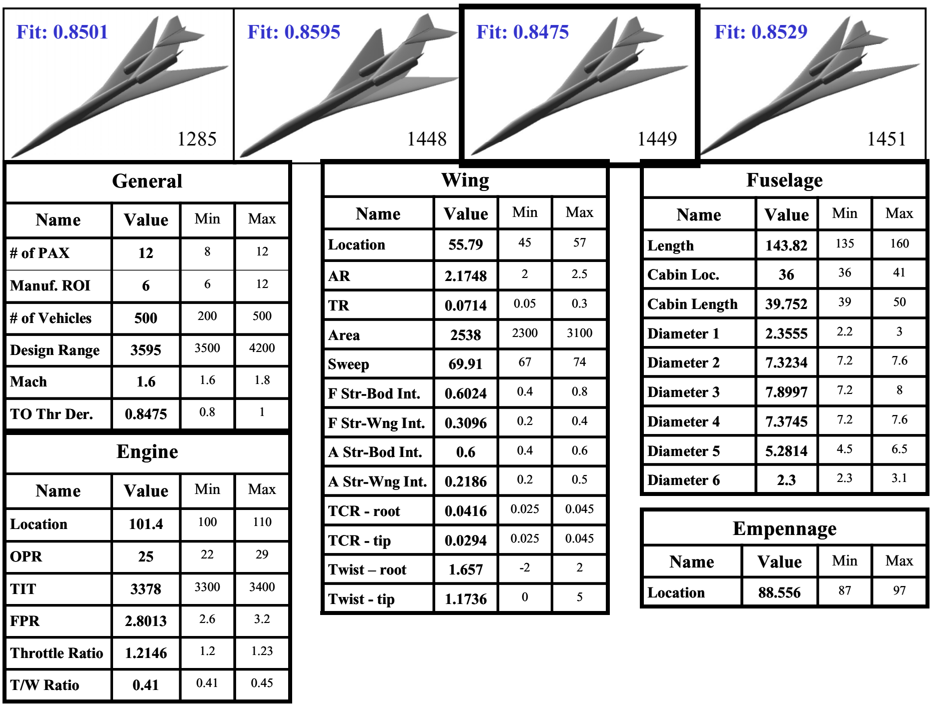
A mix of economic, size, and performance parameters were chosen as the eight objectives to be minimized in this example. Special emphasis is put on noise generation, since it is anticipated to be a primary concern for a supersonic aircraft. Hence, for the initial loop the Boom Loudness and, as a counter weight, the Acquisition Cost are given slightly higher importance of 20%, while all other objectives are set at 10%. Furthermore, certain noise levels could be prohibitively large and prevent the design from getting regulatory approval. Hence, the Sideline and Flyover Noise objectives have to have constraint values imposed on them. In addition to these constraints, the design has to fulfill certain FAA requirements regarding take-off and landing distances as well as approach speed. Furthermore, in order for the design to be physically realizable, the amount of available fuel beyond what is needed to complete the design mission needs to be positive and include a certain amount of reserve. Finally, fitness is calculated via a weighted sum of normalized objective values, penalized by a 20% increase in value whenever at least one constraint is violated. Note that the best solution is identified as the one with the **lowest** fitness. Both, objectives and constraints are tabulated in Figure 3.

Genetic Algorithm

The GA chosen for this example has a population size of 20 and makes use of a real valued 35-gene representation. New generations are created from an elite pool for the two individuals with best (lowest) fitness, and proportionate probabilistic selection for crossover. The crossover algorithm utilizes a strategy with one splice point that is selected at random within the chromosome. Since the design variables are grouped in logical categories, this crossover algorithm enables a complete swap of the engine or fuselage-engine assembly between parents. Parent solutions are being replaced with offspring. Each member of the new population has a 15% probability for mutation at ten random genes, sampling a new value from a uniform distribution over the entire range of design variable values. The GA in this example is used for demonstration purposes only and therefore employs just a small population. A population size of 50 to 100 seems more appropriate for a more involved version of the proposed approach.

Interactive Process

For this implementation example, the GA was interrupted at 80, 160, 240, 400, 560, and finally 800 generations to display the population of design alternatives found to that point in form of pictures with objective values and aircraft configurations. [Due to the page restriction, only a sample display can be presented here in Figures 2 and 3. Please refer to www.icoserver.com/~oliver/GECCO2004 for a complete series of screenshots, or consult [18] for a discussion on the relevance of the display information itself.] Since in this example the GA's population size is so small, no specific concentration on interesting solutions is being performed here, but in order to allow for a reasonable display size of the aircraft configuration, only the four best design alternatives, based on fitness and highest diversity in geometrical features, are being presented in detail on the top of the left-hand side of the display. A screenshot of the displayed chromosome related information is presented in Figure 2,



General

Name	Value	Min	Max
# of PAX	12	8	12
Manuf. ROI	6	6	12
# of Vehicles	500	200	500
Design Range	3595	3500	4200
Mach	1.6	1.6	1.8
TO Thr Der.	0.8475	0.8	1

Engine

Name	Value	Min	Max
Location	101.4	100	110
OPR	25	22	29
TIT	3378	3300	3400
FPR	2.8013	2.6	3.2
Throttle Ratio	1.2146	1.2	1.23
T/W Ratio	0.41	0.41	0.45

Wing

Name	Value	Min	Max
Location	55.79	45	57
AR	2.1748	2	2.5
TR	0.0714	0.05	0.3
Area	2538	2300	3100
Sweep	69.91	67	74
F Str-Bod Int.	0.6024	0.4	0.8
F Str-Wng Int.	0.3096	0.2	0.4
A Str-Bod Int.	0.6	0.4	0.6
A Str-Wng Int.	0.2186	0.2	0.5
TCR - root	0.0416	0.025	0.045
TCR - tip	0.0294	0.025	0.045
Twist – root	1.657	-2	2
Twist - tip	1.1736	0	5

Fuselage

Name	Value	Min	Max
Length	143.82	135	160
Cabin Loc.	36	36	41
Cabin Length	39.752	39	50
Diameter 1	2.3555	2.2	3
Diameter 2	7.3234	7.2	7.6
Diameter 3	7.8997	7.2	8
Diameter 4	7.3745	7.2	7.6
Diameter 5	5.2814	4.5	6.5
Diameter 6	2.3	2.3	3.1

Empennage

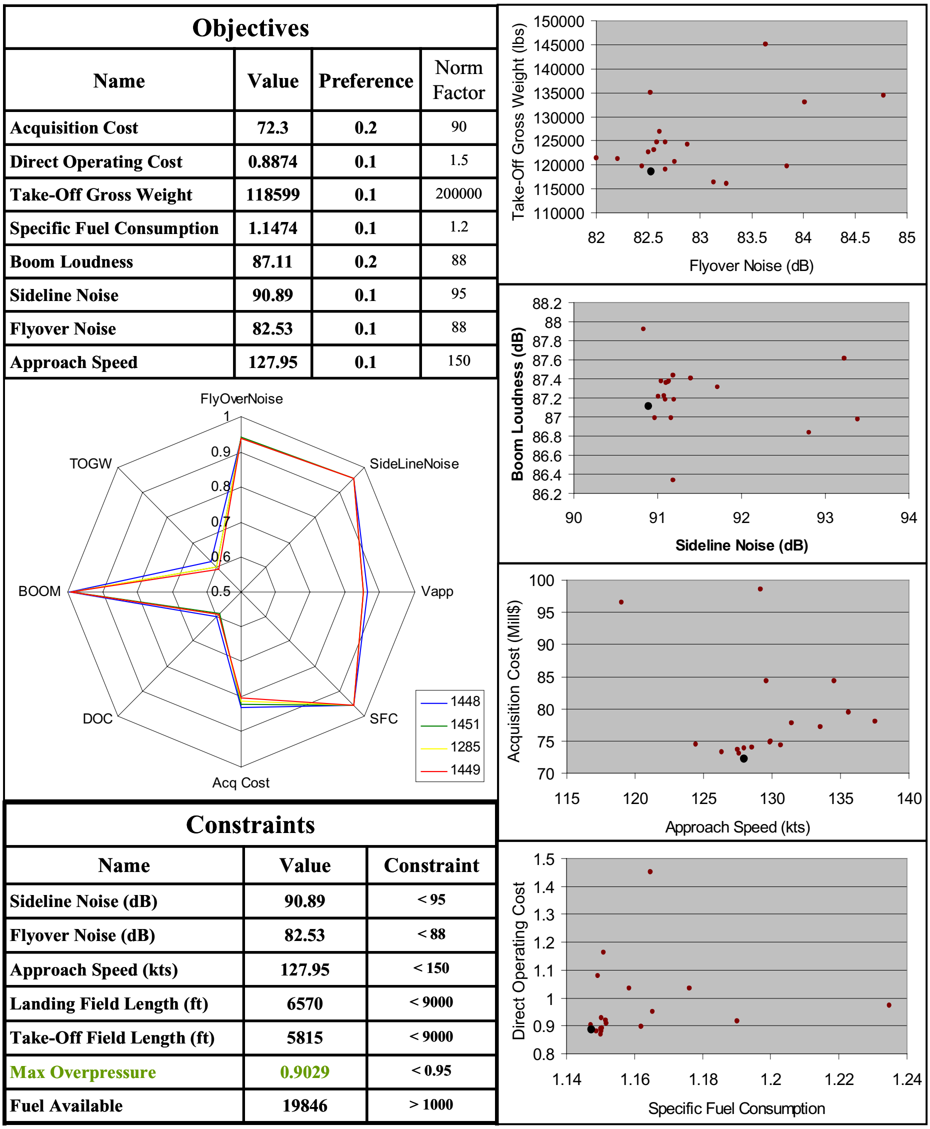
Name	Value	Min	Max
Location	88.556	87	97

Fig. 2. Left-hand Side of Screenshot – Chromosome Information

highlighting the individual with the best/lowest fitness, which is also enlarged on the screen to provide the designers with a more detailed view of the selected configuration (not displayed here). Presented below the larger image are the design variable values for the highlighted alternative, and their respective ranges, representing the actual chromosome of the highlighted alternative.

On the right-hand side of the screen, displayed in Figure 3, the designers can find the objective and constraint information pertaining to the population and the highlighted individual. On the top, a table outlines the specific objective values for the highlighted alternative, as well as the objective preferences and normalization factors used to generate the fitness values for the current population. Below the table, a spider graph compares the four presented alternatives on the basis of their normalized objective values, and to the right four graphs display the objective values for the entire population, identifying its Pareto frontier. Below the spider chart, a table lists the constraint parameter values for the highlighted alternative as well as the respective constraint values. A green font represents constraint parameter values near, orange font right around, and red font way beyond the constraint value.

Based on this information the designers can make some choices with respect to objective preferences and/or selection of features-of-interest. To limit the scope of this example, only the redirection of the search through changes in objective preferences is implemented here. However, as described before and exemplified later, designer selection of features-of-interest is an important part of interactive



Objectives

Name	Value	Preference	Norm Factor
Acquisition Cost	72.3	0.2	90
Direct Operating Cost	0.8874	0.1	1.5
Take-Off Gross Weight	118599	0.1	200000
Specific Fuel Consumption	1.1474	0.1	1.2
Boom Loudness	87.11	0.2	88
Sideline Noise	90.89	0.1	95
Flyover Noise	82.53	0.1	88
Approach Speed	127.95	0.1	150

Constraints

Name	Value	Constraint
Sideline Noise (dB)	90.89	< 95
Flyover Noise (dB)	82.53	< 88
Approach Speed (kts)	127.95	< 150
Landing Field Length (ft)	6570	< 9000
Take-Off Field Length (ft)	5815	< 9000
Max Overpressure	0.9029	< 0.95
Fuel Available	19846	> 1000

Fig. 3. Right-hand Side of Screenshot – Fitness Information

evolutionary design and should not be neglected in general. Also, all designer analysis and feedback is performed by only one individual for this implementation study, leaving the implementation of multiple designer opinions discussed in Step 3 to future research.

The first opportunity for the designers to give feedback about the design alternatives is after 80 generations. From the data displayed it is apparent that all objectives are being satisfied well except for the Boom Loudness. In an attempt to

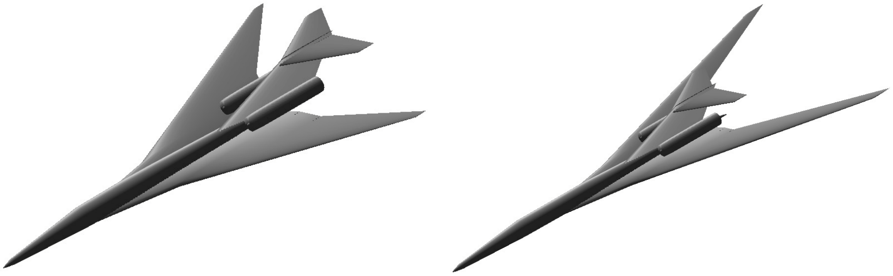


Fig. 4. Best Solutions after 400 and 800 Generations

achieve satisfactory levels for the Boom Loudness in the next iteration, its preference is increased to 30%, reducing the Acquisition Cost's importance to 10%. Note that this feedback/input constitutes a redirection of the search performed by the GA only, and does not constitute selection of a specific parent, component, or design variable value.

After 160 generations, the objective values of the best alternative indicate that the last set of preferences did not emphasize the Boom Loudness and Sideline Noise enough, since Boom Loudness did not improve and the Sideline Noise got worse. Consequently, for the next iteration the importance of both is increased to 35% while all other objectives are reduced to 5%.

After another 80 generations, the resulting population indicates that the last set of preferences still did not emphasize the Boom Loudness enough, since Boom Loudness improved only marginally. On the other hand, Sideline Noise did improve significantly, so that for the next iteration all emphasis can be given to Boom Loudness. To keep the score even for all other objectives, they are being kept at 5% with Boom Loudness at 65%.

As the interactive search progresses and the population of designs is more and more refined, it becomes more and more difficult to find significantly improved alternatives with only a few generations. Hence, 160 generations were executed for the next iteration, for which a very good solution is found in this 400th generation with largely improved values for almost all the objectives. However, in order to test the ability of this interactive evolutionary design approach to support dramatic shifts in search direction, Boom Loudness is now made the primary concern for the next iteration, setting its preference value to 93% while keeping all other objectives at 1%.

After another 160 generations, this preference setting finally produced a supersonic business jet design with a dramatically reduced sonic boom loudness below 85 dB. Unfortunately but not surprisingly, this design yields values for the other objectives that are less desirable, specifically for Sideline Noise. To remedy this shortcoming, another iteration is executed to find a solution that better balances Boom Loudness and Sideline Noise, giving the former 80% and the latter 20% preference. Consequently, all other objectives have a zero preference associated with them.

As anticipated, the new preference setting succeeded in identifying a design that balances Boom Loudness and Sideline Noise at very low values for both. Obviously, this gain does not come without a penalty in the other objectives, their values having significantly increased compared to previous iterations. However, the presented solutions after 400 and 800 generations, see Figure 4, seem to satisfy the objectives better than the published solution in [17], generated by MATLAB$^{©}$'s Fmincon function [19]. So, it can be concluded that none of these objective values are dramatically out of sync or range and the presented individual could be considered the final solution. That is, if the information displayed concentrated on the objectives only. However, at this point the display of the aircraft layout becomes critical and some comments have to be made about the unusual design configuration of this solution.

The implementation example for the presented process only considered objective preferences as an input from the designers. A more elaborate example, should also consider feedback for the design variables, affecting the selection process in the GA as much as the search process. This becomes particularly apparent when examining the configurations of the final solutions after 400 and 800 generations in Figure 4. The latter has an unusually long wing extending far beyond the vehicle's fuselage. For all practical purposes, this configuration would not be able to fly, since it would be impossible to land, unless it had a very long landing gear, and very difficult to navigate on the runway. However, the GA cannot be made aware of these issues, since they are not part of the analysis code employed for fitness evaluation. Hence, the GA by itself identifies this solution as the best (after 800 generations).

Fortunately, this weakness can be remedied by incorporating this intuitive designer knowledge into the interactive evolutionary approach via selection of features-of-interest. For example, after generation 560 the visualization already indicated a drastic change in geometry, compared to presented alternatives before, e.g. after generation 400. At this point the designers should have indicated that a configuration of this kind is undesirable by, for example, restricting the sweep and not allowing for the wing tips to be located that far aback.

Summary

To understand the impact of this interactive evolutionary approach on the generated design solutions in this implementation example, a comparison of the time series data for the Fitness, Boom Loudness, Sideline Noise and Acquisition Cost is presented in Figure 5. Note that new designer input was provided after 80, 160, 240, 400, and 560 generations. As the Fitness graph indicates, the GA is successful in reducing Fitness during each iteration, while the value-increases are solely attributable to the change in Fitness composition with every new preference values. The Fitness graph also illustrates a point made earlier that for large numbers of generations it is increasingly difficult for the GA to find better solutions.

The graph for the Boom Loudness value in turn illustrates that the GA was not improving Boom Loudness until its preference was increased to 65% (240+ generations) and only saw significant improvements at a preference of 93% (560+ generations). The Sideline Noise on the other hand saw some very good values between generation numbers 300 and 500. However, the strong pull for better Boom Loudness values increased the Sideline Noise values again, a shortcoming that was only remedied by increasing its preference to 20%. Finally, as indicated in the text, the Acquisition Cost reached good values early on and continued to improve until generation 400, but lost ground when significant preference was given to Boom Loudness and taken away from other objectives.

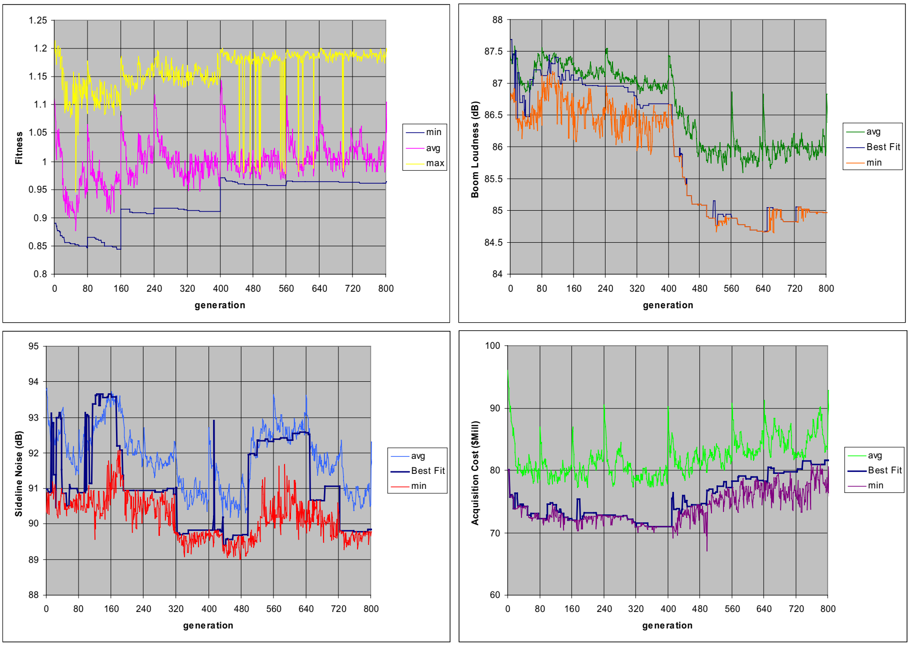


Fig. 5. Summary of Implementation Example

4 Conclusion

While the implementation example presented in this paper does not address all aspects raised in the background discussion of the proposed interactive evolutionary design approach, it is able to demonstrate that this IEC approach to design is feasible and it allows the designers to exert very specific control over the optimization process. Although it is difficult to compare this design approach to traditional techniques by virtue of their solutions, both aircraft design solutions presented in this paper achieve better objective values then a previously published solution just by redirecting the search through changing preferences for objectives that were not satisfied sufficiently by the alternatives presented.

Albeit the approach presented in this paper pushes interactive evolutionary computation toward the world of applied design, more work needs to be done before design practitioners will be able to use an interactive evolutionary design tool. Specifically, the presented example did not implement a multi-expert design feedback, although most complex design problems will draw on a multitude of experts. Hence, it is highly recommended to address this issue in the future. Furthermore, the implemented response surface equation is an easy and straightforward solution to the need for analysis, but it makes the design process susceptible to prediction accuracy. However, a more involved analysis code might increase the computational time to levels that prohibit collaborative interactive evolutionary design altogether. This issue needs further study to recommend different levels of analysis for different stages of design. Another issue that requires more research to give practitioners some guidance is related to the amount of information communicated to the designers. While large populations are needed for the evolutionary process to be efficient, analyzing more than two dozens designs at the same time may be infeasible for the average design team. Finally, the end-of-iteration condition is currently subject to discussions within the IEC community. Unfortunately, it is not clear at this point whether this condition will also be entirely problem dependent or general guidelines can be provided.

Acknowledgments. The authors would like to extend a special thanks to Dr. Dimitri Mavris of Georgia Tech's Aerospace Systems Design Laboratory for his guidance and advice on the implementation example, Michael Buonanno of ASDL for the supply of the response surface equations as well as Jesse Eyer of ASDL for the generation of the airplane images in RAM. This work was performed under ONR Contract # N00014-03M-0355.

References

1. Takagi, H.: Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation. Proceedings of the IEEE, Vol. 89. No.9 (2001) 1275 - 1296
2. Parmee, I. C., Cvetkovic, D., Watson, A., Bonham, C.: Multi-Objective Satisfaction within an Interactive Evolutionary Design Environment, Journal of Evolutionary Computation. Vol. 8. No. 2, MIT Press (2000) 197 - 222
3. Parmee, I. C., Cvetkovic, D., Bonham, C., Packham, I.: Introducing Prototype Interactive Evolutionary Systems for Ill-Defined, Multi-Objective Design Environments, Journal of Advances in Engineering Software. Vol. 32. No. 6. Elsevier (2001) 429 - 441
4. Asimow, M.: Introduction to Design, Prentice-Hall, Englewood Cliffs NJ (1962)
5. Blanchard, B. S., Fabrycky, W. J.: Systems Engineering and Analysis. 3rd edn. Prentice-Hall, New York (1998)
6. Dieter, G. E.: Engineering Design 2nd edn. McGraw-Hill, New York (1991)
7. Dixon, J.R.: Design Engineering: Inventiveness, Analysis, and Decision Making. McGraw-Hill Book Company, New York (1966)
8. Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)
9. Hwang, C.-L., Masud, A. S.: Multiple Objective Decision Making – Methods and Applications. Springer Verlag, Berlin, Heidelberg, New York (1979)
10. Branke, J., Kaussler, T., Schmeck, H.: Guidance in Evolutionary Multi-Objective Optimization. Advances in Engineering Software 32. Elsevier (2001) 499 - 507
11. Deb, K.: Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms, Congress on Evolutionary Computation. Vol. 1, IEEE (1999) 77 - 84
12. Hwang, C.-L., Yoon, K.: Multiple Attribute Decision Making – Methods and Applications. Springer Verlag, Berlin, Heidelberg, New York (1981)
13. Srinivasan, V., Shocker, A. D.: Linear Programming Techniques for Multidimensional Analysis of Preferences. Psychometrika. Vol. 38, No. 3, Psychometric Society (1973) 337 - 369
14. Michalewicz, Z., Fogel, D. B.: How to Solve It: Modern Heuristics. Springer Verlag, Berlin, Heidelberg, New York (1998)

15. Hwang, C.-L., Lin, M.-J.: Group Decision Making under Multiple Criteria– Methods and Applications. Springer Verlag, Berlin, Heidelberg, New York (1987)
16. Mavris, D. N., Bandte, O., Schrage, D. P.,: Application of Probabilistic Methods for the Determination of an Economically Robust HSCT Configuration. AIAA-96-4090. presented at Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA (1996)
17. Buonanno, M., Lim, C., Mavris, D. N.: Impact of Configuration and Requirements on the Sonic Boom of a Quiet Supersonic Jet. SAE 2002-01-2930. presented at World Aviation Congress, Phoenix, AZ (2002)
18. Bandte, O.: Visualizing Information in an Interactive Evolutionary Design Process. to appear at CEC 2004, Portland, OR (2004)
19. http://www.mathworks.com/access/helpdesk/help/toolbox/optim/optim.shtml