EVOLUTIONARY METHODS FOR DESIGN, OPTIMISATION AND CONTROL
K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou and T. Fogarty (Eds.)

THE PLACE OF EVOLUTIONARY SEARCH TOOLS IN PROBLEM SOLVING ENVIRONMENTS FOR AEROSPACE DESIGN

Andy J. Keane
School of Engineering Sciences
University of Southampton
Highfield
Southampton, SO17 1BJ, U.K.
Email: ajk@soton.ac.uk,
web page: http://www.soton.ac.uk/~ajk

Abstract. This papers sets out to show that evolutionary search tools have an important role to play in the development of Problem Solving Environments (PSE's) for aerospace design. Distributed and versatile PSE's are now being prototyped in a number of research and industrial groups around the world and can be expected to take a central place in the toolkit of future aerospace designers. The role played in these systems by evolutionary methods includes improvements to the design process as well as the search for better products via goal oriented optimization. These and related aspects of aerospace design are illustrated here with examples from the fileds of satellite, airframe and aeroengine design.

Key words: design, evolution, optimization, problem solving, aerospace, distributed.

1 INTRODUCTION

It is clear that the environments engineering designers work in are changing very rapidly. In fifteen years design offices have moved from serried ranks of A0 drafting machines, through expensive dedicated drafting computers running 2d line drawing programs to the current modern cheap, very powerful PC with full 3d solid modeling capabilities. At the same time the use of analysis codes has developed to the extent that a great many of the old trial and error / experimental approaches have now been replaced by numerical simulations. Now designers wish to automate much of the drudgery involved in using these tools so as to be free to work more creatively with the aim of achieving improved designs in shorter timescales. This is leading to the desire for Problem Solving Environments (PSE's)[1] that provide for "all the computational facilities necessary to solve a target class of problems".

To be of greatest benefit, such computing environments need to be re-configurable by design staff. Designers need environments that can be tailored to the problems in hand and to suit the needs of the team working on them. These will make use of many of the existing analysis codes but will focus on systems that allow designs to be manipulated and evolved as well as simply evaluated for performance. Design PSE's will span across companies and also geographically disparate locations[2]. They will increasingly call on parallel grid based computing and use advanced interfaces such as virtual reality systems. They will not, however, replace designers – rather they will form a key part of the support infrastructure.

2 PSE COMPONENTS

A key part of any design PSE will be a tool-set that allows the users to specify goals that the system attempts to meet by modifying the design. Optimization codes offer this kind of capability but historically they have been seen as rather specialist approaches that are only used by experts in key areas, rather than as the every day tools of all designers. There are a number of reasons for this lack of take-up by the broader design community:

- Designers have not always trusted their analysis codes sufficiently to let the results from these codes drive design decisions directly;
- Analysis codes have often been too computationally expensive to allow realistic searches to be carried out;
- Search methods have not been robust or easy to use;
- Search methods have not allowed designers to feed their experience and historical data into the search process;
- Design goals can often not be clearly stated or involve many competing objectives and constraints;
- Design often takes place at a level of abstraction that is not suitable for the direct application of full-scale analysis and more approximate tools are either unavailable or not trusted.

This list is by no means exhaustive but it does begin to set out an agenda for those who would like to build design PSE's, even in relatively narrow, single discipline areas such as wing aerodynamics or turbine blade structural dynamics. When the desire is for integrated multi-domain systems a whole range of additional problems begin to arise:

- What parameters and software are used to represent the design in these varied domains;
- Which domain should take precedence when there are conflicting requirements and in what order should individual domains be considered or can they be tackled in parallel;
- What software standards should be adhered to and what computing platforms will be used;
- Who is responsible for the integration and software engineering of such inevitably large-scale distributed systems and who will fund their development.

Inevitably, many of these issues will only be resolved over time, but it is already clear that web based technologies will play a crucial role: the use of XML, Java, Globus and CORBA all seem to lay at the heart of this kind of world. It also seems sensible to assume that the de-facto standard interface to all computing environments will be some form of graphically advanced (VR enabled) web browser running various plug-ins and visualization capabilities: there is simply too much effort being put into these products by very powerful companies to expect that more specialist software developers will have anything but a minor influence on the outcome (a Microsoft Word attachment is now the world's standard means of transmitting documents – ten years ago there were a whole range of competing word-processing packages).

3 SOME PROBLEMS IN AEROSPACE DESIGN

It is, of course, quite normal for design teams to be forever seeking improved designs in shorter tiemscales. Given suitable analysis capabilities optimizers can help acheive such goal oriented tasks. There remain, however, a number of problems that must still be dealt with in equiping designers with tools that meet these needs. Perhaps inevitably these are all connected with managing computational effort.

3.1 Multi-modal design spaces

Although optimization and search methods are routinely used in the concept design stages for most aerospace products, they have mostly been based on gradient descent methods. Such tools are unable to deal with highly multi-modal design spaces such as those found in structural dynamics problems[3]. The migration to evolutionary tools with their inherent multi-modal capabilities is now underway. Their adoption is, however, somewhat hampered by the lack of familiarity that most design staff have with such methods. Perhaps more importantly, evolutionary methods can be somewhat profligate in their use of calls to analysis codes. As concept teams increasingly wish to adopt more sophisticated approaches during analysis this aspect of evolutionary methods needs dealing with.

One obvious way of coping with the need for very many analysis calls is to adopt parallel processing schemes: population based search methods naturally fit such approaches. Even so, spreading thousands of analysis calls over hundreds of distributed processors of probably heterogeneous types is by no means simple. Two key aspects must be dealt with: first some kind of job scheduling and load-management software must be available and secondly care must be taken to deal with jobs that fail to complete either because the analysis fails or the processor being used fails. This latter aspect has recieved relatively little attention until recently, as modern computers have become astonishingly reliable. Even so, given very large compute clusters using commodity systems it is now quite normal to experience some job failures during a run. These must be dectected and suitable steps taken. Fortunately there are a number of systems on the market that address this issue and parallel evolutionary computing is now increasingly being exploited. Figure 1 shows before and after illustrations of a satellite boom optimized for enhanced vibration performance in this way. This search required over 140,000 function evaluations for convergence and was carried out using 20 processors over an extended period, allowing for failing jobs as and when they occured[4].

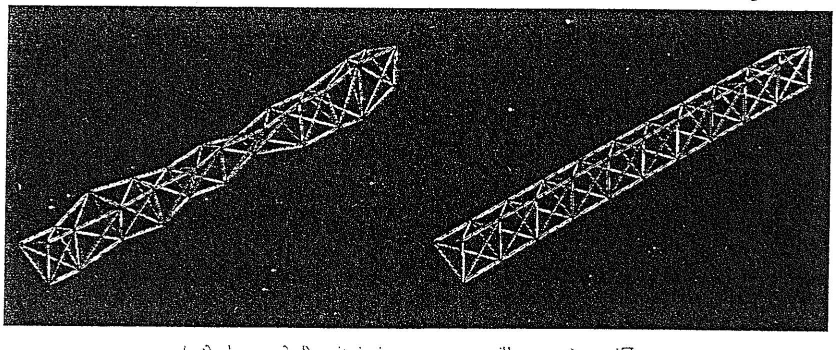


Figure 1 – satellite structure optimization (before and after).

3.2 Multi-level searches

Another way of dealing with expensive function calls is to adopt multi-level approaches, where the degree of sophistication of the search method can be varied as the design space is explored. Figure 2 illustrates the three levels of analysis capability in the Southampton multi-level wing design systems that has been developed in collaboration with BAE SYSTEMS[5]. In this system the importantly, evolutionary methods can be somewhat profligate in their use of calls to analysis codes. As concept teams increasingly wish to adopt more sophisticated approaches during analysis this aspect of evolutionary methods needs dealing with.

One obvious way of coping with the need for very many analysis calls is to adopt parallel processing schemes: population based search methods naturally fit such approaches. Even so, spreading thousands of analysis calls over hundreds of distributed processors of probably heterogeneous types is by no means simple. Two key aspects must be dealt with: first some kind of job scheduling and load-management software must be available and secondly care must be taken to deal with jobs that failt to complete either because the analysis fails or the processor being used fails. This latter aspect has recieved relatively little attention until recently, as modern computers have become astonishingly reliable. Even so, given very large compute clusters using commodity systems it is now quite normal to experience some job failures during a run. These must be dectected and suitable steps taken. Fortunately there are a number of systems on the market that address this issue and parallel evolutionary computing is now increasingly being exploited. Figure 1 shows before and after illustrations of a satellite boom optimized for enhanced vibration performance in this way. This search required over 140,000 function evaluations for convergence and was carried out using 20 processors over an extended period, allowing for failing jobs as and when they occured[4].

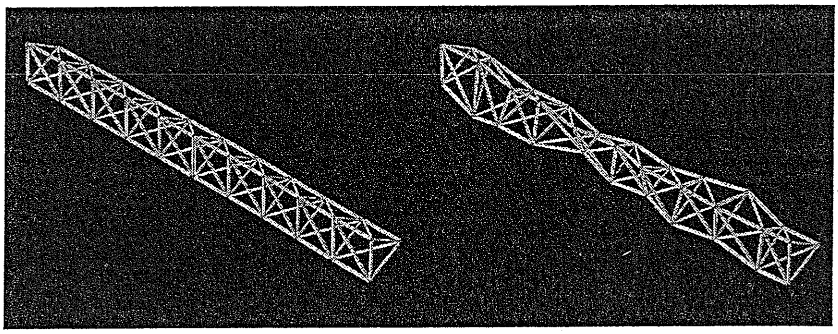
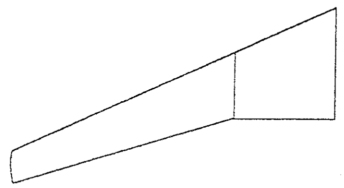


Figure 1 – satellite structure optimiztion (before and after).

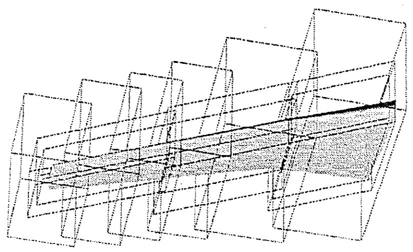
3.2 Multi-level searches

Another way of dealing with expensive function calls is to adopt multi-level approaches, where the degree of sophistication of the search method can be varied as the design space is explored. Figure 2 illustrates the three levels of analysis capability in the Southampton multi-level wing design systems that has been developed in collaboration with BAE SYSTEMS[5]. In this system the

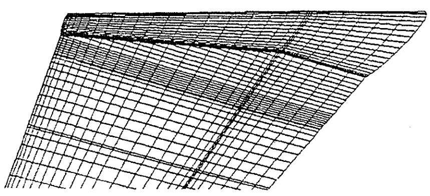
TADPOLE program, which is based on empirical models[6], can be used with large evolutionary searches to explore design spaces before moving the resulting populations on to study by the more expensive VSAERO[7] and the MGAERO[8] codes. Results from using this system (Table 1) show the benefits of multi-level searches over direct approaches, even direct evolutionary searches.



TADPOLE: 50 milliseconds



MGAERO: 3 hours



VSAERO: 5 minutes

Figure 2 – three drag analysis levels.

3.3 Response Surface Methods

The final approach considered here for improving search speeds is based on response surface methods[9]. Such methods aim to build surrogates to the true search space that can be studied in lieu of the original expensive code. These are built with as few calls as possible to the full model code and are, essentially, an automated "black box" approach to building emprical models. Of course, since they do not draw on the designer's knowledge of the search space directly they are never likely to be as accurate or robust as codes like TADPOLE. They can, however, be built and tuned automatically. Figure 3 illustrates the components of a search system designed to construct such a response surface model (RSM) automatically. Note that in this approach evolutionary optimizations are used both to search the RSM space and to guide the process involved. Thus the design of experiments analysis[10] needed to chose the intial points to be studied routinely requires optimization if suitable space filling designs are to be adopted and this can be effectively accomplished using evolutionary methods. Also, the RSM used has a number of meta-parameters that must be optimally selected to control the curve fit placed through the data. This tuning process deals with control of curvature and regression if Kriging is used to model the data[9].

Search and analysis method	TADPOLE D/q	VSAERO	MGAERO
Initial Design	3.12 m^2	3.1 m^2	2.85 m^2
GA search, 2000 VSAERO evaluations	3.06 m^2	2.59 m^2	2.95 m^2
GA search, 10000 TADPOLE evaluations followed by 2000 VSAERO evaluations	2.94 m^2	2.47 m^2	3.06 m^2
GA search, 10000 TADPOLE evaluations followed by 1000 MGAERO evaluations	2.75 m	2.87 m^2	2.37 m^2
GA search, 1000 MGAERO evaluations	3.02 m^2	3.28 m^2	2.54 m^2

Table 1 – drag results of multi-level searches

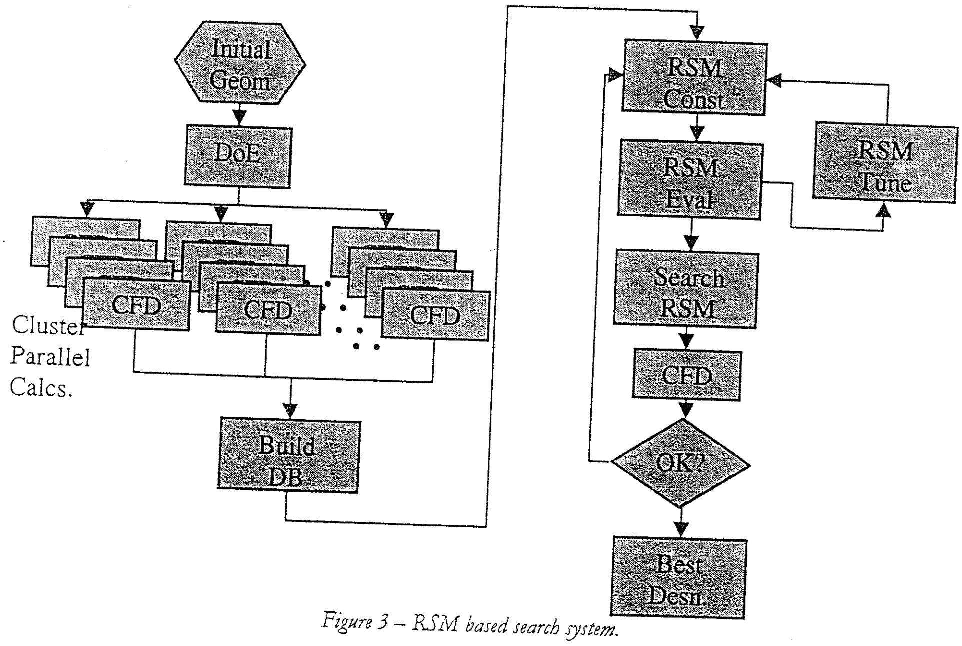


Figure 3 – RSM based search system.

Adopting this approach has allowed optimization of 3D geometries using the sz02Reynolds averaged Navier Stokes code and FAITH design system supplied by Rolls-Royce[11] (whose support is gratefully acknowledged). Typical jobs have five hour run times using this system. Figure 4 shows before and after geometries where secondary kinetic energy in the flow past a gas turbine guide vane has been reduced using 21 design variables while satisfying a tight capacity constraint.

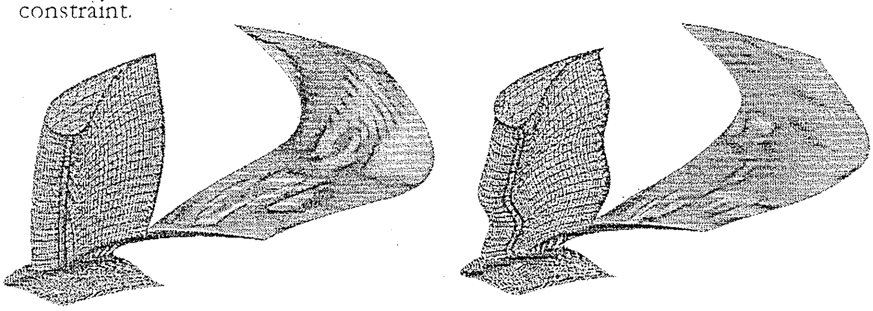


Figure 4 – before and after guide vane geometry and flow field

4 CLOSING REMARKS

In the future, it seems certain that the PC or its derivatives running a form of Microsoft Windows or perhaps Linux will be the computing platform of choice – price / performance issues now make such platforms nearly an order of magnitude more cost effective than specialist high performance computing systems. Most designers will not use super-computers: rather they will have access to hundreds of networked PC's, operated by their own or collaborating companies using various GRID like protocols.

In this world it also increasingly likely that the varied pieces of software currently used by designers will not be installed as packages on the user's machine. Rather these tools will be available as services over secure networks when the users need them – moreover, markets will form in the provision of these services (hit the search button on your browser now and you are offered web search services provided by a number of competing companies).

Thus it is reasonable to assume that a design PSE will consist of tools that the design team selects from the net and plugs together to tackle the current problem with perhaps only the browser and the software holding the description of the product under consideration being held on a locally mounted package / machine. So, when stress analysis is needed, say, the designer couples the in-house CAD data-base to a commercial FEA service, sets up the boundary conditions and asks for a solve. This solve may well be carried out on third party machines or be distributed over a disparate and rather fuzzy collection of nodes. The user may select the solver to invoke based on company policy, an understanding of a particular solvers technical merits or simply on price or speed. Given such an approach it seems sensible to aim to provide design optimization services in an entirely analogous fashion. The user couples the CAD model and (remote) analysis code(s) together by drag and drop within their web browser, indicates the parameters that may be considered for change and any constraints that must be met, along with targets to be achieved and the resource budget available (probably specified in dollars). The optimization service provider must then meet this requirement by judicious calls to the solver(s) in a timely and cost-effective manner. Such services will no doubt include many evolutionary methods.

5 REFERENCES

[1] E. Gallopoulos, E. Houstis and J. Rice. "Future Research Directions in Problem Solving Environments for Computational Science", Tech. Rep. 1259, Cen. for Supercomputing R&D, Uni. Illinois Urbana-Champaign(1991).

[2] A.J. Keane & P.B. Nair, "Problem Solving Environments in Aerospace Design", *J. of Advances in Engineering Software*, 32(6) pp. 477-487 (2001).

[3] A. J. Keane, "Passive vibration control via unusual geometries", *J. Sound Vib.* 185(3) pp. 441-453 (1995).

[4] A. J. Keane and S. M. Brown, "The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques", pp. 107-113 in Proc. Conf. on Adaptive Comp. in Eng. Design and Control 96, ed. I. C. Parmee,P.E.D.C., Plymouth (1996).

[5] A. J. Keane & N. Petruzzelli, "Aircraft wing design using GA-based multi-level strategies", pp. A00-40171, AIAA-2000-4937, Proc. 8th AIAA/USAF /NASSA/ISSMO Sym. on Multidisc. Analysis & Opt., Long Beach (2000).

[6] J. Cousin and M. Metcalfe, "The BAE Ltd transport aircraft synthesis and optimization program", AIAA 90-3295, September 1990.

[7] B. Maskew, "Prediction of subsonic aerodynamic characteristics: A case for low-order panel methods", *J. Aircraft*, **19**, No. 2, Feb. 1982.

[8] B. Epstein, A. Luntz and A. Nachson, "Multigrid Euler solver about aircraft configurations, with cartesian grids and local refinement", AIAA 89-1960, 1989.

[9] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimisation of expensive black-box functions", *J. Global Optimisation* 13(4) pp. 455-492 (1998).

[10] J. Sacks, W. J. Welch, T. J. Mitchell & H. P. Wynn, "Design and analysis of computer experiments (with discussion)", *Statistical Science* 4 pp. 409-435 (1989).

[11] S. Shahpar, "Three-Dimensional Design and Optimization of Turbomachinery Blades using the Navier-Stokes Equation", paper 2001-1053, 15th ISABE, Ed.P.J. Waltrup, Bangalore, (2001).