7TH INT. CONF. ON ARTIFICIAL INTELLIGENCE IN DESIGN (AID '02)
CD ROM, JULY, 2002, CAMBRIDGE, U.K.

AN AUTOMATED OPTIMIZATION SYSTEM FOR AIRCRAFT WING DESIGN

Y.S. ONG AND A.J. KEANE
Computational Engineering and Design Centre
School of Engineering Sciences, University of Southampton,
Highfield, Southampton SO17 1BJ, U.K.

Abstract. The optimization of a transonic civil transport aircraft is a complex and difficult task, due to the complexity of the cost surfaces and the human expertise that is necessary in order to achieve high quality results. In this paper we describe an Automated Optimization System that provide recommendations to design engineers on the choice of optimization search technique to use, especially when searching within familiar domains. The Automated Optimization System uses Artificial Intelligence, specifically machine learning techniques to perform knowledge discovery from past optimization searches and reuses this knowledge to facilitate intelligent recommendations for search routines selection. Results of a case study on the design of a transonic civil transport aircraft wing using the Automated Optimization System are presented in the paper. It is shown that the Automated Optimization System not only aids design engineers to make improved decisions when working on complex aircraft wing but also helps improve design search performance.

1. Introduction

Optimization is a mature technology that has been studied extensively by researchers over the last half century. Although available for many years it has only recently been heavily used by the design community (Keane and Nair, 1999). This take up is now happening because increases in computing power allow increasingly accurate analysis codes to be deployed in this way, see for example the work reported in (Jameson, 1999) in a recent theme issue dedicated to optimization. With a long history of research and development, optimization methods have evolved considerably and many algorithms and implementations are now available. Generally, they can be classified into two broad categories: conventional numerical optimization methods (mostly gradient based) and stochastic optimization methods.

Typical conventional methods are CFSQP; Approx; Simplex; Hooke&Jeeves and others (Lawrence and Tits, 1996; Schwefel, 1995; Siddall, 1982). Among the modern stochastic optimizers are Genetic Algorithms (GA); Simulated Annealing (SA); Evolutionary Programming (EP) and Evolution Strategies (ES) (Yin & Germany, 1993; Kirkpatrick *et al.*, 1983). Gradient-based methods have the known advantage of their efficiency; however, they are very sensitive to starting point selection and are more likely to stop at non-global optima than modern stochastic algorithms. Stochastic techniques on the other hand produce new design points that do not use information about the local slope of the objective function and thus are not prone to stalling in false optima. They do tend to require more analysis effort, however.

It remains the case, however, that much complex engineering design exploration is still carried out manually. The design engineer use computer-aided-design tools to visualize and modify designs and evaluate their performance, providing information about their merits and limitations by numerical simulations. He then enters a design-evaluate-redesign process and stops when he thinks that the design is adequate based on his experience and knowledge of past designs. Engineering design optimization helps reduces the cycle time for these design iteration loops and often finds better designs by computerizing parts of this iterative process. In general, given a set of design variables *D*, a set of bounds and constraints *B* and *C*, constrained design optimization is the problem of determining values of *D* to minimize or maximize an objective function *F(D)*, subjected to *B* and *C*.

In practice, however, unless one knows which optimization methods most suits the design problem in hand, the optimization may not perform properly or achieve truly optimum design. Some methods might not even be capable of producing a feasible design on some problems. Sandgren (1997) applied 35 nonlinear optimization algorithms to 30 engineering design optimization problems and compared their performance. Bramlette and Cusic (1989) also compared the application and performance of different methods including gradient based numerical optimization to the design and manufacture of aeronautical systems. The applicability of different conventional numerical optimization methods to aircraft design has been further explored by Sobieszczanski-Sobieski & Haftka (1996) and Vanderplaats (1984). The general conclusion obtained from all these studies is that no single optimization search technique always performs well in all optimization problems.

Nevertheless, it remains common to find design engineers relying very much on their intuition, experience and knowledge of the design domain when making a choice of optimization method to employ whenever a design search is conducted. The effect of this is that the design quality is heavily dependent on the experiences and knowledge of the design engineer and it may lead to non-optimal designs being produced at high cost due to the limited experience of novice or inexperienced designers (i.e., in reality, designers often stick to a very limited range of optimization techniques regardless of the design problem involved or the sophistication of any optimization methods suite available). It may also have a detrimental effect on design innovation by placing too much dependence on a single individual's past designs, which usually contain biases.

Few studies in the literature have directly addressed the problem of choice of optimization search routines for engineering designs. Nevertheless, the problem that we are considering shows some resemblance to those of the Problem Solving Community (Dyksen and Gritter, 1992; and Houstis *et al.*, 2001). Generally, they try to map scientific software to various classes of problems that are represented by partial differential equations (PDEs). For example, in Houstis *et al.*, 2001; based on the characteristics of PDE models, recommendations are made from the choices of numerous scientific software approaches available. In contrast, engineering design optimization cannot be easily set in these terms, as the design problems (objective functions) are not often represented directly by PDEs.

Although the system developed for mapping PDE models to scientific software is not directly applicable here, the implementation approaches presented have provided resources that have aid in the development of an Automated Optimization System. This paper thus describes the Automated Optimization System developed for tackling this problem of choice of optimization search methods in complex engineering design optimization. This involves the use of Artificial Intelligence, specifically Machine Learning techniques, to learn about the merits and limitations of different search methods on a design domain, such that recommendations of appropriate methods could be made to search on future design problems within given domain. The design of the transonic civil transport aircraft wings using the Automated Optimization System is presented in this paper. It is shown that the system can help reduce the reliance on optimization domain experts by ensuring that the minimum of knowledge on optimization techniques is required from design engineers when performing search activities. At the same time, it improves the performance of design engineering optimization.

2. Automated Optimization System (AOS)

Companies usually have limited diversity of trade and thus work-scope. For example, BAE (Airbus) focuses mainly on aircraft design; Rolls-Royce on engine design; while a ship building company focuses on ship design. Depending on the complexity of a domain, some optimization techniques that may have proven to be useful in one domain might not work so well in other domains. The same reasoning applies to individual design problems within a domain. This fundamental observation is the reason why it is important to have an Automated Optimization System (AOS) that makes recommendations to designers whenever a design-space search is to be initiated on a design problem. The AOS described in this paper is generally made up of three major components; namely the 1) Optimization Engine 2) Domain Knowledge Discovery and 3) Search Method Advisor. Here, we briefly describe each of these components.

2.1. OPTIMIZATION ENGINE

The Optimization Engine is basically an optimization software package that contains multiple sophisticated optimization and exploration techniques for design-space search. The Optimization Engine used in the AOS is the one described in (Keane, 1995), and known as OPTIONS. OPTIONS is a design exploration and optimization package that may be used to study and compare a large range of optimization methods when applied to design problems. The user provides routines describing his or her problem plus entries in a problem-specific database. It is then possible to manipulate the design manually, systematically map out the effects of design changes, or, having specified design variables, constraints and an objective function, invoke one of the many optimizers within the package. Among the many different optimization search routines in OPTIONS, some are from standard libraries (Schwefel, 1995 and Siddall, 1982), while others have been specially developed for the suite, based on ideas culled from the literature. OPTIONS Optimization Engine currently contains the following 30 optimization routine implementations:

- The Davidon-Fletcher-Powell strategy (David) by Siddall (1982);
- Fletcher's 1972 method (Fletch) by Siddall (1982);
- Jacobson and Oksman Method (Jo) by Siddall (1982);
- Powell direct search method (PDS) by Siddall (1982);
- Hooke and Jeeves direct search (Seek) by Siddall (1982);
- The simplex strategy (Simplx) of Nelder & Meade by Siddall (1982);
- The method of successive linear approximation (Approx) by Siddall (1982);

- Adaptive random search (Adrans) by Siddall (1982);
- A bit climbing algorithm (BClimb) (Davis, 1991);
- A dynamic hill-climbing algorithm (DHClimb) (Yu & Maza, 1993);
- A population-based incremental learning algorithm (PBIL) (Baluja, 1994);
- The Powell routine as implemented in the Numerical Recipes cookbook (Num Rcp) (Press *et al.*, 1986);
- A design of experiments based optimizer using either pure random numbers or pseudo random sequences (DoE) (Statnikov& Matusov, 1995);
- Repeated application of a one-dimensional Fibonacci search (Fibonacci) by Schwefel (1995);
- Repeated application of a one-dimensional Golden section search (Golden Sect) by Schwefel (1995);
- Repeated application of a one-dimensional Lagrangian interpolation search (Lagrange Int) by Schwefel (1995);
- Hooke and Jeeves direct search (Hooke&Jeeves) by Schwefel (1995);
- Rosenbrock's rotating co-ordinated search (Rosenbrock) (Rosenbrock, 1960) by Schwefel (1995);
- The strategy of Davis, Swan and Campey, with Gram-Schmidt orthogonalization (DSCG) by Schwefel (1995);
- The strategy of Davis, Swan and Campey with Palmer orthogonalizational (DSCP) by Schwefel (1995);
- Powell's strategy of conjugate directions (Powell) by Schwefel (1995);
- The Davidon-Fletcher-Powell strategy (DFPS) by Schwefel (1995);
- The simplex strategy (Simplex) of Nelder & Meade by Schwefel (1995);
- The complex strategy (Complex) by Schwefel (1995);
- Schwefel's two-membered evolution strategy (2MES) by Schwefel (1995);
- Schwefel's multi-membered evolution strategy (MMES) by Schwefel (1995);
- A genetic algorithm based on clustering and sharing (GA) (Yin & Germay, 1993);
- Simulated annealing (SA) (Kirkpatrick *et al.*, 1983);
- Evolutionary programming (EP) (Fogel, 1993);
- An evolution strategy based on the earlier work of Bäck *et al* (ES) (Back *et al.*, 1991).

2.2. DOMAIN KNOWLEDGE DISCOVERY

The Automated Optimization System makes use of a Domain Knowledge Base that contains information about the merits and limitations of each search methods on particular design domains. This form of knowledge is often specialized and applicable only to a single domain.

2.2.1. Data Acquisition

The domain knowledge can be derived from the mass of data that exists implicitly in past designs. Throughout the entire engineering design processes, designer engineers often produce a great deal of data as a result of their design-evaluate-redesign actions. These data have often been discarded, and thus represents a potential source of useful information from which knowledge may be extracted for use in future design activities. The other form of data considered in this paper is obtained via offline simulations conducted on a range of design problems sampled from the design domains that are of interest. The Latin Hypercube method is used to generate the sampled design problems so as to provide a good representation of the domain of interest.

In engineering design, it is common for design problems from a single domain to be uniquely identified by a set of key design parameters. For example in the aircraft wing domain, this set of parameters may include cruise height, mach number and fuel weight[1]. These parameters that characterize each problem are termed the 'Domain Problem Descriptors' or DPD. Here it is assumed that the DPDs vary over some predefined ranges specified by the design engineers. Offline simulations are then conducted over the set of sampled design problems by performing searches on them using every optimization search routine available in the Optimization engine. The data generated by the simulations is used to define the merits and limitations of the optimization search routines to the domain.

2.2.2. Machine Learning Techniques for Knowledge Discovery

Here machine learning techniques are used to carry out knowledge discovery on the data archived from past designs. Machine learning is a category of Artificial Intelligence that is particularly suitable for use in the Automated Optimization System because it is able to automate the process of generalizing past design data on the applicability of optimization search methods to different subsets of problems within a design domain. To extract knowledge about the merits and limitations of the many optimization search methods on a design domain, using machine learning, it is necessary to pre-process the archives from past design studies before they can be use. This

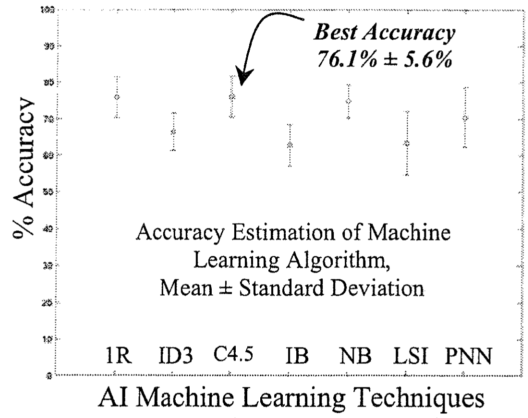
[1] Note that here cruise height, mach number and fuel weight are not design variables to be optimized but rather considered the fixed parameters of a design problem and vary uniquely for different problems within the aircraft wing design domain.

involves the conversion of the original data sources into a table-like dataset, such that the sampled design problems are labeled and ranked according to the optimization search method that performs the 'best'. Machine learning is then employed as a classifier on these pre-processed datasets. It aims to generalize or learn as much as possible from the dataset, so as to accurately identify clusters of design problems from the sampled set that belong to the same class. The resulting classes represent the search methods that were observed to be 'best', among all those available in the Optimization engine. If learning is successful, extrapolating from these data should be possible to aid in future design sessions by successfully recommending the most appropriate optimization technique that best matches a new design problem to be searched.

Note that learning from past designs through feature extraction and indexing is, of course, nothing novel. It is just an analogue to designers' actions in deriving qualitative rules from their experiences on past designs and using these rules to aid them in making decisions for new designs. However, the novelty here lies in our attempt to model and improve this learning process using machine over human learning.

2.2.3. A Brief Survey of Machine Learning Models

A brief survey of the many machine learning models described in Aha, 1992; Chen *et al.*, 1991; Deerwester *et al.*, 1990; Holte, 1993; Langley and Sage, 1994 and Quinlan, 1993 has been performed to identify a suitable learning model for this application (see figure 1 for a lists of the different learning models investigated here). These models were evaluated according to their accuracy estimation, standard deviation and transparency. The algorithms that have appeared to be the most competitive in this application are C4.5, Naïve Bayes and Probabilistic Neural Network (Quinlan, 1993; Langley and Sage, 1994; Chen *et al.*, 1991). For example, figure 1 reports the percentage accuracy estimation and standard deviation of each machine learning models when applied on the aircraft wing design domain (the details of this domain are presented in section 3). Although most of the machine learning techniques considered here allow manual tuning, this has been considered to be too time-consuming and computationally expensive. Among the techniques considered, the use of decision tree inductive learning model such as C4.5 (Quinlan, 1993) is preferred because they produce reasonable classification accuracy at relatively low cost but more importantly, because they posses the ability to generate trees or rules that provide the transparency we seek to give to designers. Designers often lack great expertise in the use of optimization methods and therefore have little confidence that extensive computational runs can produce worthwhile results as opposed to just burning up compute cycles. Therefore, when recommending optimization search methods for design optimization, it is important for the decision-making process to provide some transparency. Of the many machine learning models available, knowledge derived in the form of rules using decision tree models seems to satisfy this human-centered criterion the most. Besides, human specialists can validate these machine-generated rules and also use them to enhance the domain and optimization knowledge of less experienced designers.



LEGENDS	
1R	Simple Classifier (Holte, 1993)
ID3	Decision Trees I (Quinlan, 1993)
C4.5	Decision Trees II (Quinlan, 1993)
IB	Nearest-Neighbor (Instance-based) (Aha, 1992)
NB	Probabilistic (Naïve-Bayes) (Langley *et al*, 1994)
LSI	Information Retrieval (Latent Semantic Indexing) (Deerwester *et al*, 1990)
PNN	Neural Network (Probabilistic Neural Network) (Chen *et al*, 1991)

Figure 1: Accuracy Estimation of the different machine learning algorithms when tested over the Aircraft Wing Design Domain. The machine learning algorithms investigated are also listed.

2.3. SEARCH METHOD ADVISOR

The rules derived by the Domain Knowledge Discovery component are directly incorporated into the Search Method Advisor. When presented with a new design problem, this rule-based component actively advises on the choice of search method using past experiences.

3. Demonstration of Automated Optimization System On The Domain of Transonic Civil Transport Aircraft Wing Design

In this section, the use of the Automated Optimization System in the design of transonic civil transport aircraft wings (Keane and Petruzzelli 2000) is presented.

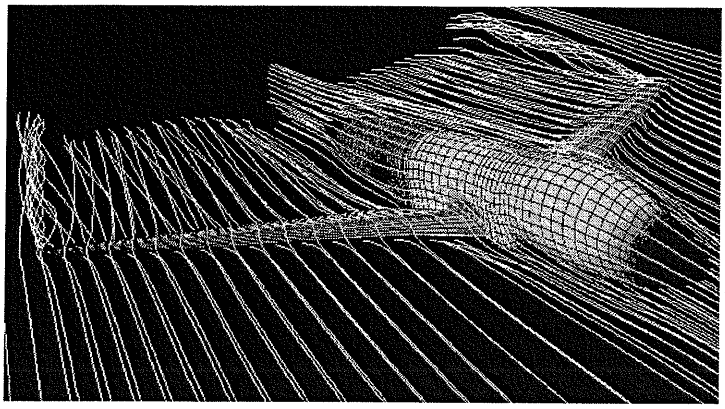
The design of the wings for a transonic civil transport aircraft is an extremely complex task. It is normally undertaken over an extended time period and at a variety of levels of complexity. Typically, simple empirical models are use at the earliest stages of concept design, followed by ever more complex methods as the design process proceeds towards the final detailed stages. The parameters used to describe the wing design problem considered here consist of the free-stream velocity and coefficient of lift of the wing together with a small number of overall wing geometry variables. The geometry is characterized by the plan-form shape of the wing together with several span-wise functions such as twist and thickness to chord ratio. These are represented by eleven parameters (i.e., eleven optimization design variables). In order to prevent the search from driving the designs to unworkable extremes, several constraints are placed on the wings designed. These are the under-carriage bay length (which must be accommodated within the root to kink section of the wing), the fuel tank volume (which must be accommodated between the main spars within the wing), the wing weight and the pitch-up margin. A typical geometric view of such an aircraft with streamlines and its wing design variables are shown in figure 2. The goal to this problem is to design a wing with minimal drag based on empirical models.

3.1. AIRCRAFT WING DOMAIN KNOWLEDGE

To use the Automated Optimization System for wing design, the construction of a Knowledge Base for the domain is essential. To do this, a set of 729 design problems is first sampled using the Latin hypercube method, each defined by a cruise height, mach number and fuel weight fraction, bounded between 7500 ~ 12,000 meters, 0.1 ~ 0.85, and 0.2 ~ 0.5, respectively[2]. The C4.5 induction algorithm is then used to extract rules from two-thirds of the processed dataset obtained via offline simulations. Table 1 shows a portion of the processed dataset used by the C4.5 algorithm for domain knowledge discovery and extraction. The remaining one-third of

[2] The ranges of cruise height, mach number and fuel weight fraction were obtained from design engineers working within the aircraft wing design domain.

the sampled design problems form the validation set, which is later used to validate the system.



11 Wing Design Variable Definitions	
Wing Area	Root thickness/chord
Aspect Ratio	Kink thickness/chord
Leading edge sweep	Tip thickness/chord
Inner panel taper ratio	Wash out at tip
Outer panel taper ratio	Fraction of tip wash-out at kink
Trailing edge kink position	

Four Design Constraints
Under-Carriage bay length
Wing weight
Fuel tank volume
Pitch-up margin

Figure 2: Geometric view of streamlines over a Transonic Civil Transport Aircraft and its wing design variables and constraints.

Table 1: A portion of the Processed Dataset constructed from past aircraft wing design activities and offline simulations.

Domain Problem Descriptors of Aircraft Wing Design Problem			'Best' Optimization Technique
F-Mach (0.1 ~ 0.85)	*Fuel-Fraction (0.2 ~ 0.5)*	*Height (7500 ~ 12,000)*	
0.35	0.225	8625.0	Approx
0.725	0.425	10125.0	PDS
0.10	0.450	9750.0	Powell
	...		
0.82	0.45	7500.0	Fletch
0.75	0.275	8250.0	2MES
0.65	0.175	10075.0	Lagrange Int

The average design search performance (i.e., search efficiency and quality) of each search routine in the aircraft wing design domain is also summarized in figure 3, with the values normalized to unity[3]. Even though there are 30 optimization search routines available in the OPTIONS system, only six were found to figure as 'best', within the design domain. The six optimization techniques identified were Approx, PDS, Powell, Fletch, Lagrange Int and 2MES. Each abbreviation represents an optimization search routine described previously in section 2.1.

Figure 4 shows the six rules found for the aircraft wing design domain, by the C4.5 machine learning induction algorithm. This knowledge can be used to aid designer engineers in future design search activities, by recommending search methods that would perform well when used to optimize a new aircraft wing design problem.

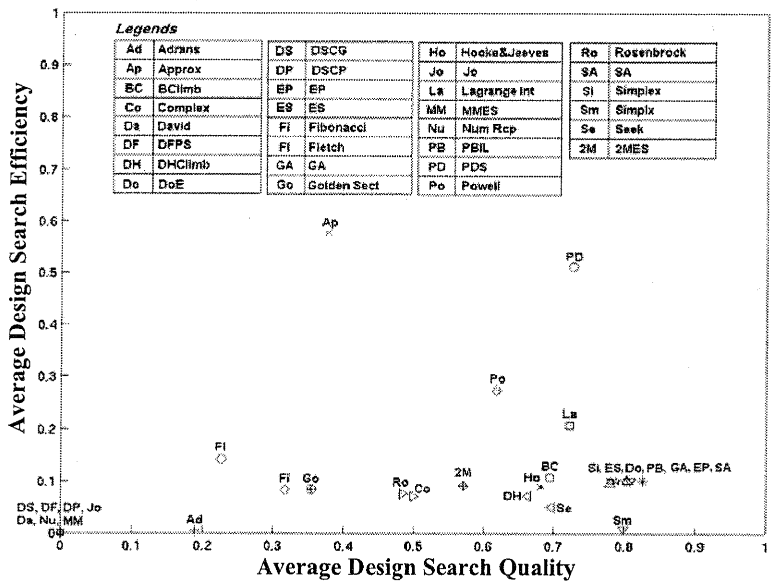


Figure 3: A Plot of Average Design Search Efficiency against Average Design Search Quality for each Search Method used in the Aircraft Wing Design Domain. The performances are normalized to unity.

[3] Here, 1 or unity represents the best possible performance a search method can achieve.

Rule 1:
{ F_Mach <= 0.1625 } *OR*
{ (F_Mach <= 0.225) *AND* (Height > 11250) } *OR*
{ (0.6625 >= F_Mach > 0.6) *AND* (Fuel_Frac > 0.45) *AND* (10125 >= Height > 9000) } *OR*
{ (0.6625 => F_Mach > 0.6) *AND* (Fuel_Frac <= 0.3) *AND* (Height > 10500) } *OR*
➔ Powell
Rule 2:
{ 0.6 >= F_Mach > 0.225 } *OR*
{ (0.6 >= F_Mach > 0.1625) *AND* (Fuel_Frac <= 0.45) *AND* (Height <= 11250) } *OR*
{ (0.6 >= F_Mach > 0.1625) *AND* (Height <= 9000) } *OR*
➔ Approx
Rule 3:
{ 0.7875 >= F_Mach > 0.6625 } *OR*
{ (0.225 >= F_Mach > 0.1625) *AND* (Fuel_Frac > 0.45) *AND* (11250 >= Height > 9000) } *OR*
{ (0.6625 >= F_Mach > 0.6) *AND* (Fuel_Frac > 0.275) *AND* (9000 >= Height > 7500) } *OR*
{ (F_Mach > 0.6) *AND* (Height > 9375) } *OR*
➔ PDS
Rule 4:
{ (0.6625 >= F_Mach > 0.6) *AND* (Fuel_Frac <= 0.3) *AND* (10500 >= Height > 9750) }
➔ Lagrange Int
Rule 5:
{ (F_Mach > 0.7875) *AND* (Height <= 9375) }
➔ Fletch
Rule 6:
{ (F_Mach > 0.725) *AND* (Fuel_Frac <= 0.375) *AND* (8250 >= Height > 7875) }
➔ 2MES

Figure 4: Knowledge of the Aircraft Wing Design Domain in the form of rules. These are derived by the C4.5 inductive algorithm using the datasets obtained from offline simulations.

3.2. PERFORMANCE OF AUTOMATED OPTIMIZATION SYSTEM FOR CIVIL TRANSPORT AIRCRAFT WING DESIGN

When assessing the Automated Optimization System, it is useful to develop standards for comparison. The following approaches are termed 'Common Designer Strategies' (CDS). These are derived from analogues to designers' behaviors (both novices and optimization experts) when working with design problems using a basic optimization engine (i.e., one that requires design engineers to choose the optimization search methods manually, based on experience). Five potential strategies have been identified as follows:

- The simplest and most obvious strategy adopted by a design engineer (usually a novice) is (CDS1) 'Random Guessing'. This just means randomly choosing optimization routines from those available in the Optimization engine to search on each design problem.

- The second strategy (CDS2) is the notion of always using the same randomly selected optimization routine given that it was able to successfully generate a feasible design the first time it was used in a domain.
- The third strategy (CDS3) is to utilize an optimization method that is thought to be most robust, e.g., the Evolutionary Programming (EP) optimization method is often regarded to be most robust and thus chosen.
- The fourth strategy (CDS4) is an analogue to the optimization method favoritism that may be displayed by a designer. Here the Genetic Algorithm (GA) is chosen to be the designer's favorite.
- The final strategy identified (CDS5), is utilizing a optimization method that has generally been accepted as having the ability to provide a design within the shortest time, e.g., Successive linear approximation (Approx) is often regarded as the fastest available method.

The normalized search performances for the basic optimization engine combined with the various 'Common Designer Strategies' and the Automated Optimization System are summarized in table 2. These performance statistics were estimated based on searches conducted on the validation samples mentioned previously. It may be seen from table 2 that the use of Automated Optimization System results in significant improvement in aircraft wing design search performance.

Table 2: Normalized Performance Measures of the Automated Optimization System based on the remaining 1/3 validation samples, in comparison with the 'Common Designers Strategies' CDS 1-5. N.B. Performance here is an average of the speed of the system and the quality of the results obtained, taken over 243 unseen problems from the domain.

Strategy for Choice of Optimization Search Technique	**Estimated Optimization Search Performance on the Aircraft Wing Design Domain[4]**
CDS1	0.2496
CDS2	0.3155
CDS3	0.3882
CDS4	0.3861
CDS5	0.4443
Automated Optimization System using Machine Learning (C4.5 Induction Method)	0.7191

[4] The nearer the value is to 1, the better is the performances of each strategy in conducting design search in the aircraft wing domain.

4. Conclusions

In this paper an Automated Optimization System for complex engineering design optimization is presented. The system is made up of three major components, the Optimization Engine, Domain Knowledge Discovery and Search Method Advisor. The C4.5 induction algorithm has been used for performing knowledge discovery on data archived from past design studies. It is shown that significant improvement in search performances can be achieved if design searches are conducted using the Automated Optimization System. Besides improvement in design search performance, the system also helps reduce reliance on optimization domain experts by ensuring that minimum knowledge of optimization techniques is required by design engineers when performing design searches, and at the same time, eliminating any human biases that may exists.

Acknowledgements

This work was funded by the University Technology Partnership for Design, which is a collaboration between Rolls-Royce, BAE SYSTEMS and the Universities of Cambridge, Sheffield and Southampton.

References

Aha, D. W.: 1992, Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms, *International Journal of Man-Machine Studies* 36(1), pp. 267-287.

Back, T., Hoffmeister, F. and Schwefel, H. P.: 1991, A Survey of Evolution Strategies, *Proceedings of the 4th International Conference on Genetic Algorithms (ICGA IV)*, editor Belewand, R. K. & Booker, L.B. Morgan Kaufman Publishers, Inc., San Diego, pp. 2-9.

Baluja, S.: 1994, Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, CMU-CS-94-163. Carnegie Mellon University.

Bramlette, M. and Cusic, R.: 1989. A comparative evaluation of search methods applied to parametric design of aircraft. *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, California. Morgan Kaufmann.

Chen, S., Cowan, C.F.N. and Grant, P. M.: 1991, Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, Vol. 2, no. 2, pp. 302-309.

Davis, L.: 1991, Bit-Climbing, Representational Bias, and Test Suite Design. *Proceedings of the 4th International Conference on Genetic Algorithms (ICGA IV),* editor Belew, R. K. and Booker, L. B. Morgan Kaufman Publishers, Inc., San Diego.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R.: 1990, Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, 41(6), pp. 391-407.

Dyksen W.R. and Gritter C.R., 1992, Scientific computing and the algorithm selection problem, *in* Houstis E. N., Rice J. R. and Vichnevetsky R. (eds.), *Intelligent Mathematical Software Systems*, North-Holland, pp. 19-31.

Fogel, D. B.: 1993, Applying evolutionary programming to selected traveling salesman problems. *Cybernetics and Systems*, 24(1) pp. 27-36.

Holte, R. C.: 1993, Very simple classification rules perform well on most commonly used datasets, *Machine Learning*, 11, pp. 63-90.

Houstis E. N., Catlin A. C., Rice J. R., Verykios V. S., Ramakrishnan N. and Houstis C. E., 2001, PYTHIA-II, A Knowledge/Database System for Managing Performance Data and Recommending Scientific Software, electronic citation: http://www.cs.purdue.edu/homes/acc/pythiaPaper.html".

Jameson A.: 1999, Re-Engineering the Design Process Through Computation, *Journal of Aircraft*, 36(1), pp. 36-50.

Keane, A.J.: 1995, The OPTIONS Design Exploration System, electronic citation: http://www.soton.ac.uk/~ajk/options /welcome.html.

Keane A. J. and Petruzzelli N.: 2000, Aircraft wing design using GA-based multi-level strategies, *Proceedings of the 8th AIAA/USAF/NASSA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, A.I.A.A., Long Beach.

Keane A.J. and Nair P.B.: 1999, Problem Solving Environments in Aerospace Design, *Technical Report*, University of Southampton.

Kirkpatrick S., Gelatt C. and Vecchi M.: 1983, Optimization by simulated annealing, *Science*, 220, pp. 671-680.

Langley, P. and Sage, S.: 1994, Induction of selective bayesian classifiers, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, Inc., Seattle, WA, pp. 399-406.

Lawrence C. T. and Tits A. L.: 1996, Nonlinear Equality Constraints in Feasible Sequential Quadratic Programming, *Optimization Methods and Software*. Vol. 6, pp. 265-282.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling W. T.: 1986, Numerical Recipes: the Art of Scientific Computing. Cambridge University Press.

Quinlan J. R.: 1993, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc., Los Altos, California.

Rosenbrock, H. H.: 1960, An automatic method for finding the greatest or least value of a function, *The Computer Journal*, Vol 3, Issue 3, pp. 175-184.

Sandgren E.: 1977, The utility of nonlinear programming algorithms, Technical report, Purdue University, Ph.D. Thesis.

Schwefel H.P.: 1995, Evolution and Optimum Seeking, John Wiley&Sons.

Siddall J.N.: 1982, Optimal Engineering Design: Principles and Applications, Marcel Dekker Inc., New York.

Sobieszczanski-Sobieski, J. and Haftka, R. T.: 1996, Multidisciplinary aerospace design optimization: Survey of recent developments, *in 34th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, AIAA-96-0711.

Statnikov, R. B. and Matusov, J. B.: 1995, Multicriteria Optimization and Engineering, Chapman and Hall, ISBN 0-412-99231-0.

Vanderplaats G. N.: 1984, Numerical Optimization Techniques for Engineering Design: With Applications, McGraw-Hill, New York.

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

Yu, D. & Maza, M. de la: 1993, Dynamic Hill Climbing: Overcoming the Limitations of Optimization Techniques, *Proceedings of the 2nd Turkish Symposium on AI and ANN*, pp. 254-260.