Intelligent Data Storage and Retrieval for Design Optimisation – an Overview

C. Peebles, C. Bil, L. Drack

This paper documents the findings of a literature review conducted by the Sir Lawrence Wackett Centre for Aerospace Design Technology at RMIT University. The review investigates aspects of a proposed system for intelligent design optimisation. Such a system would be capable of efficiently storing (and compressing if required) a range of types of design data into an intelligent database. This database would be accessed by the system during subsequent design processes, allowing for search of relevant design data for re-use in later designs, allowing it to become very efficient in reducing the time for later designs as the database grows in size. Extensive research has been performed, in both theoretical aspects of the project, and practical examples of current similar systems. This research covers the areas of database systems, database queries, representation and compression of design data, geometric representation and heuristic methods for design applications.

Keywords: database architectures, design optimisation, configuration control, data access and retrieval.

1 Introduction

Engineering design processes must efficiently incorporate analytical components of varying complexity in multiple disciplines. The computational cost of slow and/or many objective function evaluations can be prohibitive in achieving a meaningful design. Furthermore, the financial cost of lengthy design processes is also a real consideration.

A system is proposed to enable time efficient multidisciplinary design optimisation through the use of a design and state database. Furthermore, the same scheme offers similar benefits generally in simulations that depend on computationally efficient models.

The financial and computational cost of design optimisation can be addressed in two ways. The first is to reduce the computational expense of objective function evaluations, the second is to reduce the number these evaluations required.

The time required for objective function evaluations can be reduced by either selecting analytical components of lesser complexity, which, generally speaking, results in a trade-off between computational speed and accuracy, or by representing these components with a heuristic, empirical or stochastic system. The number of evaluations can be reduced by reusing the designs and states of previous design processes.

A system is proposed that stores designs and states during design optimisation. This would enable the creation of a database that could be accessed in future design optimisations, which, given an efficient search algorithm, would prevent recalculation of data already contained in the database. This situation would improve as size of the database grew. Furthermore, representative systems could be created from the data (for example Neural Networks), which could then be used in time efficient system representation in the optimisation process.

The significant issues of such a system would involve the mechanisms of efficiently storing and retrieving designs and states, data compression, efficient search algorithms and system representations.

A further extension of such a system would be, through the use of heuristics, the ability to switch between various classes of models such as low fidelity tabular data; neural network; various degrees of analytic or computational models and empirical data. This switching would depend on the availability of data in desired states, confidence, computational effort required for retrieval and other comparative measures, such as accuracy required. Formulating efficient and robust heuristics for these algorithms would be an important contribution. End-users of the optimisation and simulations based on the intelligent database would not be expected to be overly concerned with data sources whilst the system is in operation. The intelligent database system is then somewhat like an engineering application of data warehousing.

2 Database generation and use

"Database management systems are now used in almost every computing environment to organise, create and maintain important collections of information" [1]. Database systems are also proving to be a critical method for storing many kinds of data for a multitude of purposes. Databases have evolved from their beginnings in relational, network and hierarchical data models, to such modern and complex data types as Object-Oriented, Spatial and Temporal Data, allowing for vastly expanded roles and capabilities.

Each of these data models is particularly suited to a range of data types, creating specialisation within these databases. As data models have evolved, specific models have gained acceptance for specific tasks, such as "Object-Oriented" for engineering design, "Spatial" for Medical Imaging and Geographic Studies, and "Temporal" for large-scale time-dependent data.

The usefulness of a database is dependent on its applicability to store the data supplied. This is dependent on the data model of the database. Shortcomings can occur, however, when it is unsuitable or infeasible to express data in a manner that allows for storage in a DBMS (Database Management System).

One such shortcoming has been noted in the case of Commercial DBMS. This is due to the fact that Commercial DBMS (CDBMS) are primarily designed for business applications

[2], and therefore some of the important features required in design applications are not available in CDBMS. Some of the deficiencies of CDBMS in design applications have been documented in Refs [2, 3], with the primary deficiencies being that the data models inherent to the majority of CDBMS (Relational) are unsuitable for accurately portraying complex design information.

2.1 Database data models

It is widely believed [4–10] that early database models are unsuitable for design database purposes, as they fail to sufficiently model certain design problems, or are incompatible with incorporated design software. There are, however, a number of beliefs about the best way to improve this situation.

Many traditional database systems, in line with commercial DBMS, used the relational data model as mentioned in the previous section. This model was chosen for simplicity and logical nature of sets and relationships, separation of logical from physical parameters aiding casual users, and simple and precise semantics that reduce programmer burden. However, the main argument against the relational Data Model is its "flatness" of structure, whereby it loses valuable information contained in the relationships of the data. It therefore lacks expressiveness and semantic richness for which semantic Data Models are preferred [11]. It is recognised by many [2, 5, 6, 10–14] that a major requirement of design databases is the ability to accurately and naturally model the data by the designer.

Due to the inherent complexity of Design information, as well as the composite nature of design objects (components built of lesser components), the Object-Oriented Data Model, or data models built on its foundations [5, 10, 12, 13, 15–22], are believed to be suitable to design databases. The Object--Oriented Data Model allows information to be expressed as objects, containing properties and methods (actions to be performed); these objects can be composed of other objects, or combined to create larger objects. As such, Object-Oriented data models allow engineering constructs to be expressed in a logical manner by the designer. *"It is natural for designers to think in terms of the object being designed, the components (i.e. objects) that go into the design, and tools (i.e. operations) for manipulating these objects. A system that directly supports the mapping from the user's mental model to the objects and operations supported by the system will enable design engineers to interact with the system in familiar terms"* [12]. For a detailed survey of Object-Oriented database technologies, examples of existing systems and applicability for design/CAD tasks, and projections for the development of O-O technologies, the reader is referred in more detail to [22].

The very nature of the design process also defines a fundamental requirement of a design database system. *"Design is an iterative process which begins with a general description of the design object and after repeated and possibly alternative refinements terminates when a complete and correct refinement has been reached. A DDBMS must allow template definitions to be refined, must provide facilities to organise the refinements of the templates, and should support the semantics of refinements and alternatives."* [10]

For cases where an efficient Design-centered database is desired, but a relational or similar early Data Model Database already exists, there are methods to translate from the early data model to the more advanced and efficient data model for design. Ahad [23] presents such a method for translation of existing relational databases, in the form of an Object Management System (OMS), which acts to translate the modelling concepts not inherent in the early Data Model into an Object Oriented Framework.

Much of the above discussion has discussed the inadequacies of the Relational Data Model, and advantages or improvements through hybridisation or implementation of different models; however it has been noted [24] that, out of the many desired improvements on the Relational model for a number of different database applications:

- There is a large collection of constructs, each relevant to one or more application-specific environment; and
- The union of these constructs is impossibly complicated to understand and probably infeasible to implement with finite resources.

As such, *"it appears inappropriate to look for a single universal data model which will support all nontraditional applications. In short, what the CAD community wants is different from what the semantic modelling community wants which is different from what the expert database community wants, etc. Consequently, such users should build application-specific data models containing the constructs needed for their own environment … The thrust of a next-generation database system should be to provide a support system that will efficiently simulate these constructs"* [24].

2.2 Storage and very large databases

In the majority of database applications, relatively small data records are handled, therefore there are no problems in I/O accessing of the data, and no time costs for storage and retrieval. This does become a problem however when the size of records increases to the point where the memory performance of the system is no longer capable to perform these operations in a single pass [25]. This is becoming more prevalent with the advent of more complex database data models, particularly in applications for medical imaging, geographics, and CAD/CAM applications, to name a few.

One method to combat this is to compress the database index and contents, thereby reducing the necessary space in memory for the data. This method is discussed in detail in Section 4.

For a database containing large external files, Ramakrishna and Larson [26] present a composite perfect hashing scheme. Perfect hashing is an efficient and popular technique for organizing internal tables and external files, with 'perfect' meaning that the method doesn't result in memory overflows. This scheme guarantees record retrieval in a single disk access, and this method can be used in any application that can afford to store header table in internal memory.

2.3 Concurrent access to databases

A key aspect in the design process lies in the fact that it is a continual process undertaken commonly by a team, as opposed to a single designer. This means that there will arise times where multiple users need access to data within a design database, and if both users working concurrently, allowances have to be made for accurate revision histories and relevant updating of the design record.

Roller et al. [27] present a cooperative transaction model for shared engineering databases, which provides a higher degree of concurrency and process parallelism in CAD. As opposed to traditional transaction models where intra-transaction results are isolated, the presented model allows exchanging and sharing of data and supports the integration of subresults into a common solution. The realization of the transaction system is based on concepts of active and object-oriented database systems.

3 Database search methods

An important measure of the efficiency of a database system lies not only in the efficiency of data storage, but also in the efficiency of database search and retrieval. Database queries allow the user to input given parameters, allowing for a search of the data within the database for similar or matching parameters. For efficient queries, the goal is to find user-specified data from an often very large database efficiently and with an acceptable accuracy.

While query processing and optimisation can be seen as an important part of a database system, and a parameter for judging overall system efficiency, it is still but one such parameter among many others, and as we have seen previously, the weighting of these parameters will differ based on the requirements of any given system. As such, while efficient querying may be vital for a particular system that demands accurate and very fast returns for user queries, there will be other systems for which other parameters such as efficient storage and compression of very large data files is paramount, and hence inefficient querying can be acceptable, or even desirable in this light.

There is a range of extensive surveys of the literature in database querying and query optimisation [28–35]. These surveys cover many aspects of querying for different data models, different query methods, and optimisation techniques for a range of scenarios.

3.1 Query processing vs. data models

It was discussed in the previous section that there is a wide range of data models available in database systems, with each holding advantages and disadvantages for particular applications. Due to different modelling structures, levels of complexity, and indexing, these different data models will have consequences in the operation of query processes.

A particular case of this arises in distributed or particularly federated database systems, which can be composed of numerous smaller databases, each using different data models. In such a case, the query could potentially need to be written and executed separately for each such system. In response to this, Owrang O. et al. [36] have developed a Parallel Database Machine, capable of query translation between different data models. The end user would only need to be proficient in the Data model of the local DBMS. The system is capable of translating simultaneous queries in parallel to some extent, by processing independent subparts of the translation in parallel. In addition, it can easily be expanded to incorporate other data models into the distributed database system.

3D geometries present a specific problem in query searches, based on the problem of formulating efficient indexes within the database by which to govern the search. There have been several papers [32–35, 37–39] surveying the problem of 3D geometric search and offering solutions to this dilemma.

In his 1999 study [39], Keim gives an example of an implementation for searching of databases for 3D geometries. His proposed solution for an efficient similarity search is based on a new geometric structure. Keim states from his study of the field that *"It is widely recognised that 3D similarity search is a difficult problem – by far more difficult than 2D similarity search"* [39]. The most widely used techniques for accessing databases of complex objects are feature-based approaches [40, 41], which are mainly used as a simple filter to restrict the search space.

The main contribution of Keim's paper is a new geometry-based index structure that generalises the well-known R-tree approach for an efficient volume-based similarity search on 3D volume objects. This solution is based on the general concept of using both progressive and conservative approximations. These approximations are used to define a minimum and maximum volume difference measure, which allows an efficient pruning of the search space.

While this implementation has been developed with medical applications in mind, the author recognises the applicability to 3D geometries in CAD and design applications, and states that it is generally applicable to a wide range of other applications.

In a later paper (2003), Funkhouser et al. [33] describe a web-based search engine for 3D geometries, supporting queries based on 3D sketches, 2D sketches, 3D models and/or text keywords. This paper presents a new matching algorithm for shape-based queries, which provides a stated 46–245 % better performance (using five different algorithms for comparison) than related shape-matching methods, and is fast enough to return query results from a repository of 20,000 models in under a second.

In this study, which is principally aimed towards a web--based application, the authors describe novel methods for searching 3D databases using orientation invariant spherical harmonic descriptors. This answers one of the critical areas in 3D database search, in providing an efficient indexing method for 3D geometries through efficient geometric representation. Such a shape descriptor should be:

- Quick to compute;
- Concise to store;
- Easy to index;
- Invariant under similarity transformations;
- Insensitive to noise and other small extra features;
- Independent of 3D model representation, tessellation or genus;
- Robust to arbitrary topological degeneracies; and
- Discriminating of shape differences at many scales.

Unfortunately no existing shape descriptor (at time of this writing in 2003) has all of these properties. The authors therefore propose their novel shape descriptor based on spherical harmonics. The main idea is to decompose a 3D model into a collection of functions defined on concentric spheres and to use spherical harmonics to discard orientation information for each one. This yields a shape descriptor that is both orientationinvariant and descriptive.

This approach yields a significant advantage, in that it can be indexed without registration of 3D models in a canonical coordinate system, for example in the case of similar models, where even minimal dissimilarities can cause a large misalignment of the principle axes, resulting in poor alignments and poor match scores for algorithms that rely upon them.

[35] Presents an extensive survey of a number of different methods for performing similarity search in metric spaces, with the main focus being on distance-based indexing methods. It introduces a framework for performing searches based on distances, and presents algorithms for common types of queries. It surveys common query and search algorithms, highlighting with examples in the area of spatial data. Methods discussed include Ball Partitioning, Hyperplane partitioning, M-Tree, SA-Tree and Distance Matrix Methods.

3.2 Fuzzy search methods

Fuzzy Logic can be an important inclusion into search algorithms for design applications. This is due to the fact that especially in preliminary stages of the design, where much is unknown, or cases where the design cannot yet be accurately described in query, there is possibility for many applicable but not exact matches to be excluded from the query results.

This scenario is well described in [42]. An algorithm is presented which has the purpose of allowing the search of a non-homogenous database, which contains engineering design information and other data relevant to the engineering design process. The purpose of this work is to develop a quick access mechanism to heterogeneous, complex knowledge. The algorithm is based upon a new type of fuzzy query (i.e. iterative, ranked retrieval) from an exclusive, partitioned, spatial database. A prototype system which implements a small database and this retrieval from a PLOP hashing database is described, and an example is presented which demonstrates its application.

This paper attempts to design processes by providing quick access to relevant engineering knowledge stored in a database. An engineer may not have an exact definition of the information being sought. The problem may be in the early, formative stage, and the information contained in the database is stored in terms of how it was used, rather than in terms of how the engineer now intends to use it. The algorithm described above is developed to help solve this problem. A fuzzy query takes into account tradeoffs among the facets of interest that form the database index. Conventional database systems ignore such tradeoffs, and thus may miss potential items.

Relaxed queries [43] are similar in aspect to fuzzy queries. Relaxed queries use fuzzy logic to introduce a 'grey area' in the query space, in order to reduce the amount of potentially missed results from a query operation. These fuzzy techniques are especially applicable in cases such as this, where *"a query submitted to the system is usually domain knowledge related and a user often fails to appropriately formulate his/her problem to obtain all relevant sequences and relevant data."* [43]

It can be especially seen in cases such as this, when working not just with database information, but with knowledge either contained within the database or derived from it, that the way queries are formulated can have a great impact on the efficiency of queries in terms of the number of successful results generated. Special care must be taken in these cases to ensure that representation of domain knowledge does not limit the data obtained from a database.

4 Selective creation and reuse of design space data

The solving of any design optimisation problem requires the creation of a multidimensional design space (or search/solution space). This design space is a representation of the different variables being studied in the design, and describes the relationships between these variables and the imposed constraints on the system. Along with a merit or objective function, the optimal solution within this search space can be found dependent on the user's requirements.

There are many issues to be covered in this pivotal field in design optimisation tools. In light of the current study, this section will concentrate on definition and optimisation of the design space, and search methods that can be used to determine an optimal solution. There are a range of numerical methods and algorithms available to determine solutions for design spaces, depending on the nature of the variables (discrete/continuous), relationships between the constraints, etc. These methods range in computational costs, and the definition of the design space boundaries has a remarkable effect on the overall cost of a solution, or even if any feasible (much less optimal) designs are contained within the design space.

4.1 Design space creation

The creation of a design solution space is a process of representing the limitations or constraints of a design in reference to the design variables. As such, these constraints will then form boundaries that determine whether a design is feasible or infeasible. The design process then continues in a search for a solution that satisfies these constraints, and if a merit or objective function is defined that determines the 'quality' of a design, the design space is searched for a valid design that yields the best quality.

Although constraints used during design include heuristics, tables, guidelines, and computer simulations, a majority of those used can be expressed as mathematical constraints. However, it is not enough to define constraints only as equalities, because a majority of the constraints define limitations on a design, rather than stipulating an exact relationship between variables. For example, a possible constraint may be that a beam cannot deflect more than a certain amount. [44]

Methods have been investigated [45, 46] for efficient representation of complex design systems. While not concentrating on an automated design system, [46] documents a method for generating a visual description of the design space, for ease of user understanding and navigation. This system is able to take a more active role in the generation of a computable design description than traditional computer aided design systems are able to, however the design process itself may remain largely under user control. [45] Presents a method of visualising the design space for user legibility, allowing the entire design space to be shown visually on one single design chart. These charts also allow visualisation of performance changes as design variables are altered. These nested charts allow for plotting of multi-variable design spaces, without complex issues regarding visualising relationships between charts for each variable. Both of these methods are aimed primarily at team-oriented design projects, rather than computational approaches to design solutions.

Another area of interest is that of decomposition of complex design spaces. An example of this is a multidimensional design space; in such an example the numerical method used for search of the design space will determine its efficiency in such a space. As such, there may be cases where it is highly advantageous to decompose the design space into numerous simpler spaces, which can be solved simultaneously, thus creating a more efficient solution method.

Liu and Tseng [47] propose a set of algorithms for Space-Decomposition Minimisation (SDM), which decomposes a solution space into a series of sub-problems. These sub-problems, if uncoupled, can be solved independently; otherwise one of the algorithms allows for solution of coupled solution spaces. These algorithms also potentially allow the design space to be broken down into one-dimensional solution spaces, allowing for simple 1-D solution algorithms. Also, given the 1-D design spaces, the SDM algorithms can be used as a direct search method, allowing for minimum solutions to be found directly, for example without requiring gradient information for the constraint and objective functions. These algorithms can also theoretically be run in parallel environments; however this case has not been implemented and tested at the time of writing (1999).

4.2 Design space optimisation

Traditionally, the design solution space has been defined by constraints that limit the design, defining its quality for the intended function. These may be performance requirements, materials limitations, or operational limitations. Often these constraints are user-specified, and in the desire to maximise the possible return of an optimal solution, it is allowable to assume that the designer will integrate a measure of conservatism in his/her estimation of these constraints, or more likely the range of design variables under study. This has the understandable effect of increasing the solution space, but it may then contain a large number of infeasible solutions, which the computer may have to evaluate, resulting in a higher computational cost. On the other hand, in cases where the computational cost of solution is known to be high, the solution space may be defined as too small, hence removing the optimal or even any feasible solutions from the design space. In light of these considerations, a number of studies [48–50] have been conducted investigating automated processes for efficiently determining the optimal size of the design space, in order to maximise the efficiency in obtaining a solution.

Yao and Johnson [50] present an implementation of a domain propagation algorithm to be used to identify suitable bounds for design variables. The program can generate a more focused search space from the original specification of variables and constraints without omitting any feasible solutions in the original search space. The program is also able to identify cases where the original search space contains no feasible design solutions. Tabulated results show extreme favour for this program. The program contains three options for non-linear optimisation: Genetic Algorithm, Simulated Annealing, and an improved SA algorithm. The SA algorithms show better performance for complex design cases; the reasons for this performance versus other search methods are discussed further in the next section.

Constraint-base design relies on the designers experience to select the bounds of design variables, or on conservative estimation, resulting in a larger design space. As such, methods such as these are useful, as they don't rely as much on the experience of the user in determining an appropriate boundary for the solution space, as the computer can automatically generate a set of suitable bounds. Also, the domain propagation program outlined in [50] makes the existing constraint-driven designs tools more reliable by increasing the chance of finding a feasible solution. In general, the use of the domain propagation program leads to an overall speed-up of the constraint-driven design process.

5 Compression of design space data

With the increasing complexity of design data, and the more widespread use of complex data types such as volumetric, spatial and high-resolution imaging, data compression is becoming very important for adequate storage of large volumes of data. Compression makes use of the redundancy inherent in data files, allowing for data to be truncated and stored in a smaller volume. This allows for more efficient use of storage hardware, and networking and data transmission facilities by conservation of transmission bandwidth and reduction of transmission time. While compression is becoming more and more invaluable a tool for storage of large amounts of data, additional complications arise when these compressed files, often compressed using differing methods, are planned for inclusion in a database, requiring search and retrieval of data records while still in the compressed form.

5.1 Data compression methods

Research has shown [51-54] that there are a vast number of methods for compressing data. This is due in part to the large number of forms that the data can take (binary, text, image, spatial, etc), and the intended use of the data, which determines the extent to which compression can take place. In some cases, where a high level of compression is paramount, and a certain loss of clarity in the data is acceptable, so-called 'Lossy' compression methods can be used. There are many applications however, where this may be unacceptable (Medical imaging being an example), and the data must be able to be perfectly reproduced from its compressed form.

The wide range of compression algorithms, and the range of its applications, means that these algorithms can become highly specialised, which becomes a problem in large-scale projects where some degree of commonality is sought. *"Researchers continue to strive to develop their own algorithms that maximise compression rates in the least amount of time. However, the all-encompassing algorithm does not exist, and probably never will, since the measurement criteria are both data and application dependent."* [55]

Due to the large range of data models available for discussion, we will cover the data models that particularly relate to the intended study. There are also a number of cases that will be covered very briefly. For Interests' sake, the reader is referred to the specific compression areas of String Matching [53], compression of volumetric data [54], and adaptive arithmetic coding [51].

5.2 Image compression

There is much work being done in the area of image compression, particularly for the field of medical imagery. This field has specific requirements for image compression, including largely (or in some cases totally) lossless compression, often of large individual or sets of high-resolution images, and the need for highly efficient compression is highlighted by the large amount of data being produced, for example at the Ghent University Hospital in Belgium, 10 Gigabytes of medical image data is produced each week [56]. *Due to the commonality of imagery data, and the high volume of research available especially in this field of imagery, we will briefly cover the aspects of image compression*.

A range of studies has been performed on image compression techniques. These range from simple comparisons of image compression techniques [56-58], to more advanced specialised studies. One example is particularly applicable to medical imaging, but also applicable to other forms of imaging, for example geographics or satellite imaging, where large image files are produced, but where certain areas are of key significance. Reference [59] documents a method of image compression where the image is compressed in a lossy method, however with specific regions compressed losslessly. This allows for very efficient compression, however without losing image quality in designated areas of importance in the image.

A similar method [60] can be used in cases where large sets of images are produced, for example in CT or MRI imaging. It presents a method for compressing sets of images, where similarity exists between the images, allowing for reduction of the image set redundancy, and therefore higher lossless compression. This is useful for fields such as medical imagery and satellite imagery. The Centroid method proposed here extracts these similar regions (similarity patterns), to enable this higher compression to be realised. This method could realistically be implemented for any case where there exist sets of images that are highly common in content, for example in storage of competitive designs, in the case of aircraft fuselage design; this example is used to highlight a case where there are common aspects between images (e.g. fuselage dimensions and configurations), however different images are used to depict possibilities for internal layout design, allowing for comparisons between potential design choices.

5.3 Volumetric and elevation data

Volumetric data can take many forms, from geometric or particle density modelling, to such design-based examples as CFD solutions, in the form of streamlines or pressure distributions within a control volume. Volumetric data takes up an immense amount of storage space, and is quoted as a prime example of the need for efficient compression algorithms in [54]. In many cases, it is also imperative that lossless compression algorithms are utilised, such that the data can be stored efficiently if possible, but predominantly without a loss in accuracy.

A study was performed [54] to develop an efficient method of compressing volumetric data. A Method was determined which uses Optimal Linear Prediction to exploit correlations in all three dimensions, yielding typical compression of around 50 %. These results were achieved using MRI images, CT images and electron-density map data as test data; however it is believed that similar results can be obtained for other forms of volumetric data.

5.4 Database search of compressed data

Many compression techniques are utilised to compress a range of different data types and formats (a comprehensive review of these techniques can be found in [61]). The compression process has the potential to remove much of the schema, or legibility, of the file, as the redundant data is removed. While this is insignificant when regarding storage of the data, the issue becomes prevalent when you want to manipulate or search the file in its compressed form, for example in Database query processing. This issue arises because the contents of the compressed file are now required to be accessed and searched in its compressed form, which is more efficient than expending computational cost and time to decompress the file before accessing data records, with the extreme example being in very large databases, consisting of large numbers of sizeable records, as each record would need to be decompressed before query.

"Compressing information in a database system is attractive for two major reasons: storage saving and performance improvement. Storage saving is a direct and obvious benefit, whereas performance improvement derives from the fact that less physical data need be moved for any particular operation on the database." [62]

The primary concern in compressed database querying is that a common compression scheme is utilised for all the data domains in the database, thereby allowing for more efficient querying of the data while compressed. Graefe et al. have shown in their study [63] that many query-processing applications can manipulate compressed data just as well as uncompressed data, and more-so that processing compressed data can speed up query processing by a factor much larger than the compression factor of the data.

In query processing, compression can be exploited far beyond simply improving I/O performance. Exact-match comparisons can be performed on compressed data, as long as the compression scheme is consistent, as all matching values will be encoded in the same manner. In the same fashion, projection and duplicate elimination can be performed on compressed data. The main factor for these query operations is that a common compression method is used, and therefore comparisons can be made on the compressed data in a similar manner to uncompressed data, without much change to the implemented algorithms [63]. This paper also documents performance analysis, showing that for data sets larger than the memory performance of the system, performance gains larger than the compression factor can be gained, due to larger amounts of the data being stored in the workspace allocated to the query operator. This is highly applicable to databases containing large amounts of complex data.

This section highlights two major factors for implementation of a large-volume database of design data. The end intent for the proposed system would be for a very large database, including a large number of records, potentially each being of a large file size. Therefore efficient access methods are required not only for access and query in the database, but also for access to the files themselves during the query. This would require a robust filing system for efficient access, an efficient method for access of records from memory storage, for example in the form of Disk Striping over multiple physical drives, and compressed data coupled with a suitable query engine able to manipulate the compressed data.

6 Efficient representation of large data sets, designs and states

This section will primarily discuss the different and evolving methods for representing complex geometries, with particular concentrations towards design applications. Large Data sets have already been discussed previously, particularly in the chapters describing the different database technologies capable of handling these data sets. This chapter will outline the different methods available for accurate yet efficient geometric representation of simple and complex geometries, covering a range of complexities in approaches to the problem.

6.1 General shape-representation methods and cad tools

Various methods of defining and parameterising shapes for use in design optimisation processes can be found in the literature. For example, an early method for shape parameterisation is the nodal coordinate approach that uses the coordinates of nodes of the discrete FEM model as design variables [64]. In practical design optimisation situations, this unfortunately leads to a large number of design variables, and therefore a high level of inefficiency in design optimisation. Several methods to overcome these initial drawbacks have been formulated. These include the mesh parameterisation approach [65], the use of solid modelling [66], and the natural design variable method [67]. Although these approaches are relatively easy to implement for realistic design optimisation problems due to the very large number of design variables associated with them, this often makes them too costly and thus makes design optimisation prohibitive.

Another popular approach for defining and parameterising shape for use in design optimisation is the so-called spline approach, where the shape is represented by means of a series of polynomial functions. Two common methods of spline based surface representation are Bézier and B-Splines [68]. Typically, the surface to be represented is broken into a mesh of primarily rectangular curvilinear regions. A surface patch is then defined over each region, whose shape is determined by a set of control points. The shape parameters in this formulation are the coordinates of each control point [69]. It can thus be seen that in order to represent the shape of a practical object it would require a number of surface patches, often involving a large number of shape parameters. Furthermore, for design optimisation involving complex geometry the spline based approach makes it difficult to maintain smooth transitions between adjacent surface patches.

Examples have been found in the literature for proposed systems for aiding the user in CAD processes. Qin, Wright and Jordanov [70] present the development of a sketch-based CAD system interface for assisting designers during conceptual design stages. The system captures designers' intention and interprets the input sketch into geometrically more exact 2D vision objects and further 3D models. It could also allow designers to specify a 3D object or a scene quickly, naturally, and accurately.

The use of fuzzy knowledge in such a system is particularly useful in the conceptual design stages, where there is still a large proportion of uncertainty in the design. In these stages, many designers prefer to work on paper, using rough sketches to process the design. To support this early stage of geometric design and to improve the speed, effectiveness and quality of the design decision, the authors' studies indicate that a computer aided conceptual design system must allow sketched input, and must have a variety of interfaces, recognising features and managing constraints.

As a tool, CAD systems have been widely used to reduce design time and cost, and to improve design efficiency and quality. However, there are still some problems with current CAD systems. First, present CAD systems primarily support drafting and detailed design. They have little, if any, support for early stage design although early design is critically important in the development of new products. Second, the process of making a 3D design with present CAD systems is often lengthy and tedious. One reason for this is that, with 2D interface, users have to decompose 3D design tasks into 2D or 1D modelling operations and have to input detailed and complete specifications of a design. Third, the visualization capability of present CAD systems is limited, which may not always satisfy the requirements of design analysis. The integration of VR techniques allows some of these shortcomings to be addressed.

6.2 Spline-based methods

Spline-based methods are a very efficient way of representing complex curves and surfaces using a small number of control points which define one or more polynomial functions. Common methods of spline-based curve and surface representation are Bézier and B-Splines, and NURBS (Non Uniform Rational B-Splines). These methods not only allow for the more data-efficient storage of geometric objects, but also allow for geometries to be created when only a limited amount of data is available. While these methods have advantages in requiring only a small amount of control points for definition, there must be concentration of these points in areas of complex geometries, for example with rapid changes in surface geometries, inflections, etc.

In fact, NURBS have been recognised in much of the literature as a prevalent shape definition method: *"The representation of curves and surfaces in NURBS form is now an accepted industry standard; hence, it is of practical interest to have NURBS descriptions of all curves and surfaces occurring in the design and manufacturing process"* [71]. There are a number of practical examples available outlining the use of spline methods in geometric representation or design applications [72–76]. In their paper, Pottman and Farin [71] describe the use of NURBS for surface definition in sheet-metal and plate-metal based applications.

Spline methods have shown to be useful in the generation of object meshes and grids, as outlined in [77–79]. For a survey of the literature covering various procedures to automate the transition between the modelling and analysis phases for design, the reader is referred to [80]. Mastin [79] shows that solid modelling techniques using 3-dimensional Bézier functions can be used to generate grids in a simple one-step procedure. These grids are loosely formed around a collection of defined points and vertices. For example, simply the vertices of a cube can be used to define a sphere, with the grid then being defined on the sphere's surface using these vertices. Many different types of configurations and edge and face treatments can be included in the model. As such, the complexity of the geometric model is only limited by the amount of input that is desired. This solid modelling technique is designed for free-form solids where a general shape or design is to be modelled rather than for constructing a solid with precisely defined edges or surfaces.

Yu and Soni [77] sent another approach for surface grid generation using NURBS and enhanced algorithms to transform IGES entities into NURBS definitions. This application is much more accessible to engineers and designers, through the implementation of common file types already in common usage in FEM, CFD and CAD software, through the use of the Initial Graphics Exchange Specification (IGES) file format. This allows for designers to create the geometry in an existing application, allowing for efficient generation through familiarity with known software. The IGES format file can then be converted into a NURBS or B-Spline representation, ready for grid generation for future analysis. It has been noted by the authors that NURBS is becoming the *de facto* standard for geometry description in most modern grid generation applications, with many tools being available for NURBS definition.

Many of the previously mentioned applications are primarily user-based, rather than being automated procedures. This is due to the fact that free-form shape design is typically accomplished in an interactive manner, with computer-generated shapes rarely being immediately acceptable, making this part of the design process far from being rightfully called computer-aided. Often the user has to manipulate a large number of variables (control points) in order to produce the desired geometric properties. Keyser et al. propose a method for efficient exact manipulation of algebraic points and curves [81]. Hohenberger and Reuding [82] propose a method for manipulation of B-splines in an automatic optimisation scenario, through the use of weights at the control points. In many CAD systems, the use of weights for NURBS representations is inadequately supported, and often hidden from the user and therefore remains unused. In this application, the perturbation of weights at the control points is defined in an optimisation problem, with the objective being to produce a curve with a more gradual change in curvature and the smallest deviation from its initial shape. Examples of applications in automotive shape design are presented and discussed by the authors as practical examples of this method.

6.3 Partial differential equation (PDE) formulations

The PDE method defines a shape in terms of a number of surface patches that collectively describe the object's surface. The shape of the surface is defined through boundary conditions and a small set of design parameters. The boundary conditions can be specified in terms of curves in 3D-space. It is these features of the method that can be utilised for interactive design.

Although the PDE method has certain features in common with more established techniques for surface design (B-Splines, Bézier curves and NURBS), what distinguishes it from these conventional techniques is its global smoothing approach associated with its elliptic boundary-value formulation. Unlike conventional techniques (which are spline-based) for surface representation, the PDE method can parameterise complex surfaces in terms of a small set of design variables, instead of many hundreds of control points.

"The principle strength of the PDE method lies in the ease with which we can quickly change the geometry using a small number of global design parameters. Once a design has been determined, it would be a relatively straightforward procedure to derive the more commonly used design parameters (target points, upsweep angles, etc.) from the geometry. [This allows designers to obtain] useful information from simple assumptions; this is again advantageous when analysis of many models is needed." [83]

Practical examples of the use of the PDE formulation in geometric modelling and design have been found in the literature [83–87]. In [83], Dekanski et al. present a practical implementation for creation and testing of a geometric design, incorporating simplified geometrical design, CFD modelling and analysis of the gas exchange cycles in a 2-stroke engine, and optimisation of design parameters to maximise engine scavenging (removal of combustion products during each cycle).

Bloor and Wilson [87] present a method for the creation of wing geometries based on a series of 2D wing aerofoil sections. This builds on their earlier work concerned with the application of elliptic PDE's to the parameterisation of generic aircraft geometries [88]. This method has been shown to be capable of producing PDE surface patches of the wing geometries of high-speed cargo transport (HSCT) aircraft, allowing the designer to then progress with CFD analyses in this application. In this method, the 2D sectional data is interpolated between sections via a variable smoothing parameter, in order to control the PDE solution. This can be viewed as a lofting method; unlike conventional CAD lofting techniques, however, the PDE method does not use spline techniques for the loft. It allows radical changes to be made to a design very quickly and cheaply, with a minimum of user-intervention since its surface definition remains valid, i.e. closed, throughout any changes in the design variables. This method can also be used in this example to smoothly join the wings to the aircraft fuselage [88, 89].

The PDE formulation has been shown to be an advantageous parameterisation method for the efficient creation and manipulation of complex objects. The low number of design variables inherent to this method, when compared to conventional methods such as spline-based methods, makes it every efficient to integrate an optimiser to perform shape optimisation.

6.4 Expanding past geometric representations

Previous sections have outlined various methods for describing the geometric definition of an object. Methods have also been developed that allow representations to expand past this basic functionality, also including such information as internal material structure [90, 91], design histories [92], and other capabilities arising from the representation of geometric objects in Object-Oriented terms [92, 93].

"Heterogenous Solid Modelling" is an expansion of the solid modelling concept, such that a 'heterogenous object' is defined that can have different material composition within the object. This concept is a new area in CAD, and still in its infancy. In their proposal of this concept, Siu and Tan [90] present a representation scheme for heterogenous objects, whereby material information can be integrated as a part of the object representation.

Kumar et al. [91] present a similar approach, which can include not only material properties, but also grading between different materials for composites material applications such as aircraft engine turbine blades, which are a complex blending of metals and ceramics, along with a very accurate grading and geometry definition. These capabilities are called for due to recent advances in materials creation and analysis software, where the analysis must be able to keep up with the pace of materials advances.

7 Heuristics for switching between classes of models

This section will review heuristic and stochastic process that can be utilised by the knowledge-based processes of the proposed system. It is envisaged that a number of potential forms of data will be available for the design at any stage, with each having differing levels of accuracy and availability/applicability. These forms of data could range from CFD solutions to empirical tables and formulae. At a point during the design process, any number of these potential options may be open, and it is important to determine which will be the most efficient, in terms of computation and accuracy required. A range of information and practical examples have been found in the literature for intelligent design systems [94–126].

7.1 Knowledge base

An expert system is a computer program that has an extensive knowledge base in a specific domain, and uses an inference mechanism to draw conclusions in the same manner as a human expert in the same domain. Since they are not subject to human frailties such as boredom, forgetfulness, bias or tunnel vision, fully developed expert systems frequently out-perform their human models. In the design of an expert system, the development consists of three major phases; namely, knowledge acquisition, internal design of the knowledge base, and expert system validation. Among these activities, the design of the knowledge base forms the most crucial part from the performance viewpoint of the expert system [127], and there are a number of methods available to specify the reasoning of knowledge-based systems [128, 129].

A knowledge base usually consists of conceptual, procedural and declarative knowledge. Conceptual knowledge is concerned with underlying ideas, theories, concepts, hypotheses and relationships that exist within a domain. Declarative knowledge consists of the truths of a domain, and includes facts, terminology and classifications. Procedural knowledge refers to knowledge used to direct pathways of thought or actions, leading to solutions to problems that the system is trying to solve. Procedural knowledge is largely concerned with the manipulation of declarative or conceptual knowledge. This type of knowledge consists of rules-of-thumb and their control, weights of evidence, working procedures and strategies. It has been noted that the major problem with building a knowledge base to solve aspects of a design problem is in finding the most effective representation for the knowledge [130]. For fully developed expert systems, these knowledge bases can be substantial in size; however Dev and Murthy [127] present an approach to the problem of knowledge-based partitioning in the context of rule-based expert systems.

Advantages of these systems lie in the inference mechanisms used to determine the best approach for a given design domain, thereby removing the bias, favouritism and familiarity of human designers. This not only ensures that an appropriate search or optimisation method will be chosen, it also reduces the reliance on designer experience, which can limit the potential for an optimal design. Also, such systems will make full use of the data generated by designers during design-evaluate-redesign studies, which are otherwise often discarded.

7.2 Heuristic processes utilising previous design data

The design process can be made more efficient if it is able to recognise previous relevant design data; this concept is fundamental to this study. Gantovnik, Gurdal and Watson [131] propose a method for augmenting Genetic Algorithms to include memory for discrete and continuous variables, with the term 'memory' implying preserved data from previously analysed designs.

In the standard GA approach, a new population may contain designs that have already been encountered in previous generations, especially near the end of previous design optimisation processes. The memory procedure eliminates the possibility of re-creating these design candidates, thereby saving computational time. After a new generation of designs is created by the genetic operations, the binary tree used for memory storage is searched for each new design. If the design is found, the fitness value is retrieved from the binary tree without conducting an analysis. Otherwise, the fitness is obtained based on an exact analysis. This new design and its data are then inserted in the tree as a new node.

This approach proves to serve well for design systems based wholly on discrete variables. For cases that include a continuous variable, a spline interpolation method is utilised. The main idea of this approach is to construct approximations for the fitness function as a function of the continuous variable using a spline function fitted to the historical data, and interpolate from the stored data whenever possible.

7.3 Approximation methods for the design process

A common engineering practice is the use of approximation models [132, 133] in place of expensive computer simulations to drive a multidisciplinary design process based on non-linear programming techniques. The use of approximation strategies is designed to reduce the number of detailed, costly computer simulations required during optimisation while maintaining the pertinent features of the design problem. After each sequence of approximate optimisation, the approximations of system behaviour are updated with new information about the current design. Thus, many iterations of such algorithms may be required before convergence of the optimisation process is achieved, and every additional iteration adds to the cost of the process. In light of this, a primary concern in developing an approximate optimisation strategy is the proper choice of a move limit management strategy.

Two main alternatives have been investigated in the Multidisciplinary Design Optimisation community to approximate physical systems. The first approach has been the use of a simplified physical representation of the system to obtain less costly simulations as described in [134]. A second alternative for system approximation, which has grown in interest in recent years, are response Surface Approximations (RSA's) based on polynomial and interpolation models. Polynomial RSA's employ the statistical techniques of regression analysis and analysis of variance (ANOVA) to determine the approximate function.

Rodriguez et al. [135] overview the current state of the art in model management strategies for approximate optimisation. Model management strategies coordinate the interaction between the optimisation and the fidelity of the approximation models in order to ensure that the process converges to a solution of the original design problem. Approximations play an important role in multidisciplinary design optimisation (MDO) by offering system behaviour information at a relatively low cost. Most approximate MDO strategies are sequential, in which an optimisation of an approximate problem subject to design variable move limits is iteratively repeated until convergence. The move limits, or *trust region*, are imposed to restrict the optimisation to regions of the design space in which the approximations provide meaningful information.

As computers advance in speed, more efficient data sharing and exchange algorithms are developed. One observes that an increasing number of discipline sets are being encompassed in actual engineering optimisation processes. Problem complexity is observed to grow at a pace that taxes the limits of the advances in processing powers. Therefore, the dimensionality and complexity of MDO problems may always necessitate the use of approximations and decomposition strategies to make the optimisation a practical task.

7.4 Heuristics for resource selection

The proposed system would include inference algorithms that are able to choose from a selection of data types and approximation techniques. Ref [136] documents a more practical-minded implementation of the desired heuristics. Where the proposed system would implement these heuristics to determine the best calculations to perform, this implementation determines the best processes to be performed in a construction project. *"Such resource-assignment and optimisation problems demand efficient combinational computations if all the possible options are to be considered, and decision-making facilitated."* Consequently, research into efficient methods of resource optimisation has always been an area of interesting study on its own. Previous research works on resource optimisation have investigated the use of deterministic models in construction decision-making, while other works investigated the use of stochastic models in solving the problem [137–139].

Ugwu and Tah [136] present an investigation into the application of genetic algorithms (GAs) to the multidimensional problem. The objective is to investigate the use of GAs for both a numerical function optimisation and a combinatorial search problem within the framework of a decision-support system (DSS). A hybrid GA system was designed for construction-resource selection, and a genetic model that represents the problem and solution space was built into the system (methods for accurately simulating complex nonmathematical processes for solution in an optimisation process can be found in [140, 141]). A genetic state-space search (GSSS) technique for multimodal functions was used to evaluate the cost profiles that resulted from different combinations of tasks and resources. The study indicates that GA systems have huge potential applications as DSS component(s) in construction--resource assignment. The results also highlighted that GA's exhibit the chaotic characteristics that are often observed in other complex non-linear dynamic systems. The power of their use in applications is derived from their ability to combine numerical parameter optimisation with combinatorial searches within an application domain. GA's are therefore uniquely suitable for solving multidimensional optimisation problems such as this.

The algorithm discussed in this paper also demonstrates the use of a hybrid GA that is integrated with a project database to perform combinatorial optimisation. This improves the robustness of the GA because the services it provides (functional and combinatorial optimisation) are independent of the data on which it acts in performing such services. This distinct feature means that the imposition of genetic operators such as reproduction, crossover, and mutation do not result in an arbitrary loss of information, since the knowledge about the problem domain is stored in the project database. This integration of GA with the project database allows for a wide range of applications in real-time or real-life situations [142].

While the above approaches have primarily been concerned with practical applications, and the allocation of physical resources as opposed to calculations of varying complexity, it is possible that a similar approach can still be used for the proposed project. Parameters can be defined within the GA structure, relating to the cost and performance characteristics of different search methods and calculation techniques, allowing for the use of Genetic Algorithms to determine the optimal approach in an optimisation scenario. This approach could also be integrated with previously discussed methods such as memory integration for Genetic Algorithms, or approximation techniques.

8 Conclusions and recommendations

An extensive literature review has been conducted by the Sir Lawrence Wackett Centre for Aerospace Design Technology, RMIT University. The review investigates aspects of a proposed system for intelligent design optimisation. Such a system would be capable of efficiently storing (and compressing if required) a range of types of design data into an intelligent database. This database would be accessed by the system during subsequent design processes, allowing for search of relevant design data for re-use in later designs, allowing it to become very efficient in reducing the time for later designs as the database grows in size. Extensive research has been performed, in both theoretical aspects of the project, and practical examples of current similar systems.

Database systems have been reviewed and discussed. Aspects of databases such as DBMS and Data Model have been discussed, along with the general consensus in the literature of Object-Oriented technologies allowing great advantages for design applications. Aspects of database design, integration with existing software and expert systems, as well as storage issues for very large databases have also been addressed. Database query methods have been reviewed, and compared for the different data models available. Query optimisation techniques have been illustrated, and fuzzy or knowledge-based enhancement methods have also been noted.

A critical component of the proposed system is the efficient and accurate representation of design space data. Design space creation, optimisation and search methods have been discussed, along with methods for compressing various types of data that could be encountered in design. These data types include images, volumetric and basic geometric data, to name a few. Particular aspects of database searches performed on such compressed data have also been addressed, and it has been noted that searches can be performed on the data in compressed form provided that given information is known, such as the utilised compression schemes; this has a great impact on the search efficiency within the database.

Numerous methods have been discussed for efficient representation of geometric objects, covering basic techniques, spline-based methods, solid modelling, and PDE formulations. It has been found that the PDE (Partial Differential Equation) formulation shows the best efficiency for accurate portrayal of complex 3D geometries, due to its reliance on a low number of design variables, as compared to spline methods, which can rely on hundreds of control points at a time.

These sections have discussed all the important functional aspects of the proposed system. The final chapter addresses the intelligent algorithms that are required to differentiate between different design methods and model classes, a number of which may be available at any time, but which could vary in efficiency, cost and accuracy. A range of methods have been discussed for these inference mechanisms, including memory processes for Genetic Algorithms, approximation methods, and resource allocation systems.

This review provides an accurate and timely review of the literature pertinent to the proposed application. During the extent of this project, the proposed system was kept in general terms, and it is envisaged that more accurate or targeted research can be attempted when the proposition becomes more defined.

9 Acknowledgments

The authors would like to acknowledge the Defence Science and Technology Organisation (DSTO) for their support for this project.

References

[1] Silberschatz, A., Stonebraker, M., Ullman, J.: "Database Systems: Achievements and Opportunities." *Communications of the ACM*, Vol. **34** (1991), No. 10, p. 110–120.

[2] Sidle, T. W.: "Weaknesses of Commercial Database Management Systems in Engineering Applications." In: Proceedings of the 17th Design Automation Conference on Design Automation. 1980, Minneapolis, Minnesota, USA: ACM Press, NY, USA.

[3] Eastman, C. M.: "System Facilities for CAD Databases." In: Proceedings of the 17th Design Automation Conference on Design Automation. 1980, Minneapolis, Minnesota, USA: ACM Press, NY, USA.

[4] Bennet, J.: "A Database Management System for Design Engineers. In: Proceedings of the 19th Design Automation Conference. 1982, IEEE Press, NJ, USA.

[5] Lacroix, M., Pirotte, A.: "Data structures for CAD object Description." In: Proceedings of the 18th Design Automation Conference on Design Automation. 1981, Nashville, Tennessee, USA: IEEE Press, NJ, USA.

[6] Haynie, M.: A DBMS for Large Design Automation Databases. In: 1988 ACM SIGMOD International Conference on Management of Data, 1988, Chicago, Illinois, USA: ACM Press, NY, USA.

[7] Schutzman, H. B.: "ICHABOD: a Data Base Manager For Design Automation Applications. In: Proceedings of the 22nd ACM/IEEE Conference on Design Automation, 1985. Las Vegas, Nevada, USA: ACM Press, NY, USA.

[8] Haynie, M. N.: "The Relational/Network Hybrid Data Model for Design Automation Databases." In: 18th Design Automation Conference on Design Automation. 1981, Nashville, Tennessee, USA: IEEE Press, NJ, USA.

[9] Hardwick, M.: Extending the Relational Database Data Model for Design Applications. In 21st Proceedings of the Design Automation Conference on Design Automation. 1984, Albuquerque, New Mexico, USA: IEEE Press, NJ, USA.

[10] Ketabchi, M. A., Berzins, V., March, S.: "ODM: An Object Oriented Data Model for Design Databases." In: 1986 ACM 14th Annual Conference on Computer Science. 1986, Cincinnati, Ohio, United States: ACM Press, New York.

[11] Navathe, S. B.: "Evolution of Data Modeling for Databases." *Communications of the ACM*, Vol. **35** (1992), No. 9, p. 112–123.

[12] Heiler, S. et al.: An Object-Oriented Approach to Data Management: Why Design Databases Need It. In: 24th ACM/IEEE Conference Proceedings on Design Automation Conference. 1987, Miami Beach, FL, USA: ACM Press, NY, USA.

[13] Agrawal, R., Gehani, N. H.: "ODE (Object Database and Environment): The Language and the Data Model." In: Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data. 1989, Portland, OR, USA: ACM Press, NY, USA.

[14] Haynie, M. H.: "Tutorial: The Relational Data Model for Design Automation." In: 12th Design Automation Conference on Design Automation. 1983, Miami Beach, Florida, USA: IEEE Press, NJ, USA.

[15] Maryanski, F. et al.: "The Data Model Compiler: a Tool for Generating Object-Oriented Database Systems." In: The 1986 International Workshop on Object-Oriented Database Systems. 1986, Pacific Grove, CA, USA: IEEE Computer Society Press, LA, USA.

[16] Bedell, J., Maryanski, F.: "Semantic Data Modeling Support for CAD." In: Proceedings of the 1987 Fall Joint Computer Conference on Explorer Technology: Today and Tomorrow. 1987, Dallas, TX, USA: IEEE Computer Society Press, LA, USA.

[17] Loomis, M. E. S., Chaudhri, A. B.: *Object Databases in Practice*. 1998, London: Prentice-Hall International. xviii, 312.

[18] Lee, K.-H., Lee, D., Han, S.-H.: "Object-Oriented Approach to a Knowledge-Based Structural Design System. *Expert Systems with Applications*, Vol. **10** (1996), No. 2, p. 223–231.

[19] Spooner, D. L.: "An Object-Oriented Data Management System for Mechanical CAD." In: 1986 International Workshop on Object-Oriented Database Systems. 1986, Pacific Grove, CA, USA: IEEE Computer Society Press, Los Alamitos, CA, USA.

[20] Ketabchi, M. A.: "Object-Oriented Data Models and Management of CAD Databases." In: Proceedings of the 1986 International Workshop on Object-Oriented Database Systems. 1986, Pacific Grove, CCalifornia, USA: IEEE Computer Society Press, LA, USA.

[21] Ishikawa, H.: *Object-Oriented Database System: Design and Implementation for Advanced Applications*. Computer science workbench. 1993, Tokyo; New York: Springer--Verlag. cm.

[22] Zand, M., Collins, V., Caviness, D.: "A Survey of Current Object-Oriented Databases." *ACM SIGMIS Database*, Vol. 26 (1995), No. 1, p. 14–29.

[23] Ahad, R.: "A Framework to Support an Object-Oriented View of Existing Engineering Databases. In: Proceedings of the 1988 ACM SIGSMALL/PC symposium on ACTES. 1988, Cannes, France: ACM Press, NY, USA.

[24] Stonebraker, M.: "Object Management in POSTGRES Using Procedures." In: 1986 International Workshop on Object-Oriented Database Systems. 1986, Pacific Grove, CA, USA: IEEE computer society press, Los Alamitos, CA, USA.

[25] Manegold, S., Boncz, P. A., Kersten, M. L.: "Optimising Database Architecture for the New Bottleneck: Memory Access." *The VLDB Journal - The International Journal on Very Large Data Bases*, (2000), No. 9, p. 231–246.

[26] Ramakrishna, M. V., Larson, P.- .: "File Organisation Using Composite Perfect Hashing." *ACM Transactions on Database Systems (TODS)*, Vol. **14** (1989), No. 2, p. 231–263.

[27] Roller, D., Eck, O., Dalakakis, S.: "Integrated Version And Transaction Group Model For Shared Engineering Databases." *Data & Knowledge Engineering*, Vol. **42** (2002), No. 2, p. 223–245.

[28] Graefe, G.: "Query Evaluation Techniques for Large Databases." *ACM Computing Surveys* (CSUR), Vol. 25 (1993), No. 2, p. 73–169.

[29] Analyti, A., Pramanik, S.: Fast Search in Main Memory Databases. In: ACM SIGMOD, 1992, CA, USA.

[30] Jarke, M., Koch, J.: "Query Optimisation in Database Systems." *ACM Computing Surveys* (CSUR), Vol. **16** (1984), No. 2, p. 111–152.

[31] Chaudhuri, S.: "An Overview of Query Optimisation in Relational Systems." In: PODS '98. 1998, Seattle, WA, USA.

[32] Böhm, C., Berchtold, S., Keim, D. A.: "Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases." *ACM Computing Surveys* (CSUR), Vol. **33** (2001), No. 3, p. 322–373.

[33] Funkhouser T., Kazhdan, M. P. et al.: "A Search Engine for 3D Models." *ACM Transactions on Graphics* (TOG), Vol. **22** (2003), No. 1, p. 83–105.

[34] Gaude, V., Günther, O.: "Multidimensional Access Methods." *ACM Computing Surveys* (CSUR), Vol. **30** (1998), No. 2, p. 170–231.

[35] Hjaltason, G. R., Samet, H.: "Index-Driven Similarity Search in Metric Spaces." *ACM Transactions on Database Systems* (TODS), Vol. **28** (2003), No. 4, p. 517–580.

[36] Owrang, O., Omidvar, M. M.: *A Parallel Database Machine for Query Translation in a Distributed Database System*. p. 47–52.

[37] Ciaccia, P., Patella, M.: "Searching in Metric Spaces with User-Defined and Approximate Distances." *ACM Transactions on Database Systems* (TODS), Vol. **27** (2002), No. 4, p. 398–437.

[38] Kozlowski, M. C., Panda, M.: "Computer-Aided Design of Chiral Ligands: Part I. Database Search Methods to Identify Chiral Ligand Types for Asymmetric Reactions." *Journal of Molecular Graphics and Modelling*, Vol. **20** (2002), No. 5, p. 399–409.

[39] Keim, D. A.: "Efficient Geometry-Based Similarity Search of 3D Spatial Databases." In: 1999 ACM SIGMOD Intl. Conf. on Management of Data. 1999, Phil, Penns, USA.

[40] Faloutsos, C. et al.: "Efficient and Effective Querying by Image Content." *Journal of Intelligent Information Systems*, Vol. **3** (1994), p. 231–262.

[41] Mehrotra, R., Gary, J. E.: "Feature-Based Retrieval of Similar Shapes." In: Proc. 9th Int. Conf. on Data Engineering. 1993, Vienna, Austria.

[42] Manwaring, M. L., Jones, K. L., Glagowski, T. G.: *An Engineering Design Process Supported by Knowledge Retrieval from a Spatial Database*. IEEE: p. 395–398.

[43] Chen, Y., Che, D., Aberer, K.: "On the Efficient Evaluation of Relaxed Queries in Biological Databases." In: Proceedings of CIKM '02. 2002, McLean, Virginia, USA.

[44] Thornton, A. C.: "The Use Of Constraint-Based Design Knowledge to Improve the Search for Feasible Designs." *Engineering Applications of Artificial Intelligence,* Vol. **9** (1996), No. 4, p. 393–402.

[45] Burgess, S., Pasini, D., Alemzadeh, K.: "Improved Visualisation of the Design Space Using Nested Performance Charts." *Design Studies*. In Press, Corrected Proof.

[46] Chien, S. -F., Flemming, U.: "Design Space Navigation In Generative Design Systems." *Automation in Construction*, Vol. **11**(2002), No. 1, p. 1–22.

[47] Liu, C. -S., Tseng, C. -H.: "Space-Decomposition Minimization Method for Large-Scale Minimization Problems*1." *Computers & Mathematics with Applications*, Vol. **37** (1999), No. 7, p. 73–88.

[48] Chen, T. -Y., Lin, C. -Y.: "Determination of Optimum Design Spaces for Topology Optimization." *Finite Elements in Analysis and Design*, Vol. **36** (2000), No. 1, p. 1–16.

[49] Sanchez Marin, F. T., Perez Gonzalez, A.: "Global Optimization in Path Synthesis Based on Design Space Reduction." *Mechanism and Machine Theory*, Vol. **38** (2003), No. 6, p. 579–594.

[50] Yao, Z., Johnson, A. L.: "On Estimating the Feasible Solution Space of Design." *Computer-Aided Design*, Vol. **29** (1997), No. 9, p. 649–655.

[51] Chen, P. -Y., Jou, J. M.: "Adaptive Arithmetic Coding Using Fuzzy Reasoning and Grey Prediction." *Fuzzy Sets and Systems*, Vol. **114** (2000), No. 2, p. 239–254.

[52] Holtz, K., Holtz, E.: *Lossless Data Compression Techniques*. p. 392–397.

[53] Yang, E., Kieffer, J. C.: "On the Performance of Data Compression Algorithms Based upon String Matching." *IEEE Transactions on Information Theory*, Vol. **44** (1998), No. 1, p. 47–65.

[54] Fowler, J. E., Yagel, R.: "Optimal Linear Prediction for the Lossless Compression of Volume Data." *IEEE Transactions on Computers*, 1995, p. 458.

[55] Kidner, D. B., Smith, D. H.: "Advances in the Data Compression of Digital Elevation Models." *Computers & Geosciences*, Vol. **29** (2003), No. 8, p. 985–1002.

[56] Philips, W. et al.: "State-of-The-Art Techniques for Lossless Compression of 3D Medical Image Sets. *Computerized Medical Imaging and Graphics*, Vol. **25** (2001), No. 2, p. 173–185.

[57] Ageenko, E., Franti, P.: "Lossless Compression of Large Binary Images in Digital Spatial Libraries*1." *Computers & Graphics*, Vol. **24** (2000), No. 1, p. 91–98.

[58] Kivijarvi, J. et al.: "A Comparison of Lossless Compression Methods for Medical Images." *Computerized Medical Imaging and Graphics*, Vol. **22** (1998), No. 4, p. 323–339.

[59] Strom, J., Cosman, P. C.: "Medical Image Compression with Lossless Regions of Interest*1." *Signal Processing*, Vol. **59** (1997), No. 2, p. 155–171.

[60] Karadimitriou, K., Tyler, J. M.: "The Centroid Method for Compressing Sets of Similar Images." *Pattern Recognition Letters*, Vol. **19** (1998), No. 7, p. 585–593.

[61] Lelewer, D. A., Hirschberg, D. S.: "Data Compression." *ACM Computing Surveys* (CSUR), Vol. **19** (1987), No. 3, p. 261–296.

[62] Cormack, G. V.: "Data Compression on a Database System." *Communications of the ACM*, Vol. **28** (1985), No. 12, p. 1336–1342.

[63] Graefe, G., Shapiro, L. D.: "Data Compression and Database Performance." *IEEE Transactions on Computers*, 1991, p. 22–27.

[64] Francavilla, A., Ramakrishanan, C. V., Zienkiewicz, O. C. Z.: "Optimisation of Shape to Minimise Stress Concentration." *Journal of Strain Analysis*, Vol. 10 (1975), p. 63–70.

[65] Chang, K. H., Choi, K. K.: A Geometry Based Parametrisation Method for Shape Design of Elastic Solids." *Mech Struct Mach*, Vol. **20** (1992), p. 215–52.

[66] Kodiyalam, S., Kumar, V., Finigan, P. M.: "Constructive Solid Geometry Approach to Threedimensional Shape Optimisation." *Journal of the AIAA*, Vol. **30** (1992), p. 1408–15.

[67] Tortorelli, D. A.: "A Geometric Representation Scheme Suitable for Three-Dimensional Shape Optimisation." *Mech Struct Mach*, Vol. **21** (1993), p. 95–121.

[68] Schumaker, L. L.: *Spline Functions: Basic Theory*. New York: John Wiley and Sons, 1981.

[69] Braiband, V., Fleury, C.: "Shape Optimal Design Using B-Splines." *Computer Methods in Applied Mechanics and Engineering*, Vol. **44** (1984), p. 247–67.

[70] Qin, S. F., Wright, D. K., Jordanov, I. N.: "From On-Line Sketching To 2D And 3D Geometry: A System Based On Fuzzy Knowledge." *Computer-Aided Design*, Vol. **32** (2000), No. 14, p. 851–866.

[71] Pottmann, H., Farin, G.: "Developable Rational Bezier and B-Spline Surfaces." *Computer Aided Geometric Design*, Vol. **12** (1995), No. 5, p. 513–531.

[72] de Kemp, E. A.: "Visualization of Complex Geological Structures Using 3-D Bezier Construction Tools*1." *Computers & Geosciences*, Vol. **25** (1999), No. 5, p. 581–597.

[73] Garcia, A. L., de Miras, J. R., Fieto, F. R.: "Free-Form Solid Modelling Based on Extended Simplicial Chains Using Triangular Bezier Patches." *Computers & Graphics*, Vol. **27** (2003), No. 1, p. 27–39.

[74] Tsay, D. M., Lin, B. J.: "Improving the Geometry Design of Cylindrical Cams Using Nonparametric Rational B-Splines." *Computer-Aided Design*, Vol. **28** (1996), No. 1, p. 5–15.

[75] Wang, R. -H.: "Multivariate Spline and Algebraic Geometry*1." *Journal of Computational and Applied Mathematics*, Vol. **121** (2000), No. 1–2, p. 153–163.

[76] Hinton, E., Ozakca, M., Rao, N. V. R.: "Free Vibration Analysis and Shape Optimization of Variable Thickness Plates, Prismatic Folded Plates and Curved Shells: Part 2: Shape Optimization." *Journal of Sound and Vibration*, Vol. **181** (1995), No. 4, p. 567–581.

[77] Yu, T. Y., Soni, B. K.: "Application of NURBS in Numerical Grid Generation." *Computer-Aided Design*, Vol. **27** (1995), No. 2, p. 147–157.

[78] Zheng, Y., Lewis, R. W., Gethin, D. T.: "Three-Dimensional Unstructured Mesh Generation: Part 2. Surface Meshes." *Computer Methods in Applied Mechanics and Engineering*, Vol. **134** (1996), No. 3–4, p. 269–284.

[79] Mastin, C. W.: "Three-Dimensional Bezier Interpolation in Solid Modeling and Grid Generation*1." *Computer Aided Geometric Design*, Vol. **14** (1997), No. 9, p. 797–805.

[80] Natekar, D., Zhang, X., Subbarayan, G.: "Constructive Solid Analysis: a Hierarchical, Geometry-Based Meshless Analysis Procedure for Integrated Design and Analysis." *Computer-Aided Design*, Vol. **36** (2004), No. 5, p. 473–486.

[81] Keyser, J. et al.: "Efficient and Exact Manipulation of Algebraic Points and Curves." *Computer-Aided Design*, Vol. **32** (2000), No. 11, p. 649–662.

[82] Hohenberger, W., Reuding, T.: "Smoothing Rational B-Spline Curves Using the Weights in an Optimization Procedure." *Computer Aided Geometric Design*, Vol. **12** (1995), No. 8, p. 837–848.

[83] Dekanski, C. W., Bloor, M. I. G., Wilson, M. J.: "A Parametric Model of a 2-Stroke Engine for Design and Analysis." *Computer Methods in Applied Mechanics and Engineering*, Vol. **137**(1996), p. 411–425.

[84] Ugail, H., Bloor, M. I. G., Wilson, M. J.: "Manipulation of PDE Surfaces Using An Interactively Defined Parameterisation. *Computers & Graphics*, Vol. **23** (1999), p. 525–534.

[85] Du, H., Qin, H.: "A Shape Design System Using Volumetric Implicit PDEs." *Computer-Aided Design*. In Press, Corrected Proof.

[86] Ugail, H., Wilson, M. J.: "Efficient Shape Parametrisation for Automatic Design Optimisation Using a Partial Differential Equation Formulation." *Computers & Structures*, Vol. **81** (2003), No. 28–29, p. 2601–2609.

[87] Bloor, M. I. G., Wilson, M. J. I.: "Generating Parametrizations of Wing Geometries Using Partial Differential Equations." *Computer Methods in Applied Mechanics and Engineering*, Vol. **148** (1997), p. 125–138.

[88] Bloor, M. I. G., Wilson, M. J. I.: "The Efficient Parametrization of Generic Aircraft Geometry." *Journal of Aircraft*, Vol. **32** (1995), No. 6, p. 1269–1275.

[89] Sevant, N. E. et al.: "The Automatic Design of a Generic Wing/Body Fairing." In CFD 95: Third Annual Conference of the CFD Society of Canada. 1995, Banff, Canada.

[90] Siu, Y. K., Tan, S. T.: "‚Source-Based' Heterogeneous Solid Modeling." *Computer-Aided Design*, Vol. **34** (2002), No. 1, p. 41–55.

[91] Kumar, V. et al.: "A Framework for Object Modeling." *Computer-Aided Design*, Vol. **31**(1999), No. 9, p. 541–556.

[92] Batory, D. S., Kim, W.: "Modeling Concepts for VLSI CAD Objects." *ACM Transactions on Database Systems* (TODS), Vol. **10** (1985), No. 3, p. 322–346.

[93] Rappoport, A.: "Geometric Modeling: A New Fundamental Framework and its Practical Implications." In: Proceedings of the 3rd ACM Symposium on Solid Modeling and Applications. 1995, Salt Lake City, Utah, USA: ACM Press, NJ, USA.

[94] Webber, S. J.: "An Expert System for Hydraulic Fracturing." In: School of Petroleum and Geological Engineering. 1994, University of Oklahoma.

[95] Li, Y.: "An Engineering Constraint Handling and Optimisation System for Integrated Product Development." In: Industrial Engineering. 1998, Wayne State University: Detroit.

[96] Chau, K. W., Albermani, F.: "A Coupled Knowledge--Based Expert System for Design of Liquid-Retaining Structures." *Automation in Construction*, Vol. **12** (2003), No. 5, p. 589–602.

[97] Gantovnik, V. B. et al.: "A Genetic Algorithm with Memory for Mixed Discrete-Continuous Design Optimization." *Computers & Structures*, Vol. **81** (2003), p. 2003–2009.

[98] Lee, D., Kim, S. -Y.: "A Knowledge-Based Expert System as a Pre-Post Processor in Engineering Optimization." *Expert Systems with Applications*, Vol. **11** (1996), No. 1, p. 79–87.

[99] McCorkle, D. S., Bryden, K. M., Carmichael, C. G.: "A New Methodology for Evolutionary Optimization of Energy Systems." *Computer Methods in Applied Mechanics and Engineering*, Vol. **192** (2003), No. 44–46, p. 5021–5036.

[100] Walker, M., Smith, R. E.: "A Technique for the Multiobjective Optimisation of Laminated Composite Structures Using Genetic Algorithms and Finite Element Analysis." *Composite Structures*, Vol. **62** (2003), No. 1, p. 123–128.

[101] Bos, A. H. W.: "Aircraft Conceptual Design by Genetic/Gradient-Guided Optimization." *Engineering Applications of Artificial Intelligence*, Vol. **11** (1998), No. 3, p. 377–382.

[102] Zakarian, V. L., Kaiser, M. J.: "An Embedded Hybrid Neural Network and Expert System in a Computer-aided Design System." *Expert Systems with Applications*, Vol. **16** (1999), No. 2, p. 233–243.

[103] Leung, R. W. K., Lau, H. C. W., Kwong, C. K.: "An Expert System to Support the Optimization of Ion Plating Process: an Olap-Based Fuzzy-Cum-Ga Approach." *Expert Systems with Applications*, Vol. **25** (2003), No. 3, p. 313–330.

[104] Lu, Jingui, et al.: "An Improved Strategy for GAs in Structural Optimization." *Computers & Structures*, Vol. **61** (1996), No. 6, p. 1185–1191.

[105] Le Riche, R., Gaudin, J., Besson, J.: "An Object-Oriented Simulation-Optimization Interface." *Computers & Structures*, Vol. **81** (2003), No. 17, p. 1689–1701.

[106] Ko, D. -C., Kim, D. -H., Kim, B. -M.: "Application of Artificial Neural Network and Taguchi Method to Preform Design in Metal Forming Considering Workability." *International Journal of Machine Tools and Manufacture*, Vol. **39** (1999), No. 5, p. 771–785.

[107] Hauser, M., Scherer, R. J.: “Application of Intelligent CAD Paradigms to Preliminary Structural Design.” *Artificial Intelligence in Engineering*, Vol. **11** (1997), No. 3, p. 217–229.

[108] Chun, H. W.: “Automatic Simulation Program Synthesis Using a Knowledge-Based Approach.” *Simulation Practice and Theory*, Vol. **5** (1997), No. 6, p. 473–488.

[109] Soremekun, G. et al.: “Composite Laminate Design Optimization by Genetic Algorithm with Generalized Elitist Selection.” *Computers & Structures*, Vol. **79** (2001), No. 2, p. 131–143.

[110] Brekelmans, R. et al.: “Constrained Optimization Involving Expensive Function Evaluations: A Sequential Approach*1.” *European Journal of Operational Research*. In Press, Corrected Proof.

[111] de Silva Garza, A. G., Maher, M. L.: Design by Interactive Exploration Using Memory-Based Techniques.” *Knowledge-Based Systems*, Vol. **9** (1996), No. 3, p. 151–161.

[112] Manohar, P. A., Shivathaya, S. S., Ferry, M.: “Design of an Expert System for the Optimization of Steel Compositions and Process Route.” *Expert Systems with Applications*, Vol. **17** (1999), No. 2, p. 129–134.

[113] Giannakoglou, K. C.: “Design of Optimal Aerodynamic Shapes Using Stochastic Optimization Methods and Computational Intelligence.” *Progress in Aerospace Sciences*, Vol. **38** (2002), No. 1, p. 43–76.

[114] Wallace, D. R., Jakiela, M. J., Flowers, W. C.: “Design Search Under Probabilistic Specifications Using Genetic Algorithms.” *Computer-Aided Design*, Vol. **28** (1996), No. 5, p. 405–421.

[115] Bullock, G. N. et al.: “Developments in the Use of the Genetic Algorithm in Engineering Design.” *Design Studies*, Vol. **16** (1995), No. 4, p. 507–524.

[116] Chau, K. W., Albermani, F.: “Expert System Application on Preliminary Design of Water Retaining Structures.” *Expert Systems with Applications*, Vol. **22** (2002), No. 2, p. 169–178.

[117] Linkens, D. A., Min-You, C.: “Expert Control Systems – 2. Design Principles and Methods.” *Engineering Applications of Artificial Intelligence*, Vol. **8** (1995), No. 5, p. 527–537.

[118] Liebowitz, J.: “Expert Systems: A Short Introduction.” *Engineering Fracture Mechanics*, Vol. **50** (1995), No. 5–6, p. 601–607.

[119] Renner, G., Ekart, A.: “Genetic Algorithms in Computer Aided Design.” *Computer-Aided Design*, Vol. **35** (2003), No. 8, p. 709–726.

[120] Teughels, A., De Roeck, G., Suykens, J. A. K.: “Global Optimization by Coupled Local Minimizers and its Application to Fe Model Updating.” *Computers & Structures*, Vol. **81** (2003), No. 24–25, p. 2337–2351.

[121] Shyy, W. et al.: “Global Design Optimization for Aerodynamics and Rocket Propulsion Components.” *Progress in Aerospace Sciences*, Vol. **37** (2001), No. 1, p. 59–118.

[122] Mejasson, P. et al.: “Intelligent Design Assistant (IDA): A Case Base Reasoning System for Material and Design.” *Materials & Design*, Vol. **22** (2001), No. 3, p. 163–170.

[123] Prabhu, B. S., Biswas, S., Pande, S. S.: “Intelligent System for Extraction of Product Data from CADD Models.” *Computers in Industry*, Vol. **44** (2001), No. 1, p. 79–95.

[124] Stephanopoulos, G., Han, C.: “Intelligent Systems in Process Engineering: A Review.” *Computers & Chemical Engineering*, Vol. **20** (1996), No. 6–7, p. 743–791.

[125] Lagaros, N. D., Papadrakakis, M., Kokossalakis, G.: “Structural Optimization Using Evolutionary Algorithms.” *Computers & Structures*, Vol. **80** (2002), No. 7–8, p. 571–589.

[126] Papadrakakis, M., Lagaros, N. D., Tsompanakis, Y.: “Structural Optimization Using Evolution Strategies and Neural Networks.” *Computer Methods in Applied Mechanics and Engineering*, Vol. **156** (1998), No. 1–4, p. 309–333.

[127] Dev, K., Murthy, C. S. R.: “A Genetic Algorithm for the Knowledge Base Partitioning Problem*1. *Pattern Recognition Letters*, Vol. **16** (1995), No. 8, p. 873–879.

[128] Fensel, D., Groenboom, R., de Lavalette, G. R. R.: “Modal Change Logic (MCL): Specifying the Reasoning of Knowledge-Based Systems.” *Data & Knowledge Engineering*, Vol. **26** (1998), No. 3, p. 243–269.

[129] Clibbon, K., Edmonds, E.: “Representing Strategic Design Knowledge.” *Engineering Applications of Artificial Intelligence*, Vol. **9** (1996), No. 4, p. 349–357.

[130] Maher, M.: “The Importance of Representation in AI in Design.” In: Research Directions for Artificial Intelligence in Design. 1995, Key Centre of Design Computing, University of Sydney. p. 49–54.

[131] Gantovnik, V. B., Gurdal, Z., Watson, L. T.: “A Genetic Algorithm with Memory for Optimal Design of Laminated Sandwich Composite Panels.” *Composite Structures*, Vol. **58** (2002), No. 4, p. 513–520.

[132] Fluery, C., Zhang, W. H.: “Selection of Appropriate Approximation Schemes in Multi-Disciplinary Engineering Optimization.” *Advances in Engineering Software*, Vol. **31** (2000), No. 6, p. 385–389.

[133] Schmit, L. A., Farshi, B.: “Some Approximation Concepts for Structural Synthesis.” *Journal of the AIAA*, Vol. **12** (1974), p. 692–9.

[134] Barthelemy, J. F., Haftka, R. T.: “Approximation Concepts for Optimum Structural Design – a Review. *Structural Optimisation*, Vol. **5** (1993), p. 129–144.

[135] Rodriguez, J. F. et al.: “Trust Region Model Management in Multidisciplinary Design Optimization.” *Journal of Computational and Applied Mathematics*, Vol. **124** (2000), No. 1–2, p. 139–154.

[136] Ugwu, O. O., Tah, J. H. M.: “Towards Optimising Construction-Method Selection Strategies Using Genetic Algorithms. *Engineering Applications of Artificial Intelligence*, Vol. **11** (1998), No. 4, p. 567–577.

[137] Abou, R. S. M., Shi, J.: "Automated Construction-Simulation Optimization." *Journal of Construction Engineering and Management*, Vol. **120** (1994), No. 2, p. 374–385.

[138] Paulson, B. C. J., Chan, W. T., Koo, C. C.: "Construction Operations Simulation by Microcomputers." *Journal of Construction Engineering and Management*, Vol. **113** (1987), No. 2, p. 302–320.

[139] Smith, S. D., Osborne, J. R., Forde, M. C.: "Analysis of Earthmoving Systems Using Discrete-Event Simulation." *Journal of Construction Engineering and Management*, Vol. **121** (1995), No. 4, p. 388–396.

[140] Smith, K., Palaniswami, M., Krishnamoorthy, M.: "Traditional Heuristic Versus Hopfield Neural Network Approaches to a Car Sequencing Problem." *European Journal of Operational Research*, Vol. **93** (1996), No. 2, p. 300–316.

[141] Sette, S., Boullart, L., Langenhove, L. V.: "Optimising a Production Process by a Neural Network/Genetic Algorithm Approach." *Engineering Applications of Artificial Intelligence*, Vol. **9** (1996), No. 6, p. 681–689.

[142] Ugwu, O. O., Tah, J. H. M., Howes, R.: "Optimizing Construction-Method Selection Strategies in an Integrated Decision Support Environment." In: Association of researchers in Construction Management Conference (ARCOM98) University of Reading. 1998, Reading, U.K.

Craig Peebles
e-mail: craig.peebles@optusnet.com.au

Cees Bil
e-mail: bil@rmit.edu.au

School of Aerospace
Mechanical and Manufacturing Engineering
RMIT University
GPO Box 2476V
Melbourne, VIC 3001, Australia

Lorenz Drack
lorenz.drack@dsto.defence.gov.au

Maritime Platforms Division
Defence Science and Technology Organisation
PO Box 4331
Melbourne, VIC 3001, Australia