Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1999), *13*, 351–363. Printed in the USA.
Copyright © 1999 Cambridge University Press 0890-0604/99 $12.50

Evolutionary optimization within an intelligent hybrid system for design integration

D. SU AND M. WAKELAM

Department of Mechanical and Manufacturing Engineering, The Nottingham Trent University, Burton Street, Nottingham, NG1 4BU, U.K.

(RECEIVED December 2, 1998; ACCEPTED June 3, 1999)

Abstract

An intelligent hybrid approach has been developed to integrate various stages in total design, including formulation of product design specifications, conceptual design, detail design, and manufacture. The integration is achieved by blending multiple artificial intelligence (AI) techniques and CAD/CAE/CAM into a single environment. It has been applied into power transmission system design. In addition to knowledge-based systems and artificial neural networks, another AI technique, genetic algorithms (GAs), are involved in the approach. The GA is used to conduct optimization tasks: (1) searching the best combination of design parameters to obtain optimum design of gears, and (2) optimization of the architecture of the artificial neural networks used in the hybrid system. In this paper, after a brief overview of the intelligent hybrid system, the GA applications are described in detail.

Keywords: Evolutionary Optimization; Artificial Intelligence; Hybrid System; Design Integration; Power Transmission Systems

1. INTRODUCTION

It is a tedious and time-consuming task to design a product through various stages of total design, including formulation of product design specification, conceptual design, detail design, and manufacture (Pugh, 1990). To speed up the total design process, an intelligent hybrid system (IHS) approach has been developed by the authors. The IHS approach integrates the stages within the total design process by blending artificial intelligence (AI) techniques with CAD/CAE/CAM programs and packages into a single environment. The AI techniques involved include knowledge-based systems (KBS), artificial neural networks (ANNs), and genetic algorithms. The approach has been successfully applied into power transmission system design.

Design integration and application of AI in design are important research topics and have attracted researchers' great attention, for example, Su (1994), Kaftanoglu et al. (1995), Sharpe (1995), da Silva and Dawson (1997), and Counsell (1997). In comparison with the existing research, the major advanced features of the IHS approach are: (1) integration of the total design process with a successful application of power transmission system design, and (2) integration of multiple AI with CAD/CAE/CAM into a single environment where different software techniques coexist and compliment with each other.

In the IHS, the genetic algorithms are used to optimize design parameters and to determine the best combination of the architecture parameters of ANNs. In the following sections, the genetic algorithm (GA) applications within the IHS approach will be described in detail. The overall approach and other AI techniques (KBS and ANN) involved will be briefly mentioned in the context where they are relevant, for their details please refer to the papers by Su et al. (1996), Su and Wakelam (1997), Su (1998), and Wakelam (1998).

OVERVIEW OF THE IHS APPROACH

The design process and software integration are shown in Figure 1. The design process is controlled by the KBS, which consists of a system controller (SC) and several subcontrollers (SubC). The SubC controls the activities within a particular stage such as conceptual design, detail design, or

Reprint requests to: D. Su, Department of Mechanical and Manufacturing Engineering, The Nottingham Trent University, Burton Street, Nottingham, NG1 4BU, U.K. Phone: +44 (0)115 98482306; Fax: +44 (0)115 9486506; E-mail: daizhong.su@ntu.ac.uk; Web site: http://domme.ntu.ac.uk/mechdes.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

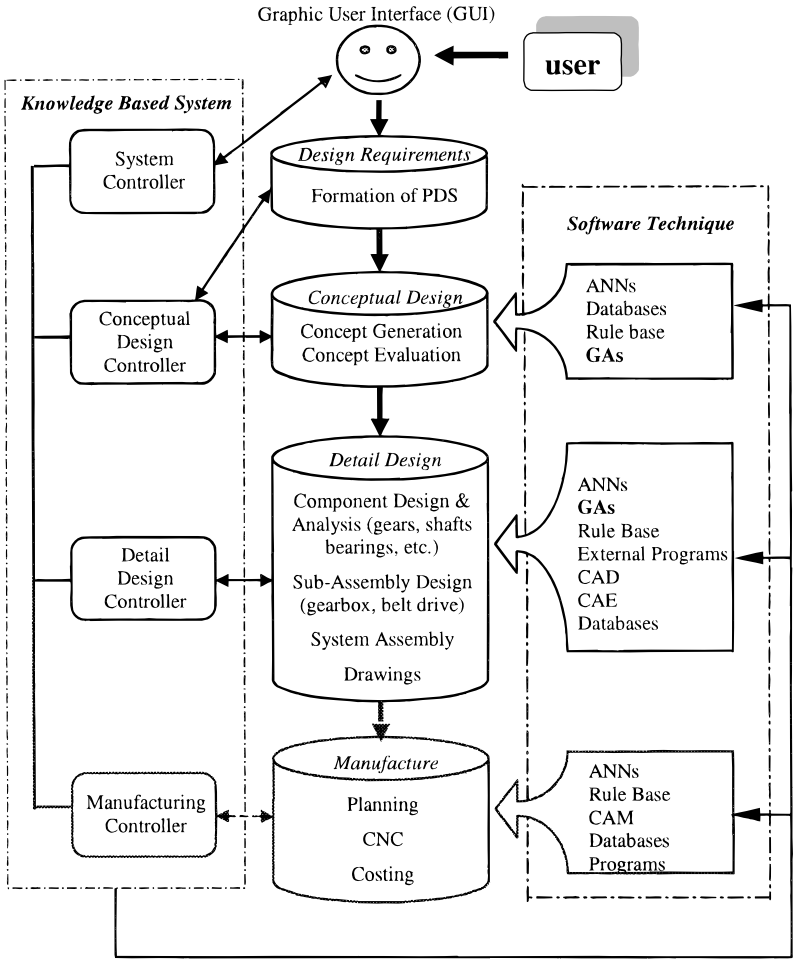


Fig. 1. Design process and software integration.

manufacture; while the SC controls the overall process by communicating with the SubCs. All the SC and SubCs are rule-based systems.

As for the software techniques involved, the design expertise is captured by the KBS and ANNs, and GAs are used for optimization; other tasks within the total design process, such as numerical analysis, engineering drawing and data processing, are conducted using relevant CAD/CAE/CAM software packages. The KBS communicates with the others and works as a coordinator.

As an application of the approach, an IHS for total design of power transmission systems has been developed. The overall design process is controlled by a master KBS developed using C^{++}, in conjunction with a friendly graphical user interface (GUI) developed using Visual BASIC. Its functions are briefly described below:

(a) ***Formulation of Product Design Specification (PDS)***. The PDS items are specified by the user via the GUI. The PDS are two types: initial requirements such as transmission power, orientation of input/output shafts, speed ratio, center distance, etc.; and evaluation criteria including size, manufacture cost, easy manufacture, etc. The system has a mechanism for further development to add more PDS items. The GUI also interfaces with the databases to store and retrieve the specified PDS.

(b) ***Conceptual Design***. The concepts to be constructed by the IHS fall in the following range

- stages of the transmission: one, two, or three;
- orientation of input/output shafts: parallel, cross, or perpendicular;
- components at each stage of the transmission: seven types of components including gears, belts, and trains.

In this phase of the design, four ANNs are used to generate the concepts. Based on the output results from the ANNs, the KBS makes a decision to select the best concept for detail design, see Su (1998) for detail.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

(c) ***Detail Design***. The selected concept is decomposed into subsystems. The detail design of each subsystem is conducted first, and then the completed subsystems are assembled together to form a final design. At this stage, conflicts may occur and the KBS has the capabilities to resolve the conflicts by redesign. Currently, the IHS can handle the detail design of two types of subsystems: gearboxes and belt drives. The final resultant design can be one of them, or combination of both. The tasks conducted in detail design include: gear strength analysis, bearing selection, shaft design, case design, belt and pulley selection, design optimization, component and assembly drawings, and parametric design of components.

(d) ***Manufacture***. In this phase, the IHS can produce cost analysis of the designed system, process planning for manufacture of major components, and CNC programs for manufacture of shafts.

Within the system, GA has been applied for two tasks: *optimizing the parameters of gear design* within the detail design stage and *searching for the best combination of architecture parameters for the ANNs* used in both conceptual and detail design stages. They are further detailed below.

3. OPTIMIZATION OF GEAR DESIGN PARAMETERS

The optimization of gear design requires the adjustment of several factors that affect the performance of the gears. These include the general dimensions and parameters such as the material property, number of teeth, pitch circle diameter, and tooth facewidth together with more detailed modifications such as shifting the tooth profile. Optimization of the gear performance is achieved by modification of these parameters. However, as the effects of the parameters are directly and indirectly related, the search area to obtain an optimum is large. Due to a lack of direct relationships between some parameters and the complexity of the search area, conventional search techniques, such as hill climbing and the Newton–Raphson method, would have difficulty in achieving a global optimum. A solution is to apply the adaptive search technique of GA to gear design optimization, enabling multiple parameters to be simultaneously adjusted. The ability to adjust indirectly related parameters simultaneously is a significant advantage of the GA, which enables the accumulative effect of multiple modifications to be evaluated without manual intervention from an external source. This is a feature not found to be available from previous work.

The optimization process adjusts parameters that define the characteristics of the gear to fulfil the following criteria:

- Achieve the minimum facewidth and module while complying with professional standards such as ISO and BS—not exceeding the permissible bending and contact stress on the teeth.
- Maximum bending stresses within both the pinion and wheel gears are approximately equal.
- Contact ratio is to be maximized to reduce vibration and noise.
- Speed ratio is to be maintained.
- Center distance of gear pair to be minimized for variable and maintained for fixed center distances.

These criteria form the basis for the fitness functions that determine the success of the configuration of parameters. The fitness functions are defined in Section 3.3.

Before the GA is applied to the problem it is advisable to establish the purpose of the various parameters. This enables the fitness functions to be sculpted to suit the application and to ensure that only critical parameters are encoded into the genome. The critical parameters considered are the tooth width, module, pressure angle, helix angle, and number of teeth in pinion and wheel. These parameters have a profound effect on the performance of the gear pair. Additional parameters are included in the optimization, which subtly alter the performance of the gear. These include the addendum coefficient (determines the length of the teeth), the addendum modification coefficients for the pinion and wheel (shifts the profiles of the teeth along the involute, relative to the reference circle), and the rack tip radius (determines the tooth's root radius, a difficult value to determine). Adjustment of these nine parameters allows the performance of the gear pair to be fine tuned to suit the desired goals.

3.1. Genome encoding

The method of encoding the information within a gene has a dramatic effect upon the performance of the optimization process. The value variation of the parameters to be optimized for gear design are of three categories:

- *Category 1—small range of variation*. Rack tip radius is a typical example of this type: in some cases, it is ranged from 0 to 0.75 mm; while in other cases, it may be any real number between 0.7 to 1.5 mm.
- *Category 2—large range of variation*. For example, toothwidth may be any real number ranging from a small value such as 20 mm up to a large value such as 300 mm or even a larger number.
- *Category 3—discrete values or integer numbers*. Examples of such parameters are *module* and *number of teeth*. For a standard *module*, the values are defined by professional standards (BS, ISO, DIN, etc.) as 0.4, 0.5, 1.0, 1.25, . . . , that is, a predefined list. As for *number of teeth*, it has to be a positive integer.

To encoding the different types of parameters, three methods have been developed accordingly: direct, percentage deviation of the value and position within a predefined list.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

(a) ***Direct***. The direct method is the most straightforward way of encoding the value within a gene for a parameter whose value variation is within a small range. The value is encoded directly into the gene either in its entirety or as a proportional value that is manipulated after decoding. For example, from a range of 0 to 31, dividing by 40 alters the range giving 0 to 0.775. If 0.8 is then added to the value the range is moved to between 0.8 and 1.575.

It has to be noticed that this method is not suitable for parameters of category 2 mentioned above, because it will take a large amount of computer memory. To overcome this problem, the *Percentage Deviation* method is a solution.

(b) ***Percentage Deviation***. This method of representing the encoded value within each gene requires two stages of decoding to extract the information. This enables the value to be stored in a condensed fashion and is ideally suited for use with initial starting points or values. This method is an alternative to storing the entire value directly within the gene in binary form, as only a percentage deviation from an initial starting point is stored, thus enabling the size of the gene to be considerably reduced for large values. The principle of the percentage deviation method is as follows: the initial encoded values within the population cover a band of possible solutions about either side of an initial start-point. This is illustrated in Figures 2 and 3 below.

The encoded deviations are contained within the binary strings of the genes. For example, a 5- or 7-bit length gives a range of 0 to 31 or 0 to 127, respectively. The range is then effectively doubled by the addition of an extra bit to indicate the polarity of the deviation, + or −. Decoding the gene and applying the percentage deviation to the start-point value produces the new value corresponding to the search. The upper and lower limits represent the bounds of the search area in which the parameters are retained.

(c) ***Pre-defined List***. The third method of encoding information is by referring to a list of predefined values, that is, which is suitable for the parameters of category 3 mentioned above. As shown in Figure 4, a pointer is moved up or down the list from a reference point to identify the position that contains the information. The position of the pointer relative to the start-point is determined by the information contained within the gene. The decoded value from the gene relates to the number of positions within the list that the pointer is moved to. The encoded value directly represents the number of spaces shifted up (+) or down (−) the list. The process, illustrated in Figure 4, shows how the gene with position 6 relates to the movement of the pointer with −5 deviation from the start-point within the predefined list. The full extent of the deviation range and the list are also given in the figure, illustrating that the movement of the pointer is restricted within the boundaries of the list.

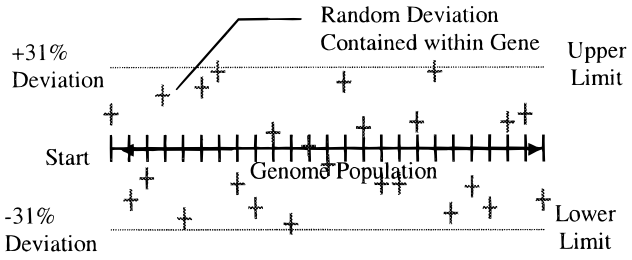


Fig. 2. Deviation about initial start point.

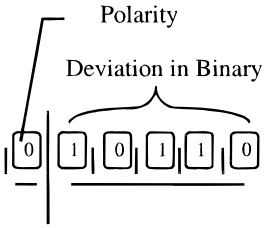


Fig. 3. Gene structure.

Limit values are imposed on the parameters to be optimized to ensure that the optimization process does not cause the values to drift into regions where the result would be invalid. These limits are maintained and implemented by production rules at the preprocess stage of the GA. The method of limiting the parameter values used is as follows. The value contained within a gene is first decoded then preprocessed to produce the value of the parameter. If this value exceeds the limits imposed the value is set to its relative limit. The action of constraining the value should not affect the GA process with regard to regeneration fitness levels, due to the method of encoding. As the encoded information is the deviation and the optimized value is the decoded resultant, the limit will produce the desired output provided the fitness for the limit is high. This characteristic of the gene that caused the high fitness will then be passed to the next generation.

3.2. Genome construction

The parameters that correspond to the genes within a genome have been identified earlier. These genes combine to form either a seven or nine gene genome, which are represented in Figures 5 and 6. As can be seen, the number of genes is dependent upon the application: for a fixed center distance, there are seven genes in the genome; while for a flexible center distance, extra two genes are added at the end of the genome.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

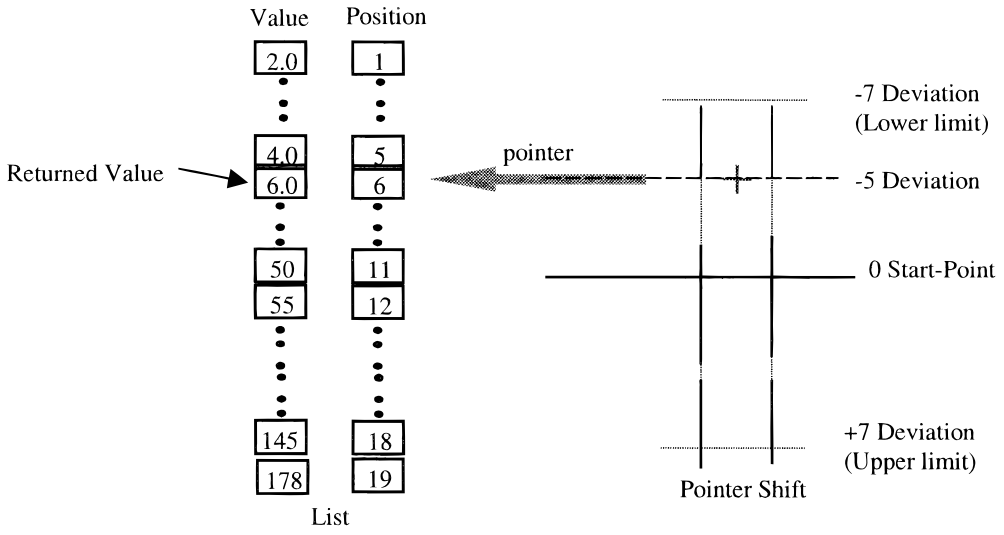


Fig. 4. Predefined list.

It is the purpose of the GA to find the combination of parameters that will produce the optimum design. However, some consideration is required with regard to the method of encoding the relevant information into the genome.

Two configurations of genome have been constructed for the two different types of gear design: fixed center distance and variable center distance. Due to the method of calculation, for the optimization of the fixed center distance, the number of teeth for the pinion $z1$ and wheel $z2$ are determined by the value of the module and the speed ratio (speed ratio remaining constant). The difference in center distance is compensated for by the profile shift of the teeth. For the profile shifts to be determined the amount that one of the profiles will be shifted must be known. The value of the pinion profile shift, $x1$, is included within the genome allowing the value of the wheels shift, $x2$, to be calculated.

For the variable center distance, the relation between the pinion and the wheel is limited to the speed ratio. Due to the lack of relationships, the number of teeth and amount of profile shift on the teeth cannot be calculated from the existing parameters, therefore these parameters are included within the genome, forming additional dimensions to the search.

The genome has been constructed in such a way that fixed and variable center distance gear designs can use the same genome. Parameters that are used within both designs form the first seven genes of the genome, while the two extra genes, x_2 and z_1, required for the variable center distance are located at the end. This structure allows the additional parameters to be ignored when not required. Their permanent inclusion within the genome will not affect the GA process as they are ignored and during crossover the positions of the transferred bits are maintained, therefore never letting the contents of these genes affect the search.

The method of encoding the information within the genes is dependent upon the effect that the parameters has upon the design. These considerations include limits, set values, and resolution. Limits have been set on several parameters to prevent them from producing values that will result in certain failure of the design or undesirable values. Table 1 contains some limited parameters and their restrictions.

3.3. Fitness functions

The fitness function, as mentioned earlier, has a dramatic effect upon the convergence of the search and the parameters contained within genomes that are transferred through

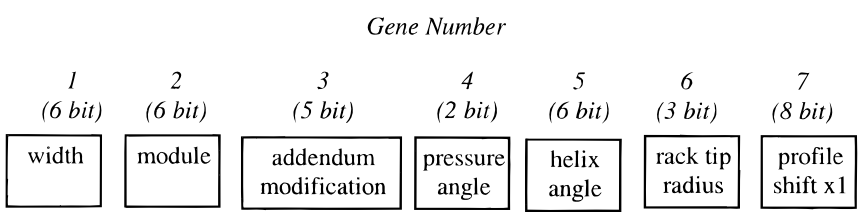


Fig. 5. Genome construction of fixed central distance.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

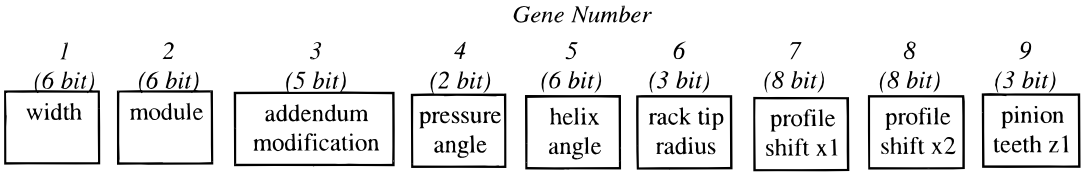


Fig. 6. Genome construction of flexible central distance.

the generations. The fitness functions generate the fitness rating for each genome and correspond to design solutions. Therefore, the selection of criteria that the fitness function comprise of must mirror the desired characteristics of the target design.

Due to the nature of gear design, there are two types of criteria for design evaluation:

- *Conditional criteria*. This type of criteria must be met, otherwise, the design fails. For example, the tooth bending and contact stresses must not exceed their allowable levels. If this condition has been met, then the design is acceptable; if not, the design fails. There are only two evaluation results against this type of criteria: *failure* or *pass (not failure)*.
- *Rating criteria*. When a design is evaluated against this type of criteria, the result falls within a range giving a degree to indicate how good the design is. For example, for the criterion "minimizing the toothwidth," if the rating marks ranging from 1 to 10 for the actual toothwidth ranging from 50 to 100 mm, a design with toothwidth of 60 mm is then evaluated with a rating mark of 8.

Among the criteria mentioned at the beginning of Section 3, criteria "*not exceeding permissible stresses*" and "*maintaining speed ratio*," are condition criteria, and the rest are rating criteria. To reflect the two types of criteria, two categories of functions, either a *gradient* or *step* function, are applied to determine the level of fitness.

The gradient functions are used by the fitness rating criteria that determine the fitness of the noncritical design characteristics. These characteristics do not directly cause failure of the design, but do influence its performance and guide toward the optimization targets to be achieved. The range given for each rating criteria is from 0 to a maximum value, the greater the fitness, the higher the value. Figure 7 and Table 2 below illustrates the principle.

The second category of fitness functions relates to critical characteristics of the design that directly influence or cause failure of the design. These fitness functions form *conditional criteria* and apply a step function to the level of fitness. As failure to meet the requirements of the function will result in failure of the design; these fitness functions have the effect of causing the overall fitness value of the genome to be drastically reduced thus encouraging removal from the population. Figure 8 and Table 3 illustrate the principle. The penalties are applied to the total fitness of the genome, reducing its value and thus its probability of transferring to the next generation.

The fitness values applied to the roulette wheel often require modification, due to the possible close proximity of some fitness values. This close proximity of genome fitness increases as the population begins to converge causing only small deviations in the fitness. Therefore, the fitness values for two different genes will cause little difference to the size of the respective roulette wheel segments. To overcome this problem, scaling methods are applied to the fitness values to emphasize the relative fitness of the genome within the population.

Several methods of scaling exist, using the sorted fitness order of the genomes and the actual fitness value. The method of linear ranking function described by Gen and Chang (1997) is applied in this work. The function combines the actual fitness value together with an element relating to its ranked position. The fitness is given by:

$$p_k = q_k + (k - popsize)\left(\frac{q}{1 - popsize}\right)$$

Table 1. *Parameter limits*

Parameter	Upper Limit	Lower Limit
Module	Position 34 within list	Position 0 within list
Addendum modification coefficient (×1 and ×2)	1.0	$x_{\min} = h_a^* \frac{\text{Min no. teeth} - \text{No. teeth}}{\text{Min. no. teeth}}$
Helix angle (angle limited)	User defined (typically <45°)	0
Helix angle (force limited)	$\sqrt{2}$.Tangential force	0

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

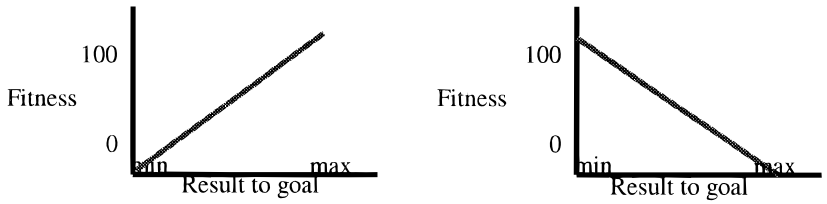


Fig. 7. Gradient fitness functions.

where

p_k = scaled fitness;

k = ranked position;

$popsize$ = population size;

q = fitness of highest ranked genome; and

q_k = fitness direct from fitness function.

This function is used in conjunction with the deviation in fitness, thus providing greater resolution to the selection roulette wheel. The effect of this is to define a noticeable ranking order without encouraging super convergence, caused by excessively biased scaling of fit genomes. This allows the search to slowly settle at first, then apply pressure to converge on a solution as the generations advance.

3.4. Case study

The optimization process is evaluated by performing numerous tests to check the controlling factors (crossover and mutation levels and convergence criteria) of the GA, the repeatability, and robustness of the process and the accuracy of the results.

The optimization has been set to achieve three goals that form the rating criteria to guide the search. These are in addition to the conditional criteria that will ensure failure of the design will not occur. The rating criteria cover:

- reduction of the gear pair size (center distance and facewidth);
- increase contact ratio; and
- equalize tooth-bending stresses of the pinion and wheel.

The case study given below is for a helical gear pair that is required to transfer 40 kW of power at 1450 rpm. The application of the gear is for moderate shock conditions, which is represented by an application factor of 1.4. The results given below compare the performance of the initial gear design with that produced from the GAs optimization (Table 4).

The geometric goals of the search have been achieved. The facewidth of the gear and the center distance have been reduced by 10.56 and 0.33%, respectively. The reduction in center distance is not dramatic, but as its increase would aid the other criteria in achieving their goals, its reduction is a sign that the GA process is working.

One of the main targets of the optimization was to increase the contact ratio of the gear pair. Its purpose is to reduce vibration and thus noise, fatigue, and shock loads. This has been achieved, improving on an initial design that already has a high contact ratio. This can be seen from the results shown in Table 5.

As can be seen from the stresses acting within the gear teeth, the improvements to the design have been achieved without reducing the margin of material safety. Yet the lengthening of the teeth by 17.78% (as shown by the increase in tooth depth) should increase the bending stress. In addition, the bending stresses acting on the teeth are within 5.33% of each other (an improvement of 1.34%) thus promoting equal wear and fatigue of the teeth. The GA has therefore modified several parameters to ensure that the design will not fail. The removal of failing designs from the search has thus encouraged stronger designs.

The optimization process was repeated several times to ensure that the final solutions were the same and thus the process displayed repeatability of results.

Table 2. *Fitness rating criteria*

Fitness Rating Criteria	Max. Fitness (max. Value)	Min. Fitness (0)
Result to goal 1	Smallest Value	Largest Value
Result to goal 2	Largest Value	Smallest Value

Table 3. *Fitness conditions*

Fitness Conditional Criteria	Pass	Penalty
Result to goal 1	$\geqq$ Permissible value	$<$ Permissible value
Result to goal 2	$\leqq$ Permissible stress	$>$ Permissible stress

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

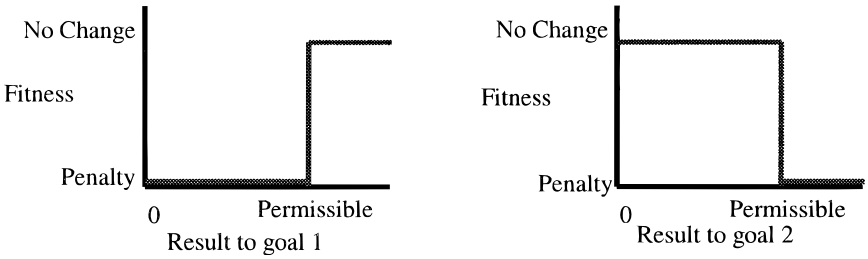


Fig. 8. Step fitness function.

4. OPTIMIZATION OF ANN ARCHITECTURES

As mentioned in Section 2, there are several ANNs in the IHS and all of them are back-propagation networks. Training this type of network has proved to be a difficult process and requires expertise in the process, due to the lack of effective rules and guides. To overcome this problem, a GA-based approach has been developed, which implements the optimization capabilities of a GENetic algorithm to define the factors that determine a NEUral networks performance (GEN-NEU).

Researchers have been making efforts to apply GA into the optimization of back-propagation networks. For example, Caudell and Dolan (1989) used them to adjust the connective weights during the training process to improve the networks convergence, and Miller et al. (1991) and Maniezzo (1994) have applied them to the optimization of networks topologies. However, the adjustment of the connective weights by implementing GAs does not provide significant improvements in network performance over the back-propagation technique.

The GEN-NEU approach takes into account three major factors that affect network performance: topology, transfer function, and training period. Optimization of these factors is performed simultaneously, considering their combined effects upon performance and convergence, thus making the optimization more effective.

4.1. Adaptive optimization process of the GEN-NEU approach

The basic principle of the GEN-NEU approach, developed for this research, is to adjust the factors required by the back-propagation training technique based upon the performance (fitness) of the network being trained. Figure 9 demonstrates the process.

The approach encodes the values for the factors that influence the performance of the network, including the transfer function, topology, and training period into separate genes within a genome. Upon initiation, the values contained within the genes are randomly set from values within the search space. The networks corresponding to the information contained within all the genomes of the population are trained and tested then sorted into a descending order. The fitness value of the genome is determined from the network output response to a series of test cases applied after training. The lower the root means squared (RMS) error between the target and the networks output the fitter the genome, which in

Table 4. *Comparison of optimized gear pair geometry with initial design*

Geometry	Initial	Optimized	Difference
Module (mm)	2.00	2.00	0
Alpha (Deg)	20.00	20.00	0
Beta (Deg)	15.00	21.00	+6
Pinion teeth	24	23	−1
Facewidth	40.00	35.78	−10.56%
Pinion tooth depth	4.5	5.3	+17.78%
Wheel tooth depth	4.5	5.3	+17.78%
Pinion rack tip radius (mm)	0.50	0.4	−0.1
Wheel rack tip radius (mm)	0.50	0.4	−0.1
×1	0.00	0.00	—
×2	0.00	0.30	—
Center distance (mm)	124.23	123.82	−0.33%

Table 5. *Comparison of optimized gear pair performance with initial design*

Performance	Initial	Optimized	Difference
Contact Stress Pinion (% below permissible)	5.10	12.22	+7.12
Contact Stress Wheel (% below permissible)	10.79	17.48	+6.69
Bending Stress Pinion (% below permissible)	10.34	29.68	+19.34
Bending Stress Wheel (% below permissible)	16.84	27.1	+10.26
Equal Bending Stresses (% difference)	6.67	5.33	−1.34
Contact Ratio	3.29	3.95	+20.06%

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

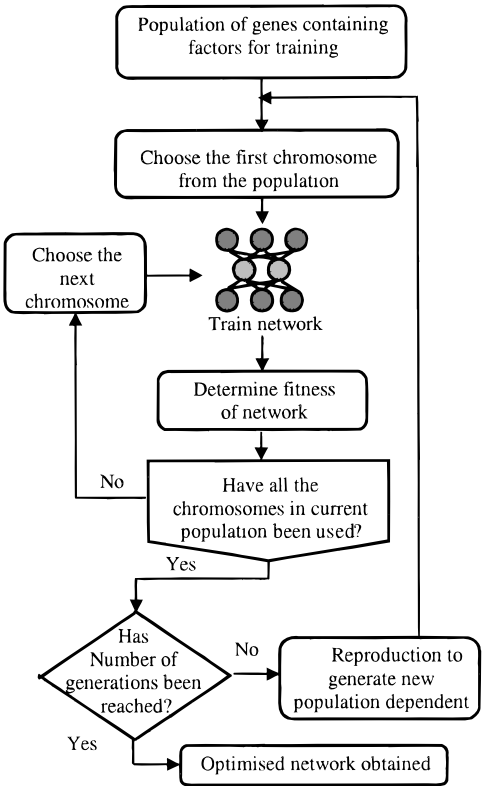


Fig. 9. The GA process of ANN training.

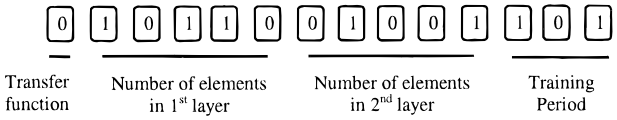


Fig. 10. A genome structure.

turn determines the genomes' probability of reproduction. Each genome's fitness rating, relative to the rest of the population, is proportional to its probability of reproduction. Thus, fitter genomes are given a greater chance of transferring their genes to the next generation, while unfit genomes are gradually removed. The fitness value is, therefore, taken as the reciprocal of the network error ensuring the fitter genomes get a greater probability of reproduction. The scaling function, described in Section 3.4, is then applied to enhance the performance of the GA.

The genomes reproduce to form the majority of the next generation. Reproduction accounts for approximately 98% of the next generation. The remainder of the next generation comprises of random selections from the current population that pass unaltered through to the next generation.

Once the next generation is established a mutation operator is applied to the population with a probability of 5%. The mutation operator applied alters the encoded information contained within the complete genome as opposed to the typical bit mutation operator (Gen & Cheng, 1997). Total mutation produces a new member to the population allowing the search to rapidly ingress into new areas of the search space without extending the population size. Thus, the total area covered by the search adaptively increases as the process progresses. Mutation also helps prevent localized minimums limiting the scope of the search. As the effects of the mutation can be dramatic upon the search the probability of mutation is set low to limit disruption once convergence commences upon a solution. The levels of reproduction and mutation are modified by trial and error to determine the best configuration for the application.

Once crossover and mutation have been performed, the new generation is complete and ready for the new fitness ratings to be determined. After the final generation the network corresponding to the fittest genome becomes the resultant network ready for application and all the connective weights, topology and transfer functions recorded.

The number of generations that the GA performs is set to a finite amount instead of using the convergence of the population. This is due to the effect of the random initialization of network connections at the beginning of training, which can lead to variations in performance. These variations can disrupt the GA convergence process; therefore, the fixed length prevents excessive computational expense for a small increase in network performance.

The information relating to the factors that affect the performance of a network during training are encoded into the genes within the genome in binary form. An example is shown in Figure 10. The decoded values of which correspond to a sigmoid transfer function, 16 elements in the first hidden layer and 9 in the second with a training factor of 5.

Binary coding has been selected as opposed to others, such as Gray coding, for two reasons. First, the type of coding will have little effect on the performance of the GA, as the genomes comprise multiple genes whose information does not necessarily relate directly to one another. Second, crossover is performed at random points during reproduction allowing genes to be spliced, resulting in significant changes in the encoded values.

The length of the training period for different applications has a direct influence on the performance of the network. Therefore, integration of the training period into the GA optimization approach will aid successful training and reduce user interaction with the training of the network.

4.2. Application of the GEN-NEU method: An example

The analysis of the training and results of the network for determination of the gear tooth face load factor for contact stress calculation, $K_{H\beta}$, will demonstrate how the GEN-NEU approach to training back-propagation networks has been applied to the integrated design system. As the approach unfolds and the network is trained, the various stages of the GEN-NEU approach are explained. The require-

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

ments of the user describing the application and the preparation of the training data are defined together with an evaluation of the resultant trained network. The performance of the GEN-NEU process and results are analyzed and for the purpose of this example a comparison with a conventionally trained network using Professional II (Neuralware, 1993), a commercial ANN package, is made.

(a) ***Data Preparation***

The information to be encapsulated within the network is contained in the graphical design aid shown in Figure 11. The graph contains a large amount of information that would be difficult to interpret into a formula for computing without the original data that the graph is constructed from. As the original information is not available (as is the typical case if the graph is taken from manufacturers, literature), the training data is extracted directly from the graph.

Data from the graph is in two forms, direct and interpolated. The direct data is formed from the existing curves, while the interpolated data represents the intermediate values between. The second form requires the designer's judgement and expertise. The interpolated curves are represented by the dashed b/d_1 curves in Figure 11.

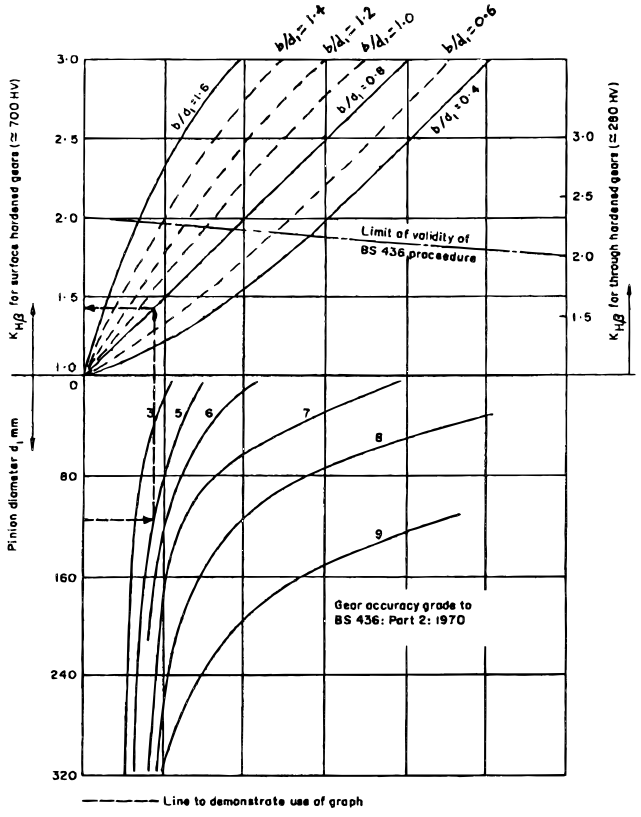


Fig. 11. Graphical design aid for $K_{H\beta}$ to be represented by network (extracted from ESDU 88033).

The training data preparation consists of determining the values of $K_{H\beta}$ (network output) for combinations of pinion diameter, gear accuracy, and facewidth ratio (network inputs). The example in Figure 11 illustrates how to use the diagram. For a pinion diameter of 115 mm, a line is projected to intersect with the curve corresponding to a gear grade of 5. From the intersection, a second line is projected up until it crosses the 0.8-facewidth ratio curve. From this intersection a final line is projected to cross perpendicular to the axis. The point at which the line crosses the axis gives the face load factor $K_{H\beta} = 1.42$.

A broad distribution of training cases within the limits of the graph is used during the preparation of the data. This helps ensure that the network represents the entire region of the graph. The structure of the training data consisted of 360 cases, covering all the accuracy grades, the direct curves for facewidth ratio and the interpolated curve for $b/d_1 = 1.2$. The test data consists of a combination of cases within the regions of the training cases and cases from the remaining interpolated curves. The performance of the network with regard to the interpolated test cases will demonstrate the networks ability to perform the application successfully.

(b) ***Results of $K_{H\beta}$ Application***

The results for the GEN-NEU training demonstrate that as the number of generations proceed the performance of the networks increase. Figure 12 illustrates the results, showing that the average error between the target and actual outputs from the network, decreases to a final value of 0.01753. At the same time, the average number of successful outputs producing an output below 2% of the target increases to 11.333. These values indicate that GEN-NEU is converging upon an optimum and its general performance is encouraging.

Performance of the networks within the GA approach is determined by the average error between the network and actual results to the test set, giving a general fitness of the network.

To remove any uncertainty in the results that could originate from effects caused by the initial random weights at the commencement of training, the resultant network is tested individually. The final network configuration obtained by GEN-NEU comprises of 3 input elements, 2 hidden layers with 22 elements in the first layer, 12 in the second layer and a single output. A sigmoid transfer function is implemented and the network was trained for 5120 passes of the training set.

These results are presented in Figure 13 and Table 6. Figure 13 illustrates the proximity of the target and actual outputs from the network. It can be seen that the general proximity of the two outputs are very close. Most importantly the proximity of the outputs to the test cases derived

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

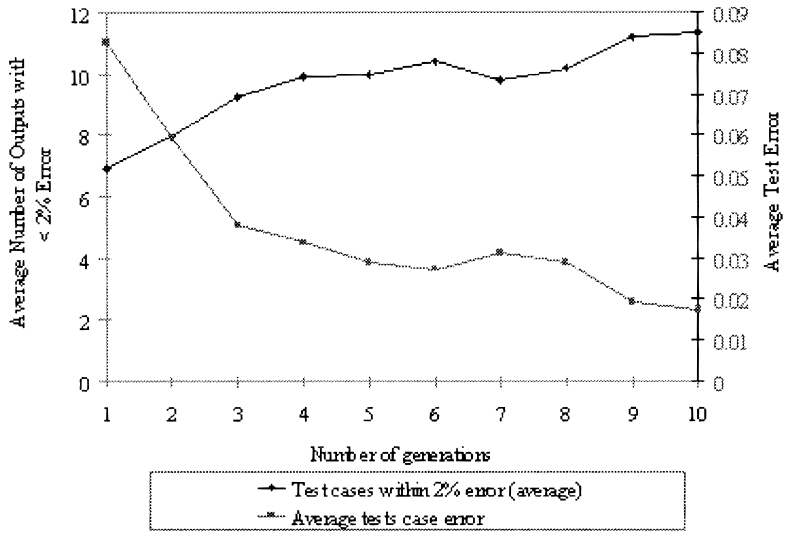


Fig. 12. GEN-NEU Training results for $K_{H\beta}$.

from interpolated curves, that network had never been presented during training are extremely close with a maximum error of 2.13%. These cases are identified by dashed boxes surrounding the points in the figure and signify that the network is capable of interpolation, producing similar results to the designer.

The performance of the network trained using the GEN-NEU program is verified by training the same network configuration under the same conditions with the commercial package Professional II. Table 6 contains the results and shows that the performance of the GEN-NEU network is better than those obtained from Professional II indicating that the network configuration is in the region of an optimum. The variation in results can be attributed to the initial random connection weights values.

5. CONCLUDING REMARKS

An intelligent hybrid approach has been developed to integrate the total design process. This is achieved by blending multiple AI and CAD/CAM/CAE into a single environment. This approach has been applied into power transmission system design. Within the approach, genetic algorithms are utilized for optimization of gear design parameters and for searching the best combination of the architecture parameters of the ANNs involved. Further to the descriptions

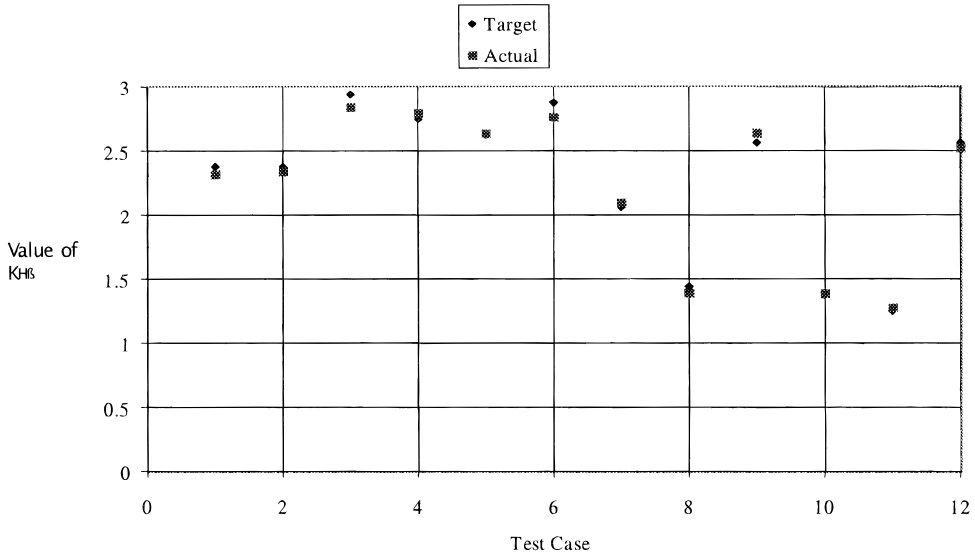


Fig. 13. Test results from $K_{H\beta}$.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

Table 6. *Comparison of GEN-NEU and professional II test results*

Input				Output			
				GEN-NEU		Professional II	
Pinion Diameter	Gear Accuracy	Facewidth Ratio	Target $K_{H\beta}$	Actual $K_{H\beta}$	% Error	Actual $K_{H\beta}$	% Error
9	3	1.6	2.3750	2.3195	2.3353	2.0707	12.8109
105	6	1.6	2.3750	2.3421	1.3860	2.3255	2.0828
207	9	1.6	2.9375	2.8429	3.2215	2.7464	6.5057
10	6	1.4	2.7500	2.7978	1.7369	2.6559	3.4213
70	7	1.4	2.6250	2.6350	0.3827	2.6813	2.1459
37	7	1.2	2.8750	2.7654	3.8137	2.7600	4.0001
159	8	1.2	2.0625	2.0982	1.7331	2.3495	13.9158
35	3	0.8	1.4375	1.3900	3.3028	1.3437	6.5260
71	8	0.8	2.5625	2.6374	2.9219	2.6392	2.9919
51	5	0.6	1.3750	1.3865	0.8384	1.4031	2.0433
320	7	0.6	1.2500	1.2766	2.1315	1.1092	11.2626
121	9	0.4	2.5625	2.5227	1.5513	2.2278	13.0623
				Average % Error	2.1129		6.7307

Shaded cells represent test cases derived from an interpolated curve that the network was not presented during training.

of the two aspects given in the previous sections, the following concluding remarks can be drawn

(a) *Concluding Remarks on the GA for Gear Design Parameter Optimization*

Presently there are several methods for the optimization of gears including the use of GAs, for example, Abersek et al. (1996). Abersek's application optimizes some of the parameters that this application addresses, as do the other gear optimization techniques. However, they do not consider modification of the tooth's profile and encourage the increase of the contact ratio for reduced vibration and noise as this application does.

In addition to parameter adjustment, when applicable, the selection of the type and magnitude of profile shift is determined. According to Niemann (1978), the magnitudes of the profile shift should be biased toward the pinion wheel being positive to avoid undercutting. From the results, it is shown that the GA has obeyed this rule, indicating that the fitness criteria do correctly define the desired result. This may be due to the limiting factors of permissible and actual stress, which form one of the main criteria from which the fitness of the design is determined. The GA optimization therefore enables designs to be improved using the desired end result as the target. Thus, intricate knowledge about the effects of individual parameters on either the final design or each other is not required, but, instead, will evolve as the search advances through the generations

(b) *Concluding Remarks on GEN-NEU*

The GEN-NEU approach to aid the training of back-propagation networks has proven to be capable of achieving successful results. As a result the laborious process of trial and error and the need for rules to determine a successful network configuration has been removed from the training process.

The implementation of the GEN-NEU method into a program provides a tool that eases network training while developing the Intelligent Integrated System and, most importantly, the GEN-NEU method enables modification to information within the system to be performed by someone, who may have little ANN experience, other than the system designer. Therefore, the GEN-NEU provides a facility for the easy alteration of knowledge, a process not easily performed with an expert system.

The inclusion of the training period into the GA search has the added beneficial effect of preventing "over training." Over training decreases the networks ability to predict and generalize outputs based upon the training data. As deviation from the test data increases when the network loses its ability to generalize, the fitness of the genomes causing this phenomenon will decrease. The result of the decreased fitness is to reduce the occurrence of this training period in the search.

Although the training data is not included within the GA, the effect of the amount of training data is compensated for by the training period and network topology. As the training period is flexible and small and large network topologies can be generated, a combination of these factors are adjusted to model the data.

GEN-NEU is a general approach, which can be used with variations of the back-propagation training technique and other supervised ANNs that require adjustment of several indirectly related factors.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press

REFERENCES

Abersek B., Flasker J., & Balic J. (1996). Expert system for designing and manufacturing of a gear box, *Expert Syst. Applications 11(3)*, 397–405.

Caudell, T.P., & Dolan, C.P. (1989). Parametric connectivity: Training a constrained network using genetic algorithms. *Proc. Third Int. Conf. Genet. Algorithms and Their Applications*, 370–374.

Counsell, M.J. (1997). Design principles for mechatronic systems. *Int. Workshop on Eng. Design*. (http://www.comp.lanes.ac.uk/publications/iced97/iced97.html).

da Silva., J.C., & Dawson, D. (1997). The development of an expert system for hydraulic systems design focusing on concurrent engineering aspects. *Int. Conf. Eng. Design*. (http://www.comp.lanes.ac.uk/publications/iced97/iced97.html)

Gen, M., & Cheng, R. (1997). *Genetic Algorithms and Engineering Design*. John Wiley & Sons, Inc., New York.

Kaftanoglu, B., Ulugul, E., & Carkoglu, N. (1995). Computer-aided optimal design strategy of power transmission systems. *Ann. CIRP 44(1)*, 83–86.

Maniezzo, V. (1994). Genetic evolution of the topology and weight distribution of neural networks. *IEEE Transact. Neural Networks, 5(1)*, 39–53.

Miller, G.F., Todd, P.M., & Hedge, S.U. (1991). Designing neural networks. *Neural Networks 4*, 53–60.

Neuralware. (1993). Professional II, Version 1.92, NeuralWare, Inc. Pittsburgh, Pennsylvania.

Niemann, G. (1978). *Machine elements: Design calculation in mechanical engineering, Vol. 2: Gears*. Springer-Verlag, New York.

Pugh, S. (1990). *Total design: Integrated methods for successful product engineering*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Sharpe, J.E.E. (1995). Computer tools for integrated conceptual design. *Design Studies 16*, 471–488.

Su, D. (1994). Gearbox KBIS: A prototype knowledge-based integrated system for gearbox design. *Proc. Int. Gear Conf.*, 201–206.

Su, D. (1998). Application of expert system and artificial neural networks for design knowledge retrieval. *Proc. Eng. Design Conf. '98*, 371–378.

Su, D., Henthorn, K.S., Jambunathan, K., & Wakelam, M. (1996). A rule-based and artificial neural network system for conceptual design. *Int. J. INGENIUM 96 (1)*, 1–8.

Su, D., & Wakelam, M. (1997). Intelligent integrated system for the design of power transmission systems. *Proc. Int. Conf. Mechanical Trans. Mech., 1–4*, 1010–1014.

Wakelam, M. (1998). *Intelligent hybrid approach for integrated design*. PhD Thesis. The Nottingham Trent University, UK.

Daizhong Su is a Reader at the Mechanical and Manufacturing Engineering Department of The Nottingham Trent University, UK. He chairs the Engineering Design and CAE subject group and leads the Concurrent Engineering Research laboratory. His research interests include artificial intelligence, evolutionary optimisation, CAD/CAM/CAE and concurrent design with about 100 refereed publications.

Mark Wakelam was a researcher at the Mechanical and Manufacturing Department of The Nottingham Trent University and received his Ph.D. from the University in 1998. He currently works in the 3D Scanners Ltd, UK. His Ph.D. thesis combines artificial neural networks, genetic algorithms and production rules into a hybrid intelligent system to integrate the various stages of the design process into a single environment.

https://doi.org/10.1017/S0890060499135054 Published online by Cambridge University Press