ELSEVIER

Engineering Structures 23 (2001) 885–901

ENGINEERING STRUCTURES

www.elsevier.com/locate/engstruct

Optimal sensor placement for fault detection

K. Worden *, A.P. Burrows

Dynamics and Control Research Group, School of Engineering, The University of Manchester, Oxford Road, Manchester M13 9PL, UK

Received 16 October 2000; received in revised form 19 October 2000; accepted 16 November 2000

Abstract

This paper describes an approach to fault detection and classification using neural networks and methods of combinatorial optimisation. Given the existence of an effective fault detection procedure, the problem arises as to how the sensors should be placed for optimal efficiency of the detector. In this paper, a neural network is used to locate and classify faults and a number of different methods are applied to determine an optimal (or near optimal) sensor distribution. A survey of recent work on sensor placement is given.

Keywords: Fault detection; Neural networks; Sensor placement; Genetic algorithms

1. Introduction

The basic problem of fault detection is to deduce the existence of a defect in a structure from measurements taken at sensors distributed on the structure. The problem is essentially one of *pattern recognition*. A number of recent papers — [1,2] are representative samples — indicate that artificial neural networks show considerable promise for fault diagnosis. In the most basic, *supervised learning*, approach to deriving a neural network, the network is presented with pairs of data vectors, the input being the vector of measurements from the system and the output being the desired fault classification. At each presentation of the data, the internal structure of the network is modified in order to bring the actual network outputs into correspondence with the desired outputs. This iterative procedure is terminated when the network outputs have the required properties over the whole training set. An exhaustive review of neural network methods in fault detection is inappropriate here as it is not the central concern of this paper; in any case, an excellent review is available [3]. For a comprehensive account of 'classical' methods of damage detection and evaluation in the context of structural engineering, the reader is referred to [4].

In [1,2] the training data was provided by finite element (FE) analysis. This has the advantage of allowing a large range of boundary conditions and static/dynamic load cases to be analysed. FE analysis is a little unrealistic as there is no limit on the spatial resolution of the data which is obtained, e.g. strains or mode-shapes. In reality, the number of sensors available will be limited and this will, of course, place restrictions on the resolution of data. As a result, it would be necessary in practice to optimise the number and location of sensors for a given problem. This paper is concerned with the specific question of optimal sensor placement for fault detection; the methods used are those of combinatorial optimisation, specifically a genetic algorithm (GA) and simulated annealing (SA).

Rytter in [4] distinguishes four levels of damage identification.

Level 1. (DETECTION) The method gives a qualitative indication that damage might be present in the structure.
Level 2. (LOCALIZATION) The method also gives information about the probable position of the damage.
Level 3. (ASSESSMENT) The method gives an estimate of the extent of the damage.
Level 4. (CONSEQUENCE) The method offers infor-

* Corresponding author. Current address: Dynamics Research Group, Department of Mechanical and Process Engineering, University of Sheffield, Mappin Street, S13JD Sheffield, UK. Fax: +44-114-2227758.

0141-0296/01/$ - see front matter
PII: S0141-0296(00)00118-8

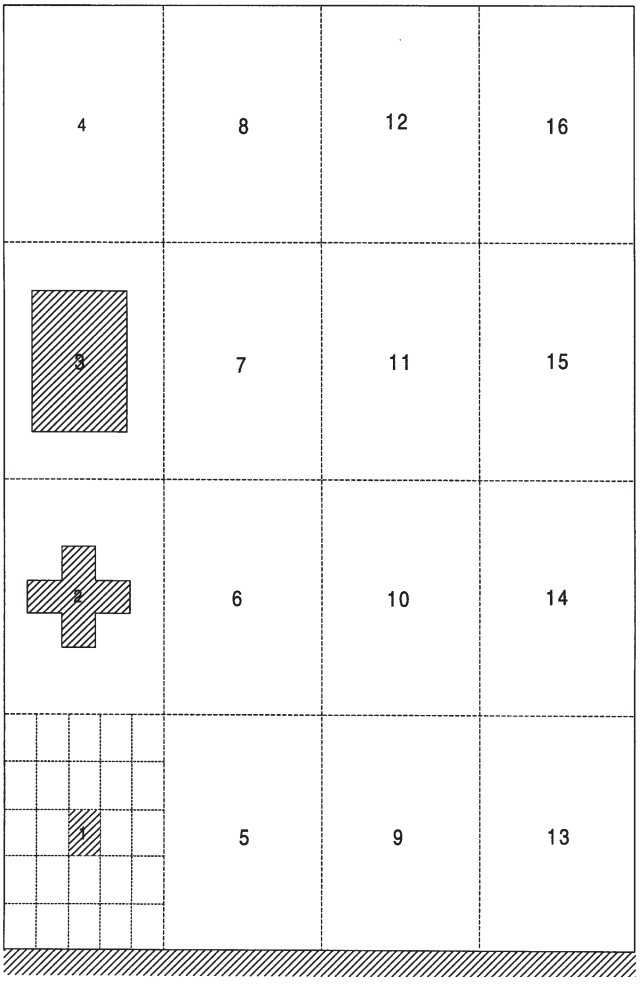

Fig. 1. Numbering system for fault locations and the three fault severity levels indicated by hatched areas.

mation about the safety of the structure, e.g. estimates a residual life.

To simplify matters, this paper is mainly concerned with a level 2 diagnostic, although it will be shown that neural networks can offer level 3 information. A more detailed analysis of neural network fault detection for the structure considered below can be found in [5].

The layout of the paper is as follows: Section 2 gives a brief survey of recent results on the sensor placement problem in structural dynamics. This is followed in Section 3 by an outline of the neural network approach to fault detection. Section 4 describes a heuristic method for sensor placement based on sequential insertion/deletion of sensors. Section 5 describes the use of a genetic algorithm (GA). Finally, Section 6 discusses the use of simulated annealing. A preliminary discussion of some of the results of this paper has appeared in [6].

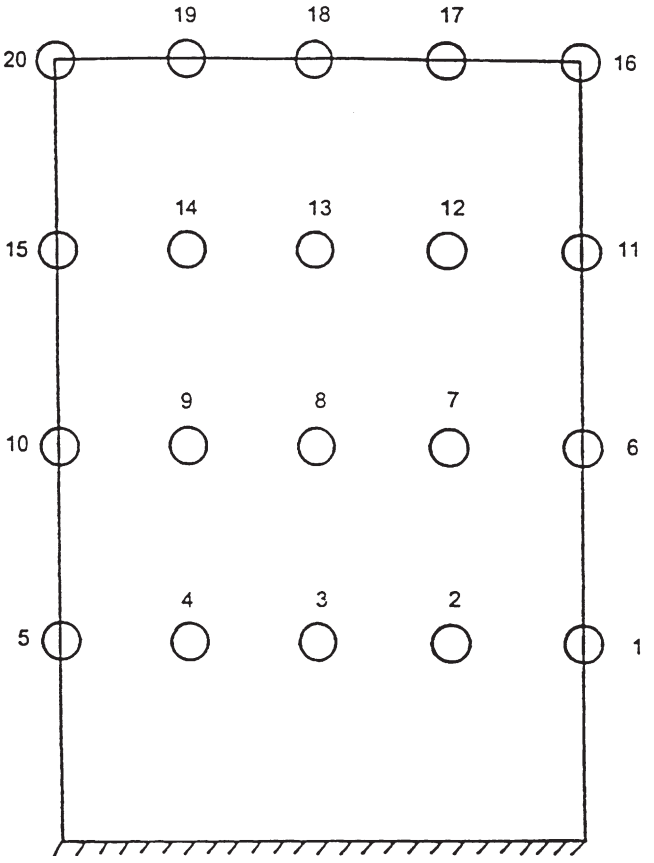

Fig. 2. Numbering system for sensor locations.

2. Sensor optimisation

The problem of determining the optimum number of sensors for a particular application, together with their best possible locations, has received considerable attention recently. The question appears to have first been addressed by control engineers [7], concerned with the identification and control of distributed parameter systems. A large body of work has accumulated and reviews exist [8]. However, much of the recent work on sensor placement has been in the field of structural dynamics, and although little has been specifically concerned with fault detection, the various methods share a common basis. The approaches differ only in their choice of objective function and optimisation algorithm.

Countless methods of optimisation have been developed. The classical approaches based on the availability of gradients and Hessians are surveyed in numerous well-established textbooks, an excellent example is [9].

Recent years have seen the introduction of powerful methods for combinatorial optimisation based on analogies with biology and physics. Artificial neural networks, based loosely on the morphology of the brain, are showing promise. In particular, Hopfield networks can be applied to a large class of optimisation problems

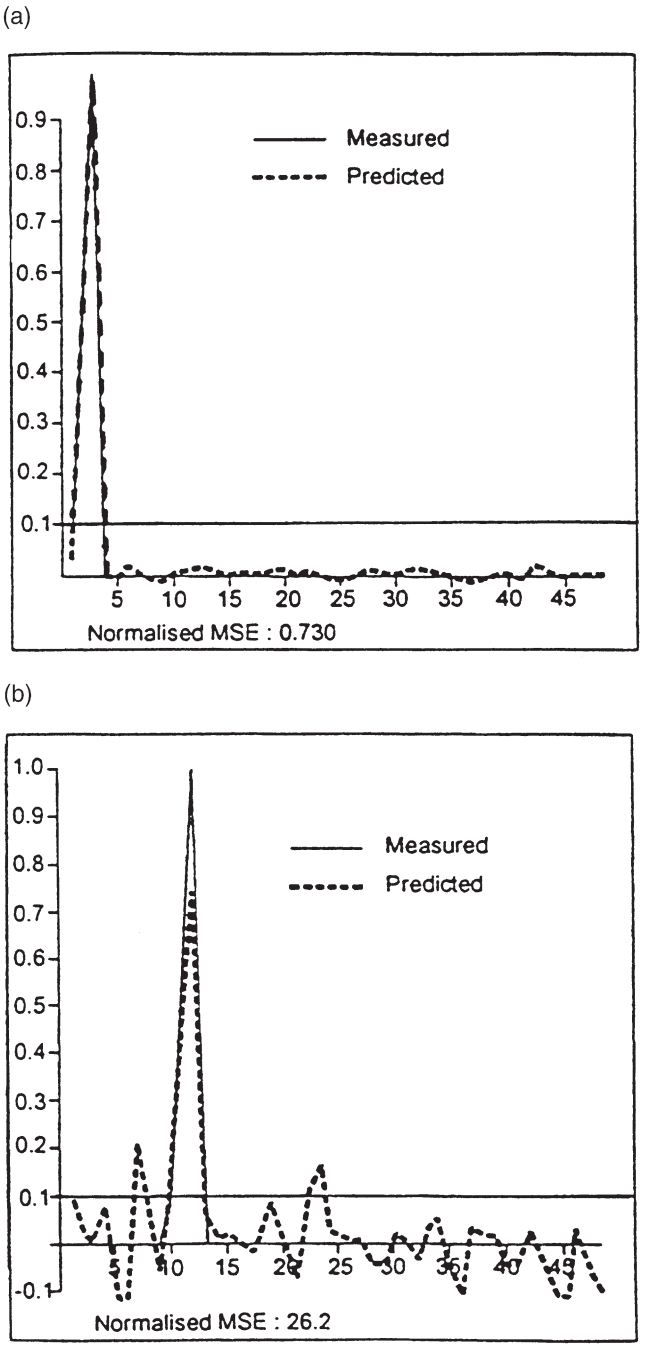

Fig. 3. Comparison between desired outputs and network predictions for location/severity networks trained on the first modeshape: (a) output 1; (b) output 4.

where the objective function can be expressed as a quadratic form [10]. This class includes classic **NP** problems like the travelling salesman problem.

Genetic algorithms (GAs) based on the Darwinian principle of natural selection, are steadily growing in popularity, with many examples of their use appearing in the engineering literature; too many to cite. The pioneering work by Holland [11] has undergone considerable development, and the subject now merits at least one textbook [12]. A discussion of the structure of the GA will be deferred until a later section where it will be applied to the problem in hand.

The method of simulated annealing is an established method of combinatorial optimisation based on an analogy with the physical process of annealing. The method sometimes allows the objective function to increase and thus offers the possibility of escape from local minima. A more detailed discussion of the method will be given later. A comprehensive account can be found in the monograph [13]. Although the method is usually applied to combinatorial problems, extensions have been developed which efficiently minimise functions of continuous variables [14].

A very recent adjunct to these approaches is *tabu search*. This is a 'metaheuristic' which can be applied to any number of optimisation methods. It allows the given method to resist local minima and as such is a sort of deterministic version of simulated annealing. At each stage in the optimisation, a *tabu list* is updated which forbids return to previously visited solutions. Alternatively, a weaker approach based on the implementation of *tabu conditions* which forbid certain transitions can be applied. A brief discussion can be found in [15]; a much more comprehensive account can be found in a survey article by one of the first exponents of the method [16,17].

Each of the methods above shows distinct advantages; attempts have been made to produce 'hybrid' schemes which display the good properties of two or more. Combinations of GAs with simulated annealing show promise [18,19].

Recent solutions to the sensor placement problem have exploited the availability of these methods to the full for a range of objective functions; it is instructive to briefly survey recent work.

In [20], an information-based approach is taken to the sensor placement problem with applications to parameter identification for linear systems. The method essentially formalises the old practical method of placing sensors near the antinodes of the low-frequency vibration modes of the system. The 'goodness' of estimated parameters is conveniently expressed in terms of their covariance matrix. As is well-known, this matrix is bounded below by the Cramer–Rao bound; the better the experiment, the lower the bound. The covariance matrix $[C]$ is the inverse of the Fisher information matrix $[F]$ which depends on the experimental conditions. Minimising the covariance of the estimates thus corresponds to maximising the information in Fisher's sense. The appropriate objective function is therefore some scalar function of the information matrix, typically the trace or the determinant. The approach is applied in [20] to optimise the sensor positions for modal identification of a vibrating beam. No explicit optimisation is carried out; for the 2-sensor problem, the objective is plotted as a surface over

Table 1
Best sensor distributions at each level using first modeshape data and three separate fitness measures

Number of sensors	Sensor distribution	Network	Average error	Maximum error	Failures
20	11111111111111111111	20:18:16	15.10	52.12	0
19	11011111111111111111	19:18:16	15.23	50.68	0
18	11011111111111111011	18:18:16	15.56	50.78	0
17	11011111011111111011	17:16:16	15.25	49.86	0
16	11011101011111111011	16:16:16	15.98	50.02	0
15	01011101011111111011	15:14:16	18.67	52.51	0
14	01010101011111111011	14:14:16	18.70	51.59	0
13	01010101011110111011	13:14:16	18.30	52.88	0
12	01010101011100111011	12:13:16	20.27	54.26	0
11	01010101011100011011	11:13:16	18.91	50.62	0
10	01010101011000011011	10:12:16	20.03	53.11	0
9	01010101010000011011	9:12:16	22.00	58.40	0
8	00010101010000011011	8:12:16	24.02	55.60	0
7	00010101010000011001	7:12:16	26.57	64.23	0
6	00010001010000011001	6:12:16	29.07	63.30	0
5	00000001010000011001	5:12:16	33.18	67.85	0
4	00000001010000001001	4:12:16	38.34	72.10	0
3	00000001010000000001	3:12:16	43.45	78.27	0
2	00000001000000000001	2:12:16	48.70	81.75	2
1	00000000000000000001	1:12:16	75.72	94.60	9

Table 2
Test of 'optimum' 10-sensor pattern against random patterns; feature is first modeshape

Source of pattern	Sensor pattern	Average error	Maximum error	Failures
Algorithm	01010101011000011011	20.03	53.11	0
Symmetry	01010101010101010101	26.74	62.86	1
	10101010101010101010	31.66	72.26	1
Random	01101101000011001110	32.55	77.39	0
	11000100100111001011	29.80	73.85	0
	00010110001111100101	29.32	75.56	1
	01010101001000111101	25.45	67.05	1
	01110111000001101010	32.12	73.20	1
	00010111000110111100	30.12	78.86	1
	11101101110000101000	27.10	79.58	1
	11011101100100000101	29.06	78.79	2
	11011101110000100100	28.84	80.19	3
	11011101110010000001	28.07	78.25	2

the plane specified by the sensor positions along the beam; the maximum is clearly shown. Sensitivity to noise is investigated. The paper contains references to earlier applications of the approach.

The choice of measurement locations for modal testing is of considerable importance in structural dynamics and is the subject of several recent studies. In [21], an a priori FE model is assumed to guide the selection of locations. Two approaches are considered. In the first, used by Leuven Measurement Systems [22], the objective function is the average driving point residue (ADPR). If N_{modes} are measured, this is given by,

$$\text{ADPR}_j = \sum_{i=1}^{N_{\text{modes}}} \frac{\psi_{ij}^2}{\omega_i} \tag{1}$$

where ψ_{ji} is the jth element of the ith eigenvector and ω_i is the ith natural frequency. Given a FE discretisation of the system, the coordinates with the highest ADPR are chosen. These are the coordinates which make the highest (weighted) average contribution to the modeshapes. The second approach in [21] exploits the idea of Guyan model reduction. The assumption is that the coordinates selected by the algorithms as *masters* are the appropriate measurement locations for modal testing. The objective function in this case is, in a sense, related to the ratios m_{ii}/k_{ii} for the ith degree of freedom as the master degrees-of-freedom are chosen where the inertia is high and the stiffness is low. A process of sequential deletion is applied until the model master degrees-of-freedom are reduced to the number of sensors required.

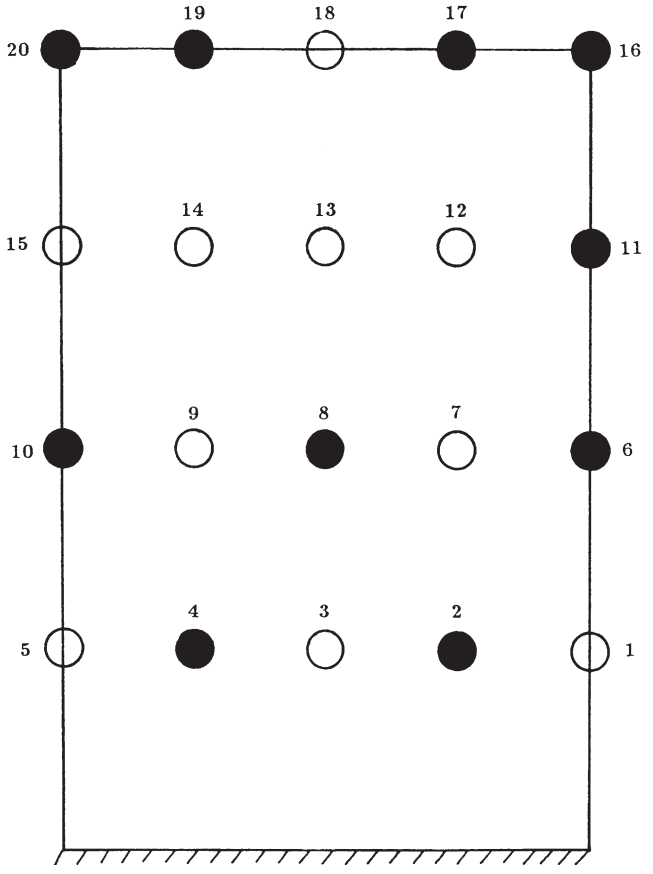

Fig. 4. Best 10-sensor pattern for modeshape diagnostic — algorithm (A).

Illustrating the methods using a simulation of a cantilever beam, the authors conclude that the APDR method works better if there are rigid body modes and the Guyan reduction gives a good set of measurement points (although not necessarily optimum) if the structure is grounded with no rigid body modes. The two methods are compared using two criteria: the first is the *modal assurance criterion* (MAC), defined by,

$$\mathrm{MAC}_{ij}=\frac{|\{\psi_i\}^{\mathrm{T}}\{\psi_j\}|^2}{(\{\psi_i\}^{\mathrm{T}}\{\psi_i\})(\{\psi_j\}^{\mathrm{T}}\{\psi_j\})} \tag{2}$$

which measures the correlation between modeshapes. For an optimal (orthogonal) set this would be diagonal, so the size of the off-diagonal elements is an indication of fitness. The second criterion is the condition number of the modeshape matrix based on a singular value decomposition; this directly measures the extent of linear dependence between modeshape vectors.

In [23] the MAC matrix is used to construct two objective functions directly. The first, Z_1 is simply the sum of off-diagonal elements. The second, Z_2, contains a weighting factor which amplifies MAC values for neighbouring modes as it is assumed that clear identification of close modes is most desirable. The sensor locations are chosen by sequentially deleting coordinates and marking those as measurement points which cause the MAC matrix to generate the largest off-diagonals. The method is demonstrated on a 6872-node model of a laser-cutting centre. The author suggests augmenting the calculated measurement points by 'hand-picked' points to offset errors in the modelling and incorrect specification of boundary conditions.

The method of effective independence (EI) is introduced in [24], based on earlier work in [25,26]. The approach is based on the EI distribution vector E_{D}, defined as the diagonal of the matrix,

$$E=[\psi_{fs}]\{[\psi_{fs}]^{\mathrm{T}}[\psi_{fs}]\}^{-1}[\psi_{fs}]^{\mathrm{T}} \tag{3}$$

where $[\psi_{fs}]$ is a matrix of FE *target modes* partitioned according to a given sensor distribution. Each diagonal element is the fractional contribution of each sensor location to the rank of E, which can only be full rank if the target mode partitions are linearly independent. This means the diagonals represent the fractional contribution of each sensor location to the independence of the target modes. The algorithm is iterative; at each step, terms in E_{D} are sorted to give the least important sensor which is then deleted. The corresponding elements in $[\psi_{fs}]$ are also deleted. The iteration concludes when the required number of sensors is obtained. It can be shown that this procedure tends to maintain the determinant of the Fisher information matrix. The sensitivity of the method to model error was investigated in [27], and an extension allowing for arbitrary measurement noise at the sensors was presented in [28].

A comparison of a number of actuator and sensor placement techniques is given in [29]. The EI method was compared to three others for sensor placement: the kinetic energy method (KE), average kinetic energy (AKE) and eigenvector component product (EVP). The KE method assumes that the sensors will have maximum observability of the modes of interest if the sensors are placed at points of maximum kinetic energy for that mode. The AVE approach places sensors according to KE averaged over all modes of interest; this avoids bias against placing sensors at nodes of particular modes. The EVP method computes the product of the eigenvector components for point i for a range of modes; a maximum for this product is clearly a candidate measurement point (again, some modification is needed if a point is a node of one of the modes). All methods proceed by sequentially deleting the worst candidate points until the correct number of sensors is obtained. On the 96 DOF NASA 8-bay truss (FE simulation), the EI, KE and EVP methods found 15-sensor locations which showed better frequency matching and MAC values than 97% of 300 randomly generated sensor patterns; KE was best. The sensor set obtained from AKE showed no improvement over the random sets. The EI, AKE and EVP methods collocated 14 of the 15 sensors at seven measurement points; the KE method collocated 12 of the 15 sensors

Table 3
Best sensor distributions at each level using first curvature data and three separate fitness measures

Number of sensors	Sensor distribution	Network	Average error	Maximum error	Probability of failure
20	11111111111111111111	20:18:16	0.07	4.07	0.0022
19	11110111111111111111	19:18:16	0.80	4.82	0.0013
18	11100111111111111111	18:18:16	0.80	3.93	0.0013
17	11101101101111111111	17:16:16	0.75	4.04	0.0013
16	11101101111111101101	16:16:16	0.90	4.97	0.0019
15	01101101101111111101	15:14:16	2.66	13.29	0.0017
14	01101101001111111101	14:14:16	2.74	12.57	0.0020
13	01010101011110111011	13:14:16	2.77	12.84	0.0014
12	10100101000111111101	12:13:16	3.60	13.84	0.0025
11	10101101000101111001	11:13:16	3.46	13.73	0.0028
10	10100101000111011001	10:12:16	5.08	15.10	0.0036
9	10101100000111010001	9:12:16	4.73	18.07	0.0039
8	00101101000101010001	8:12:16	5.57	19.90	0.0033
7	10100001000101010001	7:12:16	6.84	22.36	0.0041
6	00101000000101010001	6:12:16	7.37	18.62	0.0059
5	00101000000101000001	5:12:16	10.21	37.17	0.0158
4	00101000000001000001	4:12:16	12.10	41.86	0.0315
3	00100000000001000001	3:12:16	22.49	48.98	0.0952
2	00100000000001000000	2:12:16	43.26	69.82	0.2784
1	00000000000000000001	1:12:16	88.01	99.54	0.8030

Table 4
Best sensor distributions at each level using first curvature data and the delete stategy based on probability of failure

Number of sensors	Sensor distribution	Probability of failure
20	11111111111111111111	0.0022
19	11110111111111111111	0.0013
18	11110101111111111111	0.0009
17	11110101111111111011	0.0013
16	11010101111111111011	0.0017
15	11000101111111111011	0.0014
14	11000101111111111010	0.0017
13	11000001111111111010	0.0017
12	01000001111111111010	0.0016
11	01000001101111111010	0.0019
10	01000001101101111010	0.0023
9	00000001101101111010	0.0027
8	00000001101101111000	0.0050
7	00000001101101101000	0.0056
6	00000000101101101000	0.0086
5	00000000101100101000	0.0126
4	00000000101100100000	0.0182
3	00000000101000100000	0.0577
2	00000000001000100000	0.1711
1	00000000000000100000	0.8297

Table 5
Best sensor distributions at each level using first curvature data and the insert strategy based on probability of failure

Number of sensors	Sensor distribution	Probability of failure
20	11111111111111111111	0.0022
19	11111101111111111111	0.0016
18	11101101111111111111	0.0013
17	11101101111111111011	0.0016
16	11101001111111111011	0.0014
15	01101001111111111011	0.0016
14	01101001111101111011	0.0022
13	01101001011101111011	0.0017
12	01100001011101111011	0.0016
11	01100001011101111010	0.0016
10	01100000011101111010	0.0020
9	01000000011101111010	0.0017
8	01000000011101110010	0.0028
7	01000000011101110000	0.0034
6	01000000010101110000	0.0064
5	01000000010100110000	0.0089
4	00000000010100110000	0.0161
3	00000000010100010000	0.0341
2	00000000010100000000	0.1273
1	00000000010000000000	0.4150

at six points; this is clearly advantageous from a cost standpoint but is not a feature of the methods.

Perhaps the first use of a GA for the sensor placement problem is presented in [30]. The authors propose the GA as an alternative to the EI method which is not guaranteed to produce an optimum solution. The objective function chosen (the *fitness* function in GA terminology) is the determinant of the Fisher information matrix. The GA is applied to determine the best sensor locations from a large candidate set for modal identification of a space structure and a photo-voltaic (PV) array. In the first case, two variants of the GA are tried; one converges monotonically to the maximum, while the other oscillates. The more efficient of the algorithms

Table 6
Test of 'optimum' 10 sensor patterns against random patterns; feature is first curvature. (A) Mixed fitness measure; (B) single fitness-delete sensor; (C) single fitness-insert sensor

Source of pattern	Sensor pattern	Probability of failure
Algorithm		
(A)	01010101011000011011	0.0036
(B)	01000001101101111010	0.0023
(C)	01100000011101111010	0.0020
Symmetry	01010101010101010101	0.0032
	10101010101010101010	0.0063
Random	01101101000011001110	0.0184
	11000100100111001011	0.0053
	00010110001111100101	0.0034
	01010101001000111101	0.0213
	01110111000001101010	0.0184
	00010111000110111100	0.0073
	11101101110000101000	0.0505
	11011101100100000101	0.0156
	11011101110000100100	0.0642
	11011101110010000001	0.0686

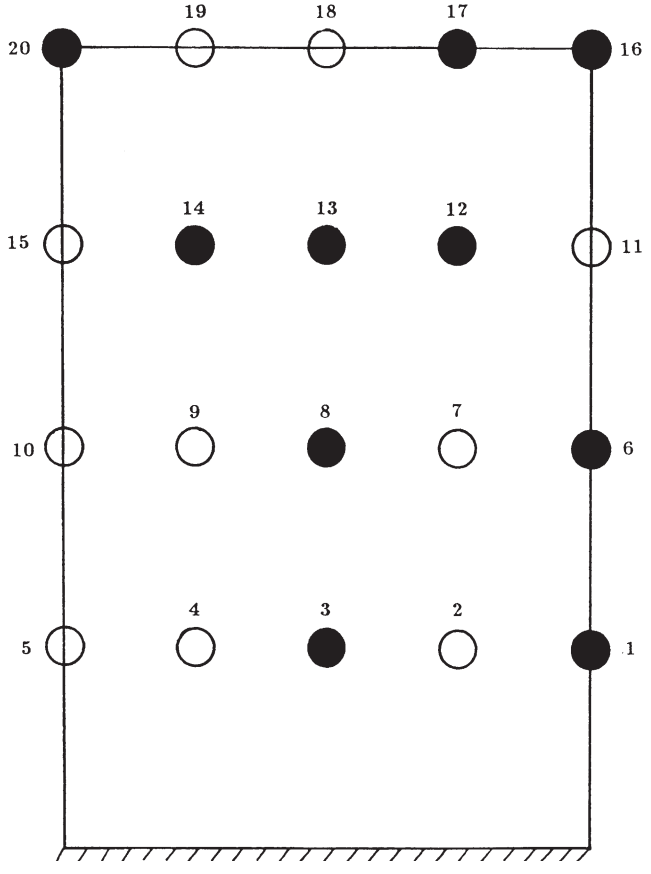

Fig. 5. Best 10-sensor pattern for curvature diagnostic — algorithm (A).

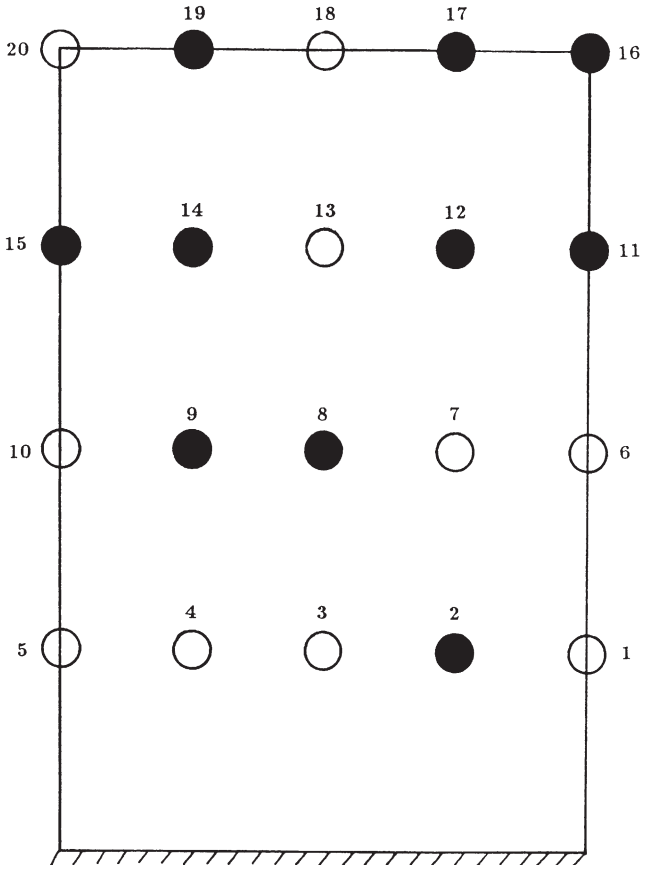

Fig. 6. Best 10-sensor pattern for curvature diagnostic — algorithm (B).

slightly outperforms the EI algorithm on both a 10-sensor and 20-sensor problem. (In fact, the two distributions differ in only a single location in each case.) However, the GA requires up to 30 times the computer time. On the PV array, essentially the same result is obtained. In practice, the true maximum is unknown and the algorithm will not know if it has been achieved. Also, the GA may converge to a local minimum. As a result, the status of the GA solution is no more certain than the EI solution.

In [31], the sensor placement problem is also solved using a GA. The objective function this time is taken as a product of two terms. The first term measures the fitness from the point of view of observability. Three variants are suggested for this component: the first is the smallest singular value of the system observability matrix which is related to the modal amplitude at the measurement points. The second index is based again on the MAC matrix, and the third is based on the determinant of the Fisher matrix. The second component of the fitness function is geometric and penalises the clustering of sensors. Results are given for a simulated cantilever plate which show agreement with physical intuition. It is also shown that the method is fairly insensitive to the observability index chosen.

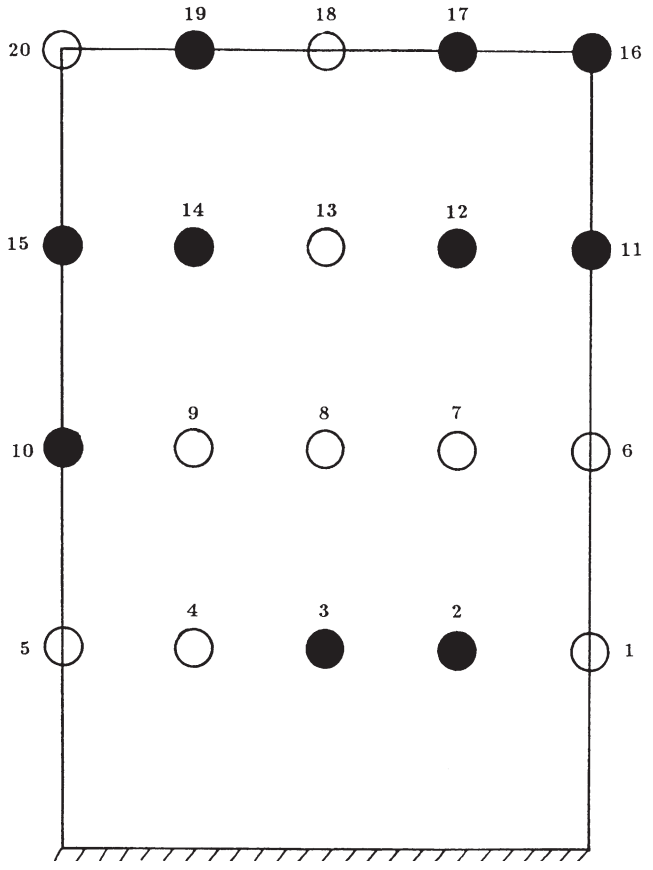

Fig. 7. Best 10-sensor pattern for curvature diagnostic — algorithm (C).

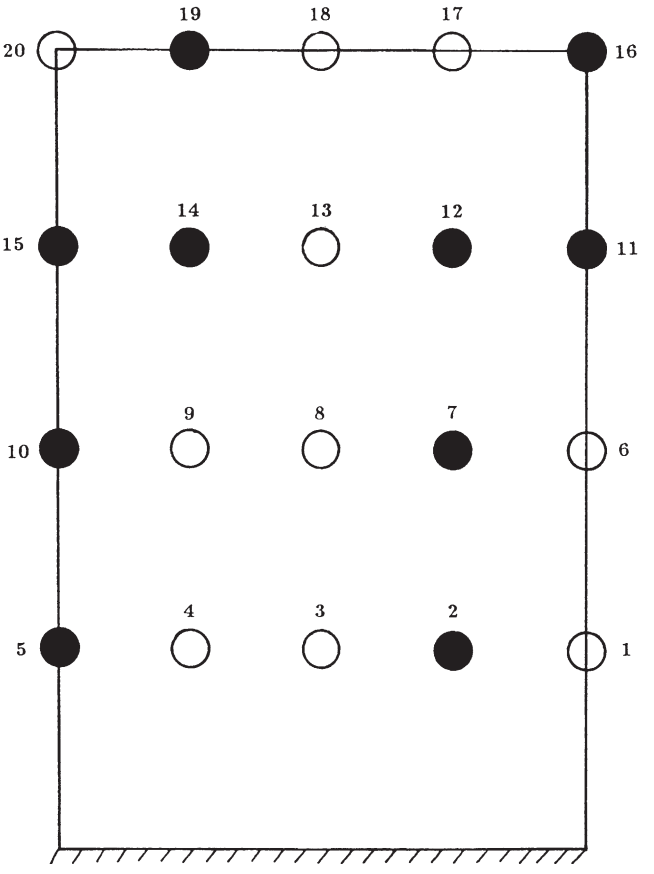

Fig. 8. Best 10-sensor pattern for curvature diagnostic — GA.

Table 7
Probabilities of error from GA sensor distributions

Number of sensors	Probability of error
9	0.0017
10	0.0017
11	0.0011
12	0.0016

In [32], two fitness functions for the GA are assessed. The first is obtained by taking the FE target modeshapes, partitioned according to the candidate sensor set, and evaluating the cross-orthogonality matrix with respect to the Guyan reduced mass matrix. The second is simply the MAC. The latter is faster to compute but less accurate as the FE modes are only truly orthogonal with respect to the mass matrix. For each measure, an error function is computed by taking the absolute value of the difference with an 'ideal' matrix: the identity in the case of the cross-orthogonality matrix, and the MAC from the full modeshapes in the case of the MAC. The GA is compared with three methods: EI, KE and the iterative Guyan reduction method of [21]. The test example is a model of a microcomputer. The GA is tested from a random starting population and one seeded with solutions from the traditional methods. The seeded GA and Guyan method perform best; Guyan produces accurate frequencies for four out of six modes, but the seeded GA provides a more uniform accuracy of estimation. The EI and KE methods give very poor answers.

A different but related application of a GA is presented in [33]. The object of the study is to compute transformations from strain to displacement so that strain frequency response functions (FRFs) can be converted to displacement FRFs. A sensor distribution is required which optimises the transformation. Applications are anticipated in smart structures with embedded fibre-optic sensors. The fitness function is an error between representative measured displacements and those computed from the transformed strain mode. The fitness is weighted by the number of sensors raised to some power in order to trade-off accuracy against number of sensors. The procedure is demonstrated on a FE cantilever plate; 10 strain and displacement modes are extracted and the resulting transformation achieves moderate accuracy with 13 out of 50 sensor locations populated.

A sensor placement study specifically for damage detection can be found in [34]. The EI algorithm is modified to allow the placement of sensors according to strain

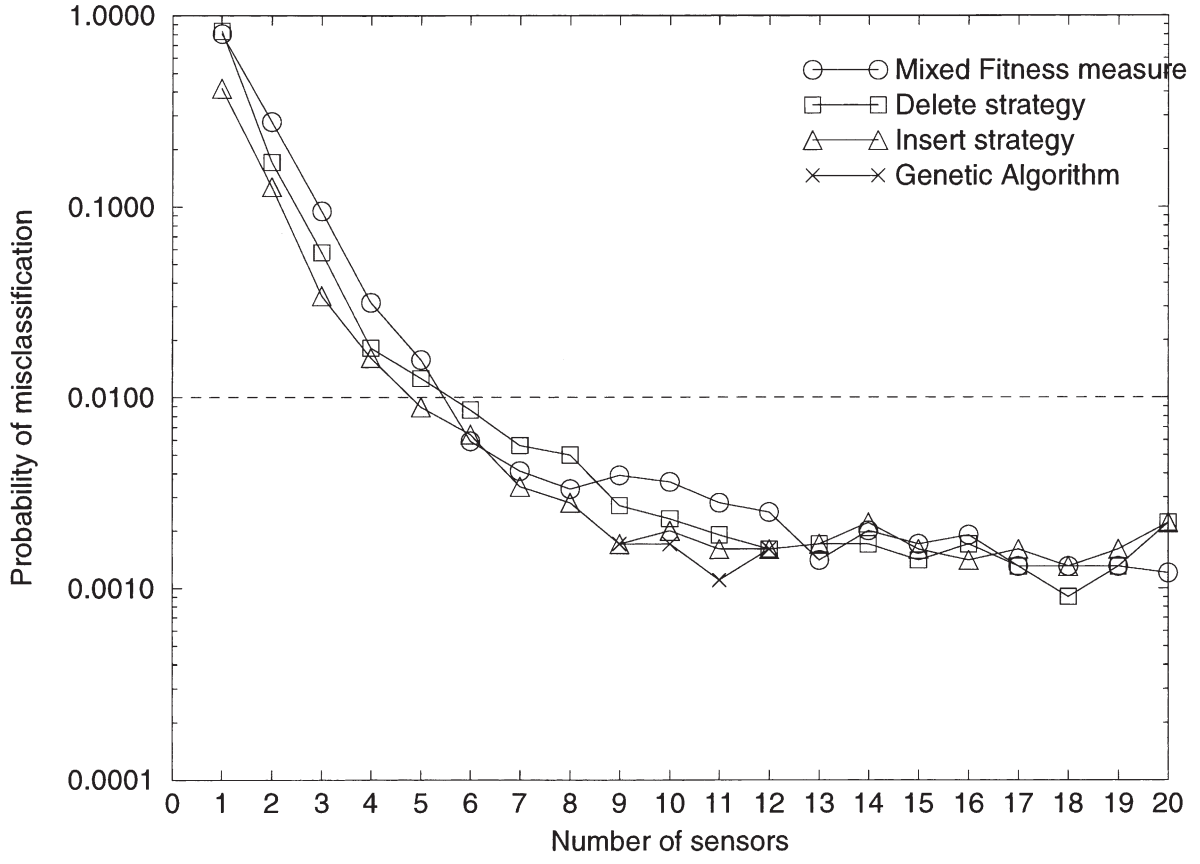

Fig. 9. Probability of error as a function of sensor number.

energy distributions. Rather than maximise the Fisher matrix, the algorithm places sensors along critical load paths in the structure. As for EI, a matrix is constructed with diagonal elements representing the fractional contribution to the strain energy from each sensor. The algorithm iteratively deletes those sensors which contribute least. EI and the new algorithm are used to compute sensor positions for an 8-bay truss structure and the resulting configurations are used to identify structural damage using FE updating. Both methods correctly locate the damaged element but the update based on EI also indicates damage in other areas. The study indicates that damage detection procedures may be sensitive to the sensor distribution.

3. Fault detection in a cantilever plate

The first task of the current study was to produce a benchmark structure for the fault diagnosis and establish the structure and training regime for the neural network. FE simulation was used to produce the training data as it provides a simple means of varying the location and severity of damage within a structure. The neural network used was a standard multi-layer perceptron (MLP) which was trained using the well-known back-propagation algorithm.

A cantilever plate of dimensions 300×200×2.5 mm was chosen as the basic structural model. A discretisation into 20×20 finite elements was used to generate the training sets for a neural network. Faults of different severity were simulated by 'removing' small groups of elements, i.e. setting their value of Young's modulus to zero. The 20×20 mesh was sub-divided into a 4×4 grid, with each grid bounding a group of 5×5 elements into which one of three fault severities could be introduced by deleting one, five or nine elements, as shown in Fig. 1. These are referred to as fault levels 1, 2 and 3, respectively.

Measurements were assumed to be available at only the nodes of the 4×4 grid as shown in Fig. 2, thus only a sub-set of the total information from the FE model was being used.

The basic input/output topology of the fault diagnostic neural network was fixed by the geometry of the model. Measurements were available at the nodes of the 4×4 grid (apart from the nodes at the fixed edge), therefore the input layer of the network required 20 nodes. The outputs of the network were required to indicate the existence of faults at the centres of the grid squares and

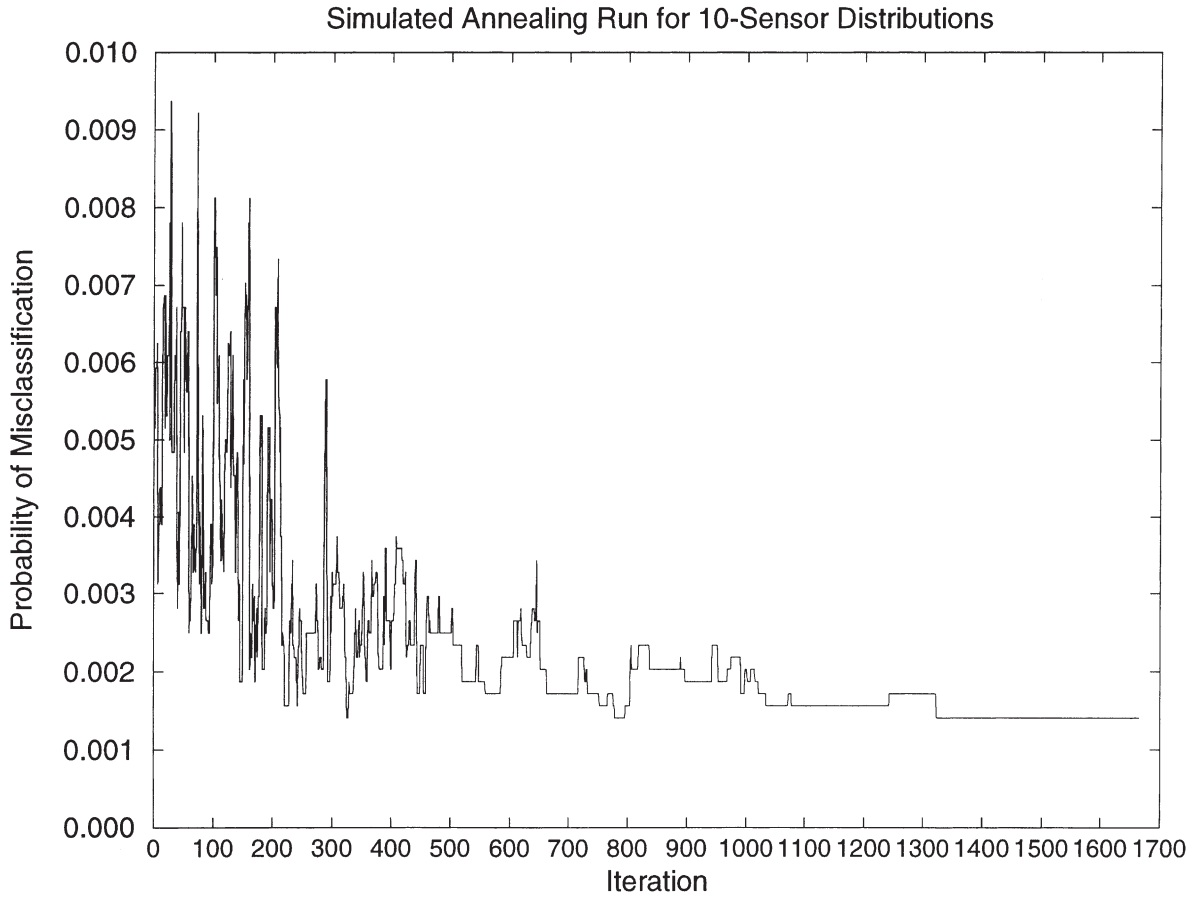

Fig. 10. Simulated annealing run for 10-sensor distributions.

therefore totalled 16. The geometry of the problem did not fix the number and structure of the network hidden layers; this is a matter for trial and error despite the current amount of attention focused on this problem.

The network was trained to respond roughly in proportion to the severity of a fault (which was related to the area of the damage, Fig. 1); the desired outputs were set at 0.1, 0.5 and 1.0 for the three fault levels. A single hidden layer was used giving a final network structure with 20:18:16 nodes.

The measurements used to train the networks were modeshapes and curvatures. The curvatures were obtained by twice differentiating the modeshape data on the original 20×20 mesh using a centred difference. In practice, the curvatures could be sampled directly using strain gauges.

Training the network on the first modeshape data gave very good results. The difficulty for the network was in reporting accurately the lowest level of fault; the desired output of 0.1 being commensurate with the output variances of the network. This was partly due to the problem of small changes occurring in the modeshapes when the damage was located in the elements at the free end of the cantilever plate. Fig. 3 shows these effects in a comparison between the desired and actual network outputs over the whole training set. Fig. 3a shows the comparison for network output 1 which diagnoses problems in an element near the fixed end of the cantilever. The data corresponding to a fault in element 1 is at the beginning of the training set so this is where output 1 is required to be high. The remainder of the data corresponds to faults elsewhere, so output 1 is required to be zero. The noise in the output is low and a correct quantification of the damage is obtained. In contrast, Fig. 3b shows the output corresponding to element 4 near the free end, the level of variation is above the minimum value of 0.1 required to detect fault level 1. It may be possible to minimise this insensitivity by weighting the modal responses for certain elements to give more significance in the training data set; this was not investigated in this work.

Had the network been trained using the 20×20 mesh instead of the 4×4 mesh, improvements in the diagnosis would probably have occurred. However, this highlights the problem of what is possible in practise in terms of the number of measurement points available compared with the data set available from a FE model; the latter can be orders of magnitude higher.

Table 8
Best 10-sensor distributions from simulated annealing

10100001001101111001
10000100101101101011
11100000001111101010

4. Sensor placement by iterative insertion/deletion

As already stated, using an FE model to generate a training data set for a neural network has few restrictions in terms of the number of data points (corresponding to computed displacements, strains etc.). For the plate used in this work, the modeshapes were defined at 20 points (the actual number available was 400), which may be impractical from a modal test point of view. Thus the question arises as to how many sensors are necessary to produce a diagnostic network with a given probability of error and where the optimum positions of the sensors would be. For this study it is assumed that the sensors can only be placed at the 20 locations shown in Fig. 2.

The first approach to this problem was based on a defined *measure of fitness*. An initial state is assumed in which all the sensor locations are occupied. If a single sensor is removed, there are clearly 20 possibilities for 19-sensor patterns. Data from each of these sensor distributions are used to train diagnostic networks and the ability of the trained networks to locate/classify the faults is ranked according to the measure of fitness. The *fittest* is considered to be the best 19-sensor distribution and the corresponding missing sensor is permanently deleted. This leaves a distribution of 19 sensors and the procedure is repeated to find the best 18-sensor distribution etc.

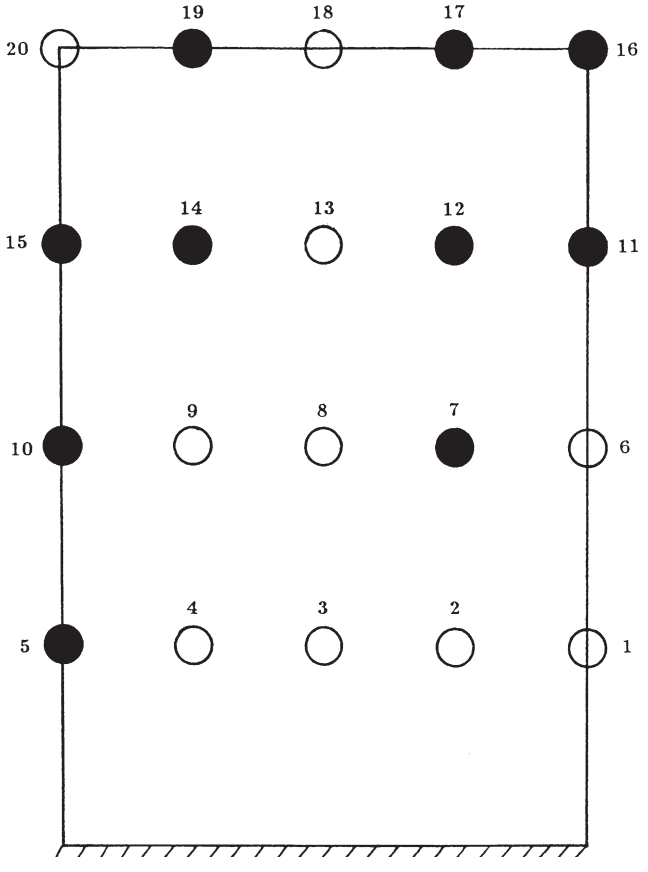

Fig. 11. Best 10-sensor pattern for curvature diagnostic — SA.

This is clearly a time-consuming process and a procedure was used which allowed automatic fitting of many networks. The networks were trained on the full data sets but used an input mask option which instructed them to disregard data from missing sensors. For example, if measurements from 20 sensors are available, and data from only three sensor locations, say 3, 10 and 17, is used the network program is passed the mask 00100000010000001000. This representation is used throughout this work.

In order to establish a measure of fitness, a normalised mean square error (MSE) was defined between the desired network responses y_i and those estimated by the network after training $\hat{y}_i$,

$$\mathrm{MSE}(\hat{y})=\frac{100}{N_T\sigma_i^2}\sum_{j=1}^{N_T}(y_i(j)-\hat{y}_i(j))^2 \tag{4}$$

where i represents the ith output neuron and N_T is the number of training sets indexed by j. σ_i^2 is the variance of the output y_i. The network performance can be judged on the average value of the MSE over the whole set of outputs or the maximum of the set of output MSEs can be used.

The first application of the algorithm described above was to a diagnostic network trained on noisy modeshape patterns for the level 3 faults. (Two hundred copies of each clean pattern were made and Gaussian noise of RMS 0.1% of the pattern magnitude was superimposed.) A mixed measure of fitness was used for the networks. In each case, the average MSE and maximum MSE were obtained; also, the network was trained on the *clean* level 3 patterns and the number of misclassifications was recorded. The fitness was established by ranking the networks according to average error and maximum error and awarding points corresponding to the position of the network in the resulting tables. For example, if 11111011111111111111 had the lowest average error and third lowest maximum error, it would score 4 points. The sensor distribution with the lowest points score with no misclassifications was judged fittest. This is a little convoluted; the justification was that using an approach based on more than one fitness measure might be less subject to local minimum problems. In this case, the number of nodes in the hidden layer was varied as the number of sensors decreased.

The results of applying the algorithm are given in Table 1. It was necessary to test them. In order to do

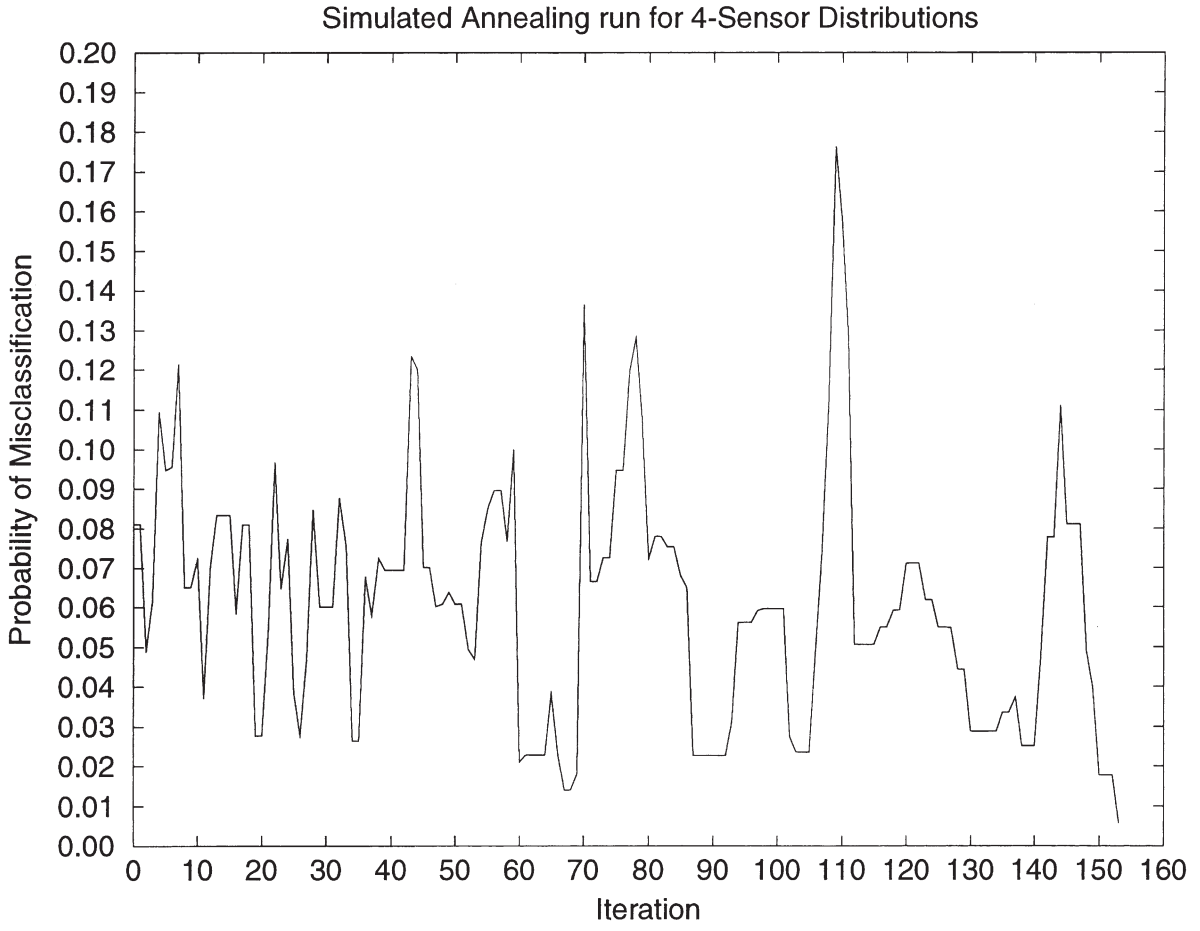

Fig. 12. Simulated annealing run for 4-sensor distributions.

this the 'best' 10-sensor distribution from the algorithm was singled-out and compared to 10 randomly generated distributions. The results are shown in Table 2. Also in the test are two patterns which are suggested by symmetry. Suppose that an infinite plate is considered with a fixed number of sensors per unit area, arranged on a square grid. If the density of sensors is halved, it seems reasonable that the best strategy for removing half of them is to delete alternate sensors. This results in a distribution, also on a square grid, with the distance between sensors magnified by a factor $\sqrt{2}$. On the basis of this argument, the two patterns resulting from deleting alternate sensors on the plate are considered. It is clear that these patterns are not guaranteed to be optimal for a finite plate with boundaries.

Consideration of the results shows that the 10-sensor pattern from the algorithm is superior to all of the randomly generated ones and also those suggested by symmetry. The successful pattern shows the lowest average error and maximum error and has no misclassifications. This gives some confidence in the algorithm. Fig. 4 shows the 'best' 10-sensor pattern from the algorithm.

The procedure was repeated for a curvature-based diagnostic network trained on noisy data corresponding to level 3 faults. (Again, 200 copies of the clean data with 1.0% RMS Gaussian noise added.) The algorithm was essentially applied in the same manner except that the probability of error over the noisy data set was used as a fitness variable rather than the number of misclassifications over the clean data. The networks, and hence patterns, were ranked according to the three measures: average error, maximum error and probability, and the points score was obtained as described above. The resulting n-sensor patterns with the lowest probability of error are given in Table 3.

Note that these patterns are not necessarily those judged fittest by the algorithm as that also takes into account average and maximum MSE. The reason for defining fitness according to three different measures when the object is to minimise one of them in preference to the others is as follows. There is no compelling reason why the best 7-sensor pattern say, should contain *any* of the members of the best 4-sensor pattern. If the algorithm is based on the probability of error only, any pattern is forced to contain all patterns with a lower number of sensors — the n-sensor and $(n-1)$-sensor patterns differ

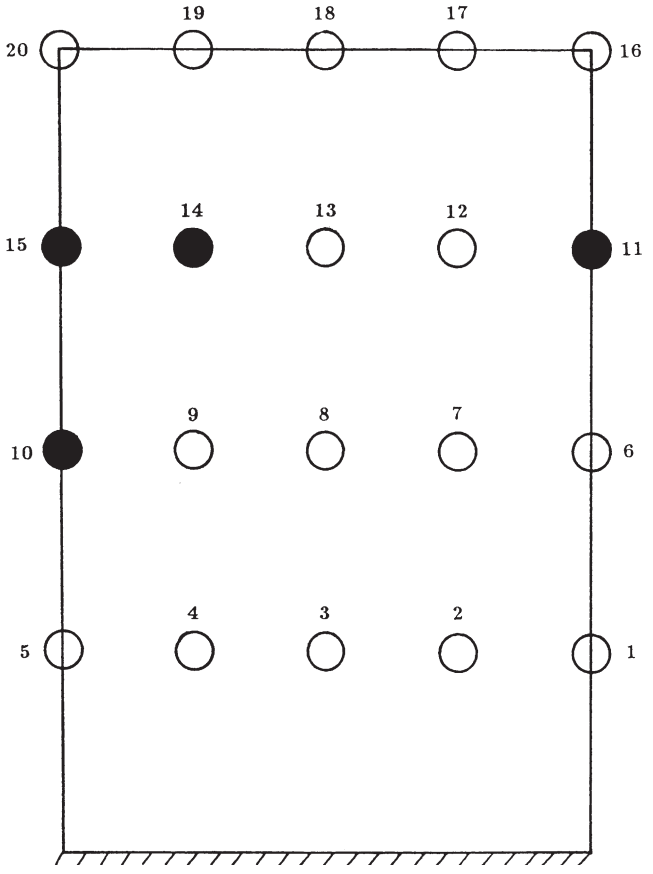

Fig. 13. Best 4-sensor pattern for curvature diagnostic — SA.

in a single location. The approach followed above allows them to differ in *two* locations and it was initially thought that this difference might be significant.

Having said this, the results should, of course, be compared to those from the algorithm based on the probability of error *only* and this approach was taken next. In this case, in each population of *n*-sensor patterns, that with the lowest probability of error is judged the fittest. The resulting patterns are given in Table 4.

Before testing the results, it is important to note that there is an obvious variation of the algorithm which should be tried. The difference is simply that the iteration starts with a plate devoid of sensors and generates diagnostics for all 1-sensor patterns. The pattern with the lowest probability of error is carried on to the next stage when another sensor is added. This approach is clearly faster; at the first step, 20 single-input neural networks are trained compared with 20 19-input networks in the first approach. Also, if the object of the exercise is to obtain patterns with a small number of sensors, the probability of having the true optimum surely decreases with each iteration and it is therefore advantageous to start with low numbers. This method was applied, again using the probability of error as sole fitness measure, and the results are given in Table 5.

The three applications of the algorithm yielded three 10-sensor patterns which could be tested against random patterns as in the last section. The results are given in Table 6. The pattern from algorithm version (A), based on the mixed fitness measure is worst; it is actually inferior to one of the random patterns and one of those suggested by symmetry. The patterns produced by algorithm versions (B) and (C), i.e. the delete and insert versions, respectively, are better than all of the test patterns. The pattern from (C) is slightly superior. The three patterns from the algorithm are given in Figs. 5–7. It is reassuring to note that the patterns from variants (B) and (C) of the algorithm show a marked degree of similarity, despite the fact that the implementations are significantly different.

It will have been noted that, in the results above, the probability of misclassification sometimes increases with the sensor number. This is a result of the network training. Each network is initialised with randomly chosen connection weights, so for each sensor number, the probability is only a sample from a distribution of possible results. The results show that the variance of these distributions is small.

5. Sensor placement using a genetic algorithm

For the sake of completeness, a brief discussion of genetic algorithms (GAs) will be given here; for more detail the reader is referred to the standard introduction to the subject [12]. The training data used for this section were the level 3 curvature data.

Genetic algorithms are optimisation algorithms which evolve solutions in a manner analogous to the Darwinian principle of natural selection. They differ from more conventional optimisation techniques in that they work on encoded forms of the possible solutions. Each possible solution, i.e. each set of possible parameters in solution space, is encoded as a gene. The most usual form for this gene is a binary string, e.g. 00011010110. The first hurdle in setting up a problem for solution by genetic algorithm methods is working out how best to encode the possible solutions as genes. In the case of the sensor location problem, a natural coding is provided by the input masks described earlier, e.g. the gene 00100010000000001000 represents a solution in which transducers are placed at positions 3, 7 and 17.

Having decided on a representation, the next step is to generate, at random, an initial population of possible solutions. The number of genes in a population depends on several factors, including the size of each individual gene, which itself depends on the size of the solution space.

Having generated a population of random genes, it is necessary to decide which of them are fittest in the sense of producing the best solutions to the problem. To do this, a fitness function is required which operates on the

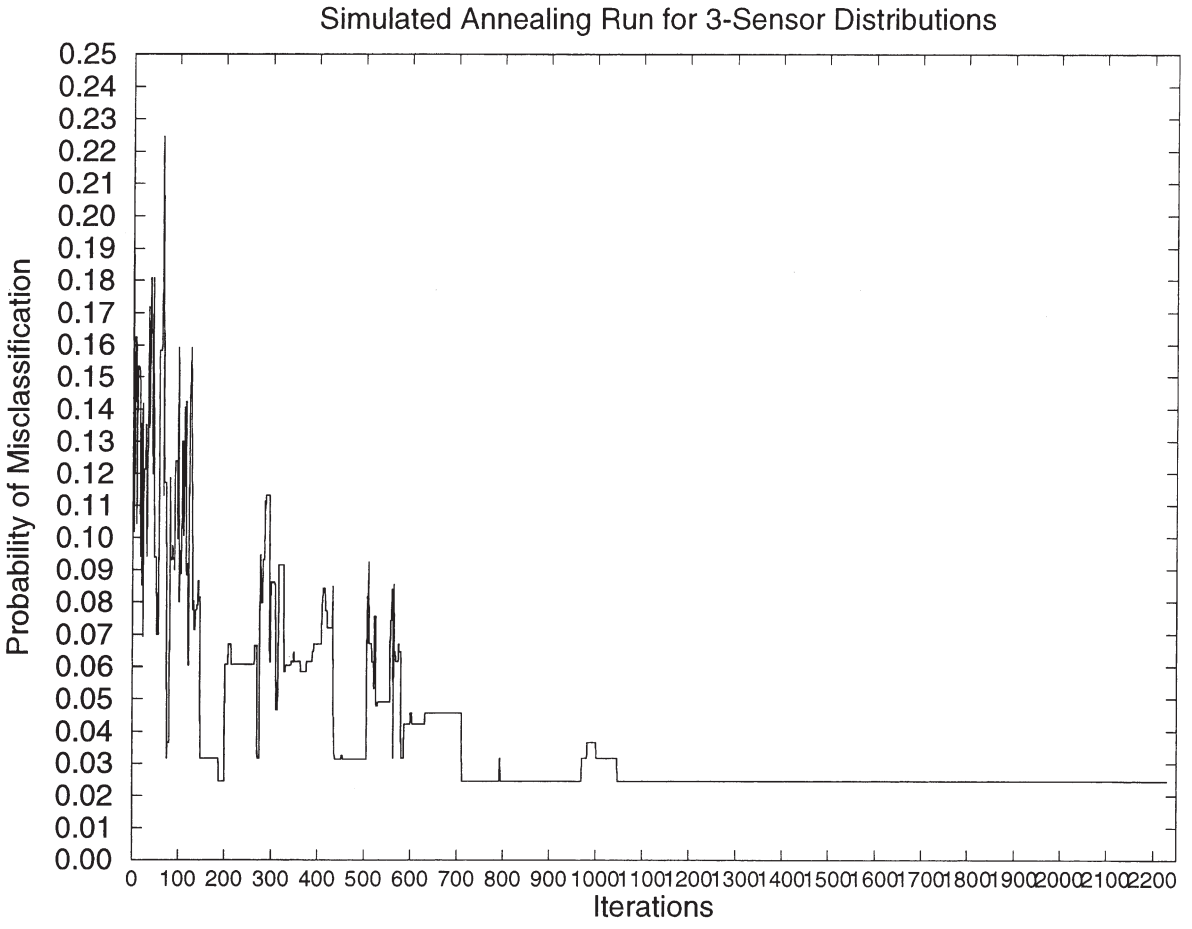

Fig. 14. Simulated annealing run for 3-sensor distributions.

encoded genes and returns a single number which provides a measure of the suitability of the solution. These fitter genes will be used for mating to create the next generation of genes which will hopefully provide better solutions to the problem. Genes are picked for mating based on their fitness. The probability of a particular gene being chosen is equal to its fitness divided by the sum of the fitnesses of all the genes in the population. Once sufficient genes have been selected for mating, they are paired up at random and their genes combined to produce two new genes. The most common method of combination used is called *crossover*. Here, a position along the genes is chosen at random and the substrings from each gene after the chosen point are switched. This is 1 point crossover. In 2 point crossover a second position is chosen and the gene substrings switched again. There is a natural fitness measure for the sensor location problem, namely the inverse of the probability of misclassification. This is modified here by the addition of a penalty function which suppresses solutions which do not have the desired number of sensors. A simple quadratic penalty function was used.

If a gene in a particular generation is extremely fit, i.e. is very close to the required solution, it is almost certain to be selected several times for mating. Each of these matings, however, involves combining the gene with a less fit gene so the maximum fitness of the population may be lower in the next generation. To avoid this, a number of the most fit genes can be carried through unchanged to the next generation. These very fit genes are called the *elite*.

To prevent a population from stagnating, it can be useful to introduce perturbations into the population. New entirely random genes may be added at each generation. Such genes are referred to as *new blood*. Also, by analogy with the biological process of the same name, genes may be *mutated* by randomly switching one of their binary digits with a small probability.

With genetic methods it is not always possible to say what the fitness of a perfect gene will be. Thus the iterative process is usually continued until the population is dominated by a few relatively fit genes. One or more of these genes will generally be acceptable as solutions.

Note that the gene encoding used here is only suitable for numbers of sensors close to 10. This is because the number of distinct n-sensor genes is $\binom{20}{n}$ and this is very strongly peaked at 10. This means that genes intro-

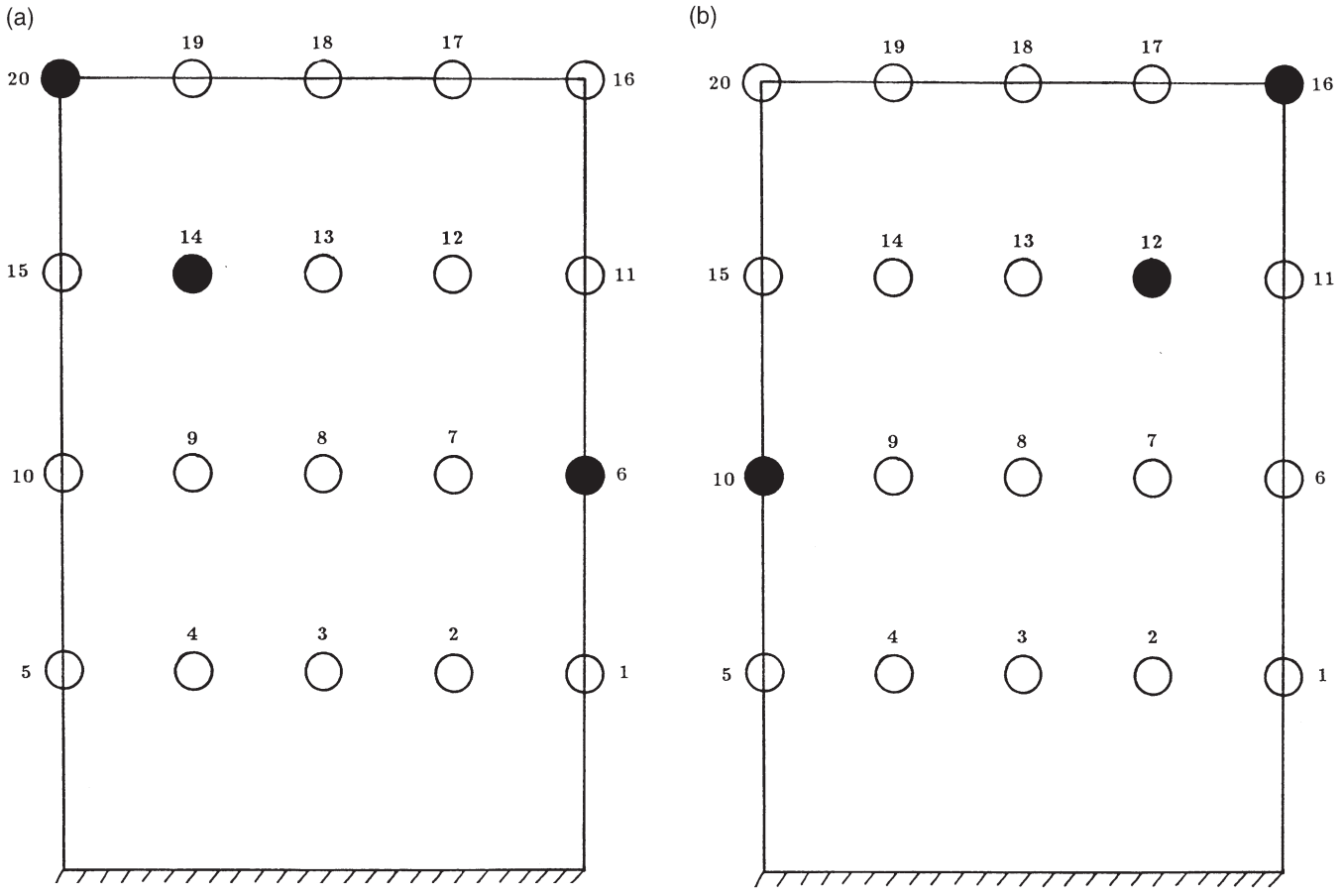

Fig. 15. Best 3-sensor patterns for curvature diagnostic: (a) SA; (b) exhaustive search.

duced into the population by the various operations will have highest probability of having close to 10 sensors. For this reason the GA was only used to generate distributions with 9–12 sensors. Fifty generations were used in each case, the GA used a single member elite and introduced five new blood at each step. The network structure was given a constant number (16) of hidden units. The resulting probabilities of misclassification for the fittest gene in each case are given in Table 7. Fig. 8 shows the 10-sensor pattern from the GA (actually, the mirror image); it is again encouraging that it places eight from 10 sensors in common with the insert strategy (C).

All the results for 10-sensor distributions based on the curvature diagnostic are summarised in Fig. 9 which gives the probability of error as a function of sensor number for all three variants of the insertion/deletion algorithm. It is clear that the method based on a mixed fitness measure has no advantages over the simpler approach based on a single fitness variable. As might be expected, variant (C) gives significantly better patterns than (B) with small numbers of sensors, while variant (B) is slightly superior for patterns with high numbers. The GA outperforms all methods over the limited range available; however, it should be recalled that the network structure used is a little more complex. Fig. 10 is one solution to the problem of determining how many sensors are needed to guarantee a certain probability of success. Note that a 5-sensor distribution is sufficient to give a greater than 99% chance of successfully locating the faults.

This is very much a preliminary study. Work continues using more complex gene encodings which do not bias the search towards the 10-sensor distributions.

6. Sensor placement using simulated annealing

The simulated annealing (SA) algorithm is based on an analogy with the physical process of annealing. Normal polycrystalline solids have a microscopic domain structure which is quite disordered; under equilibrium conditions, the solid is only at a *local* minimum of energy. However, if the solid is heated to a high temperature and then cooled slowly, it will arrive at a highly ordered state — the *global* minimum of energy. Consider a random search method for optimization: this attempts moves in random directions in the search space, accepting a trial move only if the objective function decreases. One can think of the method as moving downwards gradually on the surface specified by the objective.

Because the method accepts all downward moves it will converge to a local minimum if the starting point for the algorithm is within the basin of attraction of that minimum. In order to escape from local minima, one can modify the algorithm by adding a certain amount of *thermal motion* which allows the objective to sometimes increase. This is the basis of SA: the allowed upward movements are large in the early stages of the algorithm when local minima are to be avoided, corresponding to a high temperature regime. In the later stages, the thermal motion is small to allow convergence, corresponding to low temperature. To decide if an upward movement is allowed, Boltzman statistics are used. A probability,

$$p=\exp\left(\frac{-\Delta E}{T}\right) \tag{5}$$

is computed, and if less than a uniform random deviate on the interval [0,1], the move is accepted. Here, ΔE is the change in the objective function (positive), and T is an external control parameter, the *temperature* which decreases as the algorithm proceeds. Downward movements are always accepted. The origins of the method are in the Metropolis algorithm [35], although the serious development of the method is associated with Kirkpatrick et al. [36].

The implementation of the algorithm used here follows that in [37]. The objective function is the probability of misclassification. The initial temperature is chosen to be of the same order of magnitude as the probability of error for a randomly chosen sensor distribution. Moves are made by operating on the bit vectors described in the previous sections. If a 10-sensor distribution is desired, an initial distribution is chosen randomly with 10 bits set. Trial moves involve changing the position of a 1 randomly. Fifty trial moves are taken at each temperature; unless 10 moves are accepted, in each case, the temperature is decreased by a multiplicative factor of 0.9. Fifty temperature steps are taken. It is clear that the method is applicable to a general n-sensor distribution, unlike the GA which, because of the coding used, only works for n around 10. A curvature diagnostic with a neural network structure with 16 hidden units was assumed.

The first run of the algorithm was directed to find a 10-sensor distribution. Fig. 10 shows that the algorithm converged after 1350 iterations to a solution with a probability of error of 0.0014, an improvement on the GA and the best result so far. In fact, during the run, the algorithm found three solutions with this error; they are given in Table 8. The second of these distributions is depicted in Fig. 11; it shows eight sensors in agreement with the GA solution and eight sensors in common with the algorithm (C) solution.

The results so far have shown that five sensors are needed if the diagnostic is to be 99% accurate. The next run of the SA algorithm attempted to find a 4-sensor distribution with this property. It converged successfully after 150 iterations (Fig. 12) to a solution with a probability of error of 0.0059. The distribution is given in Fig. 13. It is interesting to note that this is a sub-distribution of the 10-sensor solution.

The final run of the algorithm attempted to find a 99% accurate 3-sensor distribution. This proved beyond the power of the method, which converged after 1000 iterations to a solution with probability of error 0.0245 (Fig. 14). The distribution is shown in Fig. 15a. Because of the low number of sensors, in this case it was possible to carry out an exhaustive search for the true minimum. This was discovered to be 0.0242 and the corresponding distribution (Fig. 3b) was simply the mirror image of that found by the SA. The small discrepancy can be attributed to the different neural network training presentations. The best distribution is not a subset of the best 4-sensor distribution found.

7. Conclusions

Several methods have been presented for determining sensor distributions for fault diagnosis. On curvature data, which appears to be the best feature for training neural network diagnostics, three of the methods [methods (C), SA and GA] show impressive agreement as to the best 10-sensor distribution. The SA algorithm finds a 4-sensor distribution for the plate which has a 99.5% probability of correct classification. In the case of the 3-sensor distribution where the true minimum was known, the SA algorithm actually found the best distribution. On this showing, the combinatorial methods SA and GA appear to offer the most promise, with the SA algorithm proving slightly superior; although it must be said, that the implementation of the GA was not optimal. Comparisons of the computational effort are difficult as the methods converged to solutions over a wide range of iterations.

Acknowledgements

This work has been partially supported by Aircraft Systems, The Defence Research Agency. The authors would also like to thank Professor Geof Tomlinson and Dr Wieslaw Staszewski of the University of Sheffield for valuable discussions, and to thank Dr Dan Shevitz of Los Alamos National Laboratory for a couple of important observations.

References

[1] Kudva J, Munir N, Tan PW. Damage detection in smart structures using neural networks. Smart Mater Struct 1992;1:100–12.

[2] Worden K, Ball AD, Tomlinson GR. Fault location in a framework structure using neural networks. Smart Mater Struct 1993;2:189–200.
[3] Farrar CR, Prime MH, Doebling SW, Shevitz DW. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. Los Alamos National Laboratories report LA-13070-MS, 1996.
[4] Rytter A. Vibration based inspection of civil engineering structures, Ph.D. thesis. Denmark: Department of Building Technology and Structural Engineering, University of Aalborg, 1993.
[5] Worden K, Tomlinson GR. Damage location and quantification using neural networks. In: Edwards JH, Kerr J, Stanley P, editors. Engineering integrity assessment., 1994:11–31.
[6] Worden K, Burrows AP, Tomlinson GR. A combined neural and genetic approach to sensor placement. In: Proceedings of the 15th International Modal Analysis Conference, Nashville, Tennessee, 1995.
[7] Yu TK, Seinfeld JH. Observability and optimal measurement locations in linear distributed parameter systems. Int J Control 1973;18:785–99.
[8] Kubrusly CS, Malebranche H. Sensors and controllers location in distibuted systems — a survey. Automatica 1985;21:117–28.
[9] Fletcher R. Practical methods of optimization. 2nd ed. Chichester: John Wiley & Sons, 1987.
[10] Hopfield JJ, Tank DW. Neural computation of decisions in optimization problems. Biol Cybernetics 1985;52:141–52.
[11] Holland JH. Adaption in natural and artificial systems. Ann Arbor: University of Michigan Press, 1975.
[12] Goldberg DE. Genetic algorithms in search, machine learning and optimisation. New York: Addison Wesley, 1989.
[13] van Laarhoven PJM, Aarts EHL. Simulated annealing: theory and applications. Reidel Publishing Co, 1987.
[14] Corana A, Marchesi M, Martini C, Ridella S. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. ACM Trans Math Software 1987;13:262–80.
[15] De Werra D, Hertz A. Tabu search techniques: a tutorial and application to neural networks. OR Spectrum 1989;11:131–41.
[16] Glover F. Tabu search: part I. ORSA J Computing 1989;1:190–206.
[17] Glover F. Tabu search: part II. ORSA J Computing 1990;2:4–32.
[18] Mahfoud SW, Goldberg DE. Parallel recombinative simulated annealing: a genetic algorithm. Parallel Computing 1995;21:1–28.
[19] Yip PCP, Pao YH. Combinatorial optimization with use of guided evolutionary simulated annealing. IEEE Trans Neural Networks 1995;6:290–5.
[20] Kirkegaard PH, Brincker R. On the optimal location of sensors for parametric identification of linear structural systems. Mech Systems Signal Processing 1994;8:639–47.
[21] Penny JET, Friswell ML, Garvey SD. Automatic choice of measurement locations for dynamic testing. AIAA Journal 1994;32:407–14.
[22] LMS International. Large-scale modal tesing of a space frame structure — from pretest analysis to FEA model validation. Sound Vibr. March 1991:6–16.
[23] Breitfeld T. A method for identification of a set of optimal points for experimental modal analysis. In: Proceedings of the 13th International Modal Analysis Conference, Nashville, Tennessee, 1995.
[24] Kammer DC. Sensor placement for on-orbit modal identification and correlation of large space structures. J Guidance, Control Dynam 1991;14:251–9.
[25] Shah PC, Udwadia FE. A methodology for optimal sensor locations for identification of dynamic systems. J Appl Mech 1978;45:188–96.
[26] Udwadia FE, Garba JA. Optimal sensor locations for structural identification. In: JPL Proceedings of the Workshop on Identification and Control of Flexible Space Structures, 1978:247–61.
[27] Kammer DC. Effect of model error on sensor placement for on-orbit modal identification. J Guidance, Control Dynam 1992;15:334–41.
[28] Kammer DC. Effects of noise on sensor placement for on-orbit modal identification of large space structures. Trans ASME 1992;114:436–43.
[29] Larson CB, Zimmerman DC, Marek EL. A comparison of modal test planning techniques: excitation and sensor placement using the NASA 8-bay truss. Department of Mechanical Engineering, University of Houston, preprint, 1994.
[30] Yao L, Sethares WA, Kammer DC. Sensor placement for on-orbit modal identification via a genetic algorithm. AIAA Journal 1993;31:1167–9.
[31] Frauchi CG, Gallieni D. Pre-test optimisation by genetic algorithm. In: Proceedings of the 19th International Seminar on Modal Analysis, Leuven, Belgium, 1994.
[32] Stabb M, Blelloch P. A genetic algorithm for optimally selecting accelerometer locations. In: Proceedings of the 13th International Modal Analysis Conference, Nashville, Tennessee, 1995.
[33] Foss GC, Haugse ED. Using modal test results to develop strain to displacement transformations. In: Proceedings of the 13th International Model Analysis Conference, Nashville, Tennessee, 1995.
[34] Hemez FM, Farhat C. An energy based optimum sensor placement criterion and its application to structural damage detection. Center for Aerospace Structures, Department of Aerosapce Engineering, University of Colorado at Boulder, preprint, 1994.
[35] Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E. J Chem Phys 1953;21:1087–92.
[36] Kirkpatrick S, Gelatt CD, Vecchi MP. Science 1983;220:671–80.
[37] Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical recipes — the art of scientific computing. Cambridge: Cambridge University Press, 1986.