Optimisation of Vessel Resistance using Genetic Algorithms and Artificial Neural Networks

Andrew Mason, Patrick Couser, Formation Design Systems, Fremantle/Australia,
AndyM@formsys.com

Garth Mason, Cameron R. Smith, Brian R. von Konsky, Curtin University of Technology

Abstract

It has been found that an artificial neural network is able to produce results of sufficient accuracy to be useful for preliminary prediction of vessel resistance, with the major benefits of: being relatively simple to set up; being easily retrained with new data; and that Froude number may be easily included as an independent variable. In this work the Neural Network is fitted directly to the original tank test data, rather than to a set of smoothed curves varying in the Froude number axis. In addition, different network architectures have been investigated, with networks containing two hidden layers being seen to perform well. A procedure for the optimisation of design parameters using a Genetic Algorithm has also been evaluated. The Genetic Algorithm uses the approximation provided by the Neural Network response surfaces for its objective function. This has been found to be effective and of acceptable performance.

Nomenclature

B	Demi-hull maximum beam of waterline
C_F	Coefficient of frictional resistance [ITTC-57 Correlation line]
C_R	Coefficient of residuary resistance
C_T	Coefficient of total resistance
C_W	Coefficient of wave resistance
F_n	Froude number [$\upsilon / \sqrt{(gL)}$]
g	Acceleration due to gravity
L	Vessel length between perpendiculars
$L/\nabla^{1/3}$	Slenderness ratio
R_e	Reynolds number [$\upsilon L/\nu$]
S	Catamaran demihull centerline separation
T	Demi-hull draft
υ	Vessel velocity
WSA	Wetted surface area
∇	Demi-hull volume of displacement
ν	Fluid kinematic viscosity
ρ	Fluid density

1. Introduction

Artificial Neural Networks (ANNs) have been used for classification and prediction across many disciplines including medicine, engineering, and computer science, primarily because of their ability to model non-linear functions quickly and accurately.

In the field of naval architecture, interpolation and prediction of hull resistance from model experiments and tank testing have traditionally been made using statistical regression equations. However, this is a problem that is also well suited to neural networks. Specifically a neural network may be a more favourable option than statistical methods since they have been shown to provide greater flexibility and can more quickly create accurate models of complex systems, *Jain et al. (1996).*

This paper describes an investigation into the accuracy of ANNs as prediction tools for hull resistance and follows on from the work presented in *Couser et al. (2004).* The goal of the original investigation was to determine a predictive model for residuary resistance (C_R), based on input values of Froude number (F_n), Separation-Length ratio (S/L), Breadth-Draught ratio (B/T) and Slenderness ratio ($L/\nabla^{1/3}$). The data used for the investigation originated from a series of tank tests described in *Molland et al. (1994,1995), Insel and Molland (1992).*

In the previous work, the authors had the choice of using the raw tank test data, or using a larger dataset derived from a least squares fit to the F_n axis of the raw data. At that time it was felt that the larger dataset would give superior results, however in retrospect it was felt that the additional fitting step may have introduced errors. As a result it was decided to attempt fitting of the original experimental data directly, despite the data being highly non-linear in the F_n axis, there being a relatively small number of points and the presence of significant noise in the data.

The non-linearities in the solution surface fitted to the resistance data by the neural network raise an additional problem when it comes to using the output of the trained ANN within an optimisation procedure. Because the five-dimensional solution surface is multimodal, it is not feasible to use a gradient-based optimisation method to reliably find the optimum set of parameters.

To address this problem the authors have implemented a simple Genetic Algorithm (GA) based optimisation framework to investigate the application of GAs to catamaran design optimisation. The framework was designed to be as flexible as possible in order to experiment with the GA parameters to determine the best combination of these settings.

2. Neural Networks

According to *Sarle (1994)*

> Neural Networks are general purpose, flexible, non-linear models that, given enough hidden neurons and enough data, can approximate virtually any function to any desired degree of accuracy. In other words, Neural Networks are universal approximators. Neural Networks can be used when there is little knowledge about the form of the relationship between the independent and dependent variables.

Most ANNs have some sort of training rule whereby the weights of connections are adjusted on the basis of training data. In other words, ANNs learn by example and exhibit some capability for generalization beyond the training data. There are many different types and topologies of ANNs. The form that has found widest application is the feed-forward multi-layer perceptron or MLP.

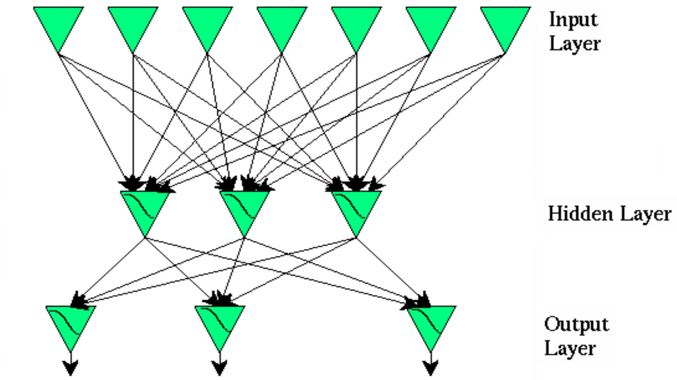


Fig.1: MLP topology

MLPs have an input layer with a series of inputs, one or more hidden layers and an output layer. The number of input elements is equal to the number of variables in the input dataset and the number of outputs equal to the number of result values required. The number of hidden layers and the number of elements in them can vary, and there are several techniques used to determine the optimum structure.

There is considerable overlap between the fields of ANNs and statistics. In ANN terminology, statistical inference means learning to generalize from noisy data. Most ANNs that can learn to generalize effectively from noisy data are analogous to statistical methods. For example:

- MLPs are a subset of the class of non-linear regression models.
- MLPs with no hidden layer are basically generalized linear models.
- MLPs with one hidden layer are closely related to projection pursuit regression.
- MLPs with multiple inputs, multiple outputs and one or more hidden layers are analogous to multivariate, multiple non-linear regression.

Despite the similarities between ANNs and statistical methods, ANNs have some specific advantages-

- Power. ANNs are capable of modelling extremely complex functions. In particular, ANNs are non-linear. For many years linear modelling has been the commonly used technique in most modelling domains, since linear models had well-known optimisation strategies. Where the linear approximation was not valid (which was frequently the case) the models suffered accordingly. ANNs also keep in check the “curse of dimensionality” problem, which bedevils attempts to model non-linear functions with large numbers of variables.

- Ease of use. ANNs learn by example. The ANN user gathers representative data, and then invokes training algorithms which enable the ANN to learn the structure of the data. Although the user requires some knowledge of how to select and prepare the data, the level of user knowledge needed to successfully apply ANNs is much lower than that required for more traditional statistical methods.

2.3. Training and Testing Overview

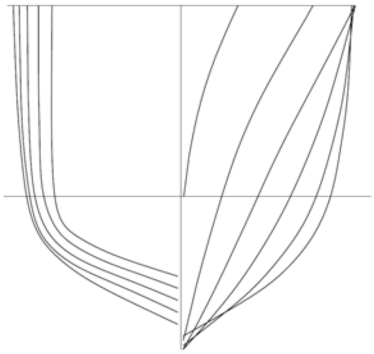
The ANNs discussed in this paper were developed and tested using software called NeuroIntelligence, *Alyuda (2005).* NeuroIntelligence is a commercial ANN program, which specialises in the training of multi-layer perceptrons. NeuroIntelligence contains a variety of training algorithms, including back propagation, conjugate gradient descent, quasi-Newton and Levenberg-Marquardt methods.

2.4. Training and Testing Data

The data set used to train and test the ANN is a series of geometrically similar catamaran models based on the hull shape shown together with the parameters described in Table I.

Table I

Model	B/T	$L/\nabla^{1/3}$	Demihull WSA m^3
3b	2.0	6.27	0.434
4a	1.5	7.40	0.348
4b	2.0	7.41	0.338
4c	2.5	7.39	0.34
5a	1.5	8.51	0.282
5b	2.0	8.50	0.276
5c	2.5	8.49	0.277
6a	1.5	9.50	0.24
6b	2.0	9.50	0.233
6c	2.5	9.50	0.234



Both C_T and C_W data from a series of tank test experiments on scale models were available from *Molland (1994),* in four catamaran configurations (*S/L*=0.2, 0.3, 0.4 and 0.5) for each model. In this work the ANN was fitted directly to the C_T data, rather than deriving C_W for each model. Over the past decade there has been considerable discussion over the breakdown of C_T into its F_n and R_e dependent components for these types of catamaran vessels. The decision to fit the model C_T data was made as it allows the reader to select their own scaling method.

It is essential when training an ANN that an unbiased estimate of the generalisation error be available. In order to achieve this, it is necessary to partition the data into mutually exclusive training and validation sets. Data from the validation set are held out from the training process and are used to estimate the error between the fitted neural network and the original function. The network architecture and early stopping of the training process can be determined using the validation set error. However this can contain some bias toward the chosen network architecture and as such a separate, independent test set should also be kept aside for a truly unbiased estimate of the network error.

The resistance data available for neural network training was randomly partitioned into mutually exclusive training (85%), validation (13%) and test (2%) sets. Because of the small number of data points in the dataset it was not possible to allocate a large number of points to the validation and test sets, and these were set to reasonable minimum values based on recommendations from *Guyon (1997)*

2.5. ANN Architecture

The initial test network architectures were constrained to three layers, one for input, hidden and output layers respectively. This architecture was chosen as previous research indicated that multiple hidden layers are rarely effective in terms of both accuracy and training speed, *Neocleous and Schizas (1995).*

A search was performed using the architecture search function in NeuroIntelligence. This allowed multiple networks to be trained with different numbers of hidden layer neurons and the results collated to display the optimum network. Training runs used the Quasi-Newton method for 50,000 iterations, and 10 retrains were performed for each network topology to minimise error. The range of architectures searched was from 4-4-1 (i.e. 4 inputs, 4 hidden layer neurons and 1 output), to 4-17-1 (i.e. 4 inputs, 17 hidden layer neurons and 1 output).

Fig.2 displays the results of the search process with both the Average Absolute Error (AAE) and the standard deviation of the results reducing as the number of hidden layer neurons increased. Improvement in error values tended to level off at about 10 hidden layer neurons, with improvements from that point onward being small.

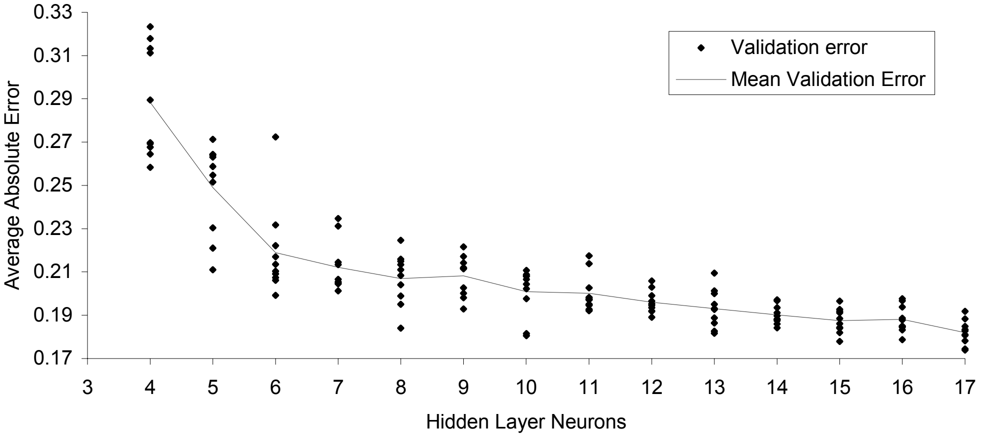


Fig.2: Validation set error for 10 retrains of each network architecture

It is important that both training and test set error be in broad agreement with the validation set error, and it can be seen from Fig.3 that the three sets of values correlate well as the number of hidden layer neurons increases.

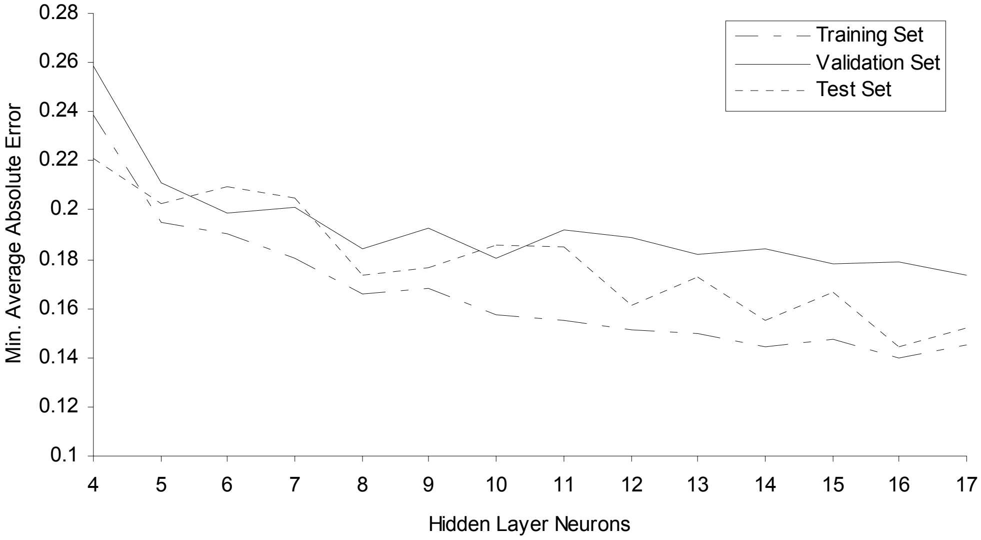


Fig.3: Minimum error values

Neural networks trained with large numbers of hidden layer neurons tend to suffer from overfitting. This overfitting cannot always be determined from the minimum validation set error if the dataset is small. One way of minimising the risk of overfitting is to select the network architecture based on a criterion that balances minimum error against network complexity. One such measure is Akaike's Information Criterion, *Akaike (1974)*, which has been widely used for model selection for both conventional statistical models and neural networks.

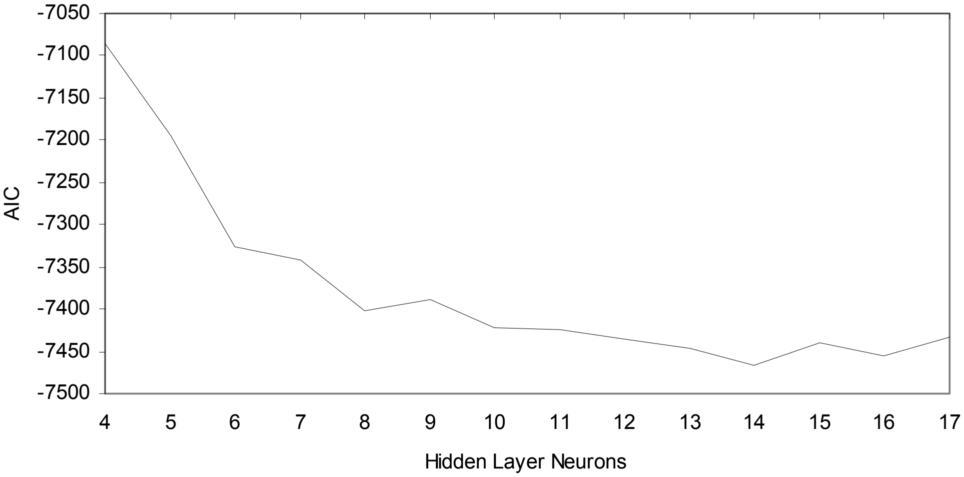


Fig.4: Akaike's Information Criterion

A graph of Akaike's criterion for the trained networks, Fig.4, shows a minimum value at 14 hidden layer neurons and it can be concluded that this network architecture is a reasonable compromise between error values and network complexity. An additional consideration is that the standard deviation of the Average Absolute Error values for the validation set is at a minimum at 14 hidden layer neurons, which gives some confidence that these results will be repeatable in subsequent training.

Couser et al. (2004) found that a hidden layer containing approximately 15 neurons was optimal for this dataset, and it is encouraging that despite the different treatment of the data, the new analysis is in broad agreement with this conclusion. In this work architectures containing two hidden layers were investigated but were not found to be advantageous, except for an indication that architectures with two hidden layers may be effective if the number of neurons in the first hidden layer was small.

As a result, an investigation was performed to determine whether network architectures with the same or less total number of degrees of freedom could be found that would outperform the single layer architecture. It was felt that if these degrees of freedom could be redistributed into two hidden layers the ability of the ANN to model a highly non-linear dataset might improve.

As the total number of two layer architectures was high it was not feasible to search all possible configurations. Instead architectures were investigated that had either the same total number of hidden layer nodes as the previously determined optimum (i.e. 14), one less total hidden layer nodes than the optimum (i.e. 13), two less total hidden layer nodes than the optimum (i.e. 12) or three less total hidden layer nodes than the optimum (i.e. 11). For each of these total numbers of hidden layer neurons different architectures were evaluated containing between 1 and 8 neurons in the second hidden layer.

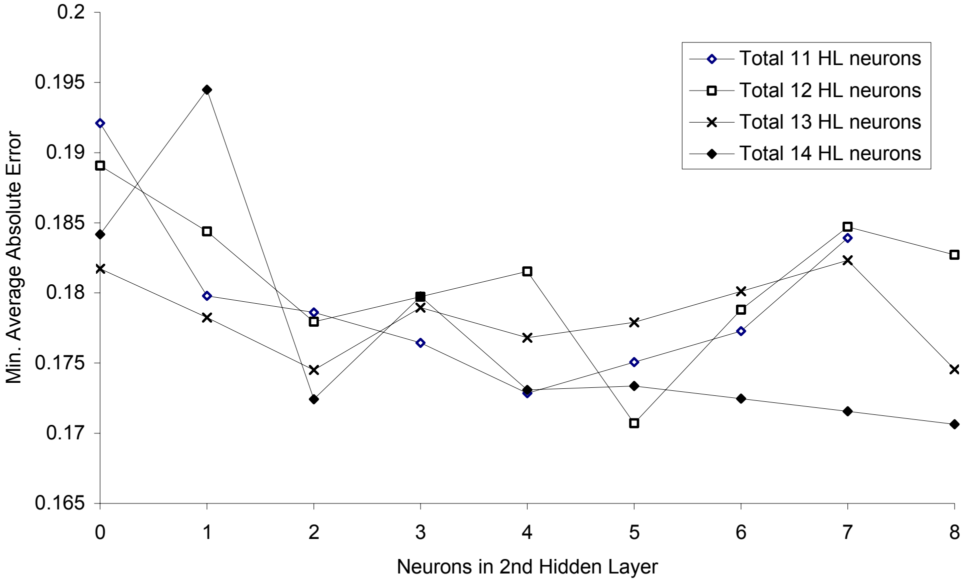


Fig.5: Minimum error values for two hidden layers

Fig.5 clearly shows a trend towards lower validation set errors when more than one neuron is placed on a second hidden layer. The networks containing a total of 11 and 12 hidden layer neurons have minima at 4 and 5 second layer neurons respectively, while the networks with a total of 13 and 14 hidden layer neurons continued to improve up to the point where there were 8 neurons in the second hidden layer, the maximum tested.

On the basis of this testing, the network selected used a 4-7-5-1 architecture. This network had training and validation set errors only slightly higher than the 4-14-1 architecture, but scored significantly better on Akaike's criterion due to it having two fewer nodes in the hidden layers.

Appendix 1 displays sample C_T curves derived from this ANN architecture for models 4b, 6a and 6b. These models have been selected for illustration as they have proven to be the configurations most difficult to fit in previous work, due to the combination of significant noise and nonlinearities in the F_n axis.

3. Optimisation using Genetic Algorithms

An ANN derived from the catamaran resistance data permits C_T to be estimated when a set of specific input parameters is defined. However it is often desirable to fix the value of one or more parameters and then determine the optimum values of the remaining parameters.

To achieve this, a search or optimisation method is required. This process is complicated in the case of catamaran resistance due to the interaction of wave patterns between the two demi-hulls, which may cause significant non-linearities in the vessel's resistance and an associated likelihood of a multi-modal solution space, *Molland et al. (1994)*. In this case it is necessary to use a method capable of global optimisation to find the minimum resistance.

One such method is the Genetic Algorithm (GA). GAs are a search and optimisation method based on the process of biological evolution. GAs differ from traditional optimisation techniques in that they involve a search from a population of solutions, not from a single point.

Each iteration of a GA involves a competitive selection that penalises poor solutions. The solutions with high fitness are recombined with others to produce members of the next generation, which inherit properties from their parents. Solutions are also mutated by making small, random changes to one or more free parameters. Recombination and mutation are used to generate new solutions, which are biased towards regions of the space for which good solutions have already been seen.

The steps involved in a genetic algorithm are as follows:

```
Initialise the population
Evaluate initial population
REPEAT
     Perform competitive selection
     Apply genetic operators to generate new solutions
     Evaluate solutions in the population
UNTIL some convergence criteria is satisfied
```

One of the strengths of GAs is that they perform well on noisy solution spaces where there may be multiple local optima. GAs tend not to get stuck on a local minima and can often find the global optimum.

GAs have been used by several researchers to optimise catamaran designs, *Doctors and Day (2000), Tuck and Lazauskas (1996), Hearn and Wright (1998).* However these have calculated resistance directly using first principles analysis. One weakness of GAs is that they are slow due to the high number of evaluations of the objective function required. This means that the performance penalty implicit in first principle analysis of resistance is a significant impediment to effective optimisation using GAs.

The use of an ANN based objective function allows the fitness to be evaluated in a small fraction of the time taken to perform first principles analysis and permits the GA optimisation to complete in a reasonably small amount of time. It also allows a researcher to use existing experimental data that may have been derived from tank testing which cannot be reproduced easily.

4. GA parameter settings

4.1. Exploration versus exploitation

A GA based search can be viewed as a trade off between the maintenance of sufficient diversity in the population to permit the coverage of the entire solution space (exploration), and the need to converge on the optimum solution within an acceptable time (exploitation), *Eiben and Schippers (1998).*

There are several variables or algorithms that can be applied to manipulate the degree of exploration or exploitation that the GA exhibits, *Goldberg (1989)*. These include:

- Selection method
- Elitism
- Population niching - fitness sharing

4.2. Selection

Much research has been done into different selection methods with two of the most widely used methods being roulette wheel selection and tournament selection. Roulette wheel selection works by choosing two individuals from the population and comparing their fitness scores. The individual with the highest fitness score is selected to be a parent for an individual in the subsequent generations.

Roulette wheel selection assigns a likelihood of representation in the following generation proportional to the magnitude of an individual's fitness score. The difficulty inherent in the use of roulette wheel selection is that the likelihood of representation in the following generation is dependent on the scaling factor used to scale the results of the objective function to create fitness scores. Changing this scaling factor can dramatically change the results of the selection process, and it is difficult to determine the correct scaling factor ahead of time.

On the other hand, tournament selection is a simple comparison of fitness values and is independent of their scaling. As a result, tournament selection is both easier to implement and more robust in use than roulette wheel selection.

4.3. Elitism

Elitism ensures that the next generation's best individual will be at least as good as any from the previous generation by automatically including the best individual from the previous population, *Andris and Frollo (2002).*

4.4 Fitness sharing

Fitness sharing works by scaling the fitness of individuals based on how similar they are to all other individuals in the population. Individuals that are very similar to others in the population are given a slight penalty, while individuals that show novel features are rewarded by increasing their fitness relative to the population. This is intended to ensure that the population maintains sufficient diversity to avoid premature convergence on false optima, *Goldberg (1987)*.

5. GA based optimiser prototype

A prototype GA based optimisation framework, HullGA, has been developed by the authors in order to evaluate the feasibility of using an ANN model for the objective function of an optimisation process. The GA code used for the prototype developed by the authors was developed and tested in MATLAB 5.3 *(The MathWorks 2003).* The objective function used is the predicted total resistance coefficient (C_T) for the catamaran configuration at a selected F_n.

A design was adopted that allowed some experimentation with different parameters which may affect the evolution process. Parameters which can be adjusted via the user interface are –

- Population size
- Limit on number of generations evaluated
- Selection method – roulette wheel or tournament selection
- Mutation rate
- Elitism - on or off
- Fitness sharing - on or off

The prototype also codifies specific exploration and exploitation phases, each of which used their own settings for the above parameters.

A preliminary evaluation of the effects of the interplay of the different evolution parameters has been conducted, but no comprehensive study has been done due to the large amount of time required. Initial conclusions are as follows –

5.1. Population size

The goal of testing population size is to ensure that the population is large enough to maintain sufficient diversity to avoid premature convergence.

Tests were run for population sizes ranging between 10 and 150. Negligible differences in rate of convergence or the fitness level of the best solution were observed for populations greater than 20 individuals. From these results, it appears that there is no advantage to be gained from a population size greater than 20 for this particular dataset.

5.2. Selection method

Much research has been done into different selection methods and there is support in the literature for both roulette wheel selection and tournament selection.

In the case of the HullGA prototype it was found that tournament selection gave the best results, primarily due to the difficulty of choosing suitable fitness scaling parameters to avoid premature convergence when using roulette wheel selection.

5.3. Elitism

The use of elitism during the exploration phase may lead to premature convergence in a multi-modal solution space. Initial experimentation with the HullGA optimiser indicated that it may be best to use elitism only during the exploitation phase of the optimisation process.

5.4. Fitness sharing

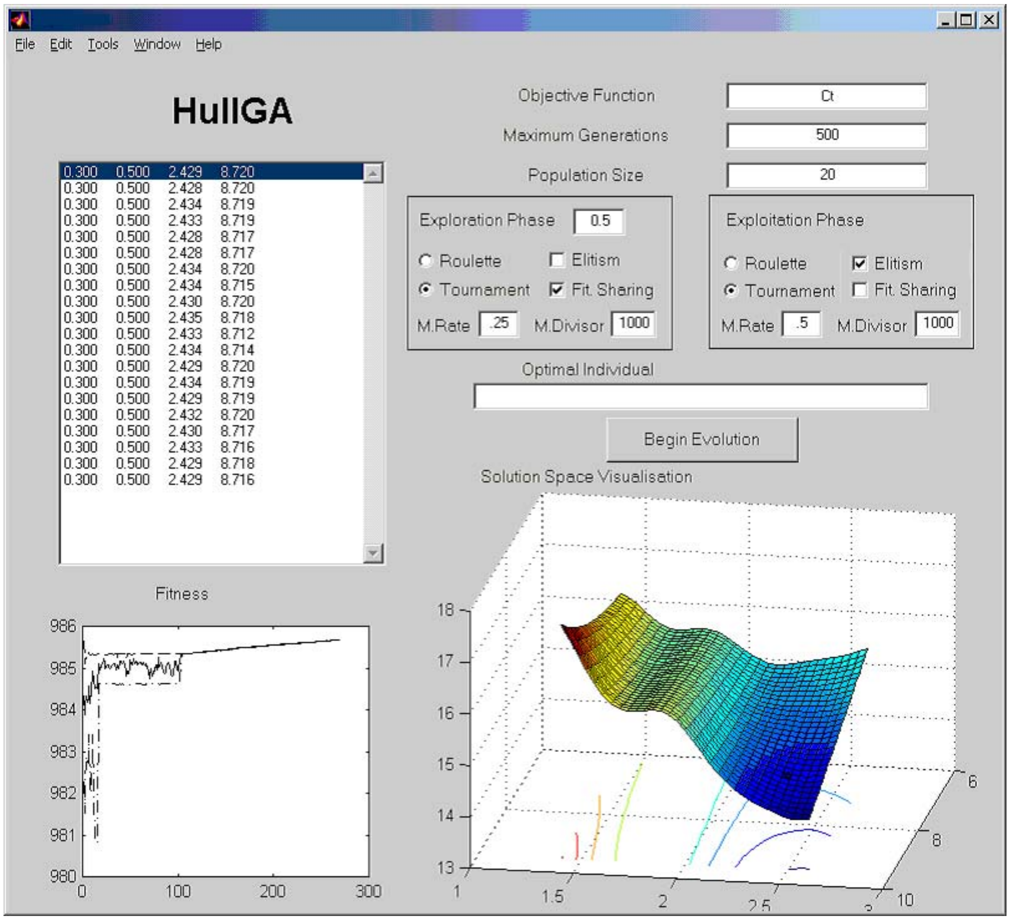
Fitness sharing has been shown to be effective in maintaining population diversity during the exploration phase, which helps to prevent premature convergence on local rather than global optima. During the exploitation phase, however, this diversity may inhibit convergence to an optimum solution and thus fitness sharing may not be useful during this phase.

Although roulette wheel selection appears to benefit from fitness sharing, previous research has found that fitness sharing can display chaotic interactions with tournament selection, with unexpected results, *Oei et al. (1991)*. This interaction between evolution parameters is an area for future investigation.

6. Example catamaran optimisation

Based on the test results, a GA with tournament selection and fitness sharing in the exploration phase, and tournament selection and elitism in the exploitation phase, was used to optimise catamaran hull forms for minimum resistance The exploration and exploitation phase were evenly divided over the total number of generations.

Fig.6 demonstrates an example at Fn 0.3. In this case resistance was optimised based on range constraints for S/L Ratio (0.15 to 0.5), B/T ratio (1.5 to 2.5), and $L/\nabla^{1/3}$ ratio (6.27 to 9.51) as described in *Molland et al. (1994).*



Objective Function: Ct
Maximum Generations: 500
Population Size: 20

Exploration Phase: 0.5
○ Roulette ☐ Elitism
● Tournament ☑ Fit. Sharing
M.Rate: .25 M.Divisor: 1000

Exploitation Phase
○ Roulette ☑ Elitism
● Tournament ☐ Fit. Sharing
M.Rate: .5 M.Divisor: 1000

Optimal Individual:

```
0.300   0.500   2.429   8.720
0.300   0.500   2.428   8.720
0.300   0.500   2.434   8.719
0.300   0.500   2.433   8.719
0.300   0.500   2.428   8.717
0.300   0.500   2.428   8.717
0.300   0.500   2.434   8.720
0.300   0.500   2.434   8.715
0.300   0.500   2.430   8.720
0.300   0.500   2.435   8.718
0.300   0.500   2.433   8.712
0.300   0.500   2.434   8.714
0.300   0.500   2.429   8.720
0.300   0.500   2.434   8.719
0.300   0.500   2.429   8.719
0.300   0.500   2.432   8.720
0.300   0.500   2.430   8.717
0.300   0.500   2.433   8.716
0.300   0.500   2.429   8.718
0.300   0.500   2.429   8.716
```

Fig.6: The HullGA user interface

Due to the multi-dimensional solution space, in this case four free parameters, it is not feasible to visualise the complete surface. The 3D surface plot in Fig.6 captures C_T values (vertical axis) against B/T ratio and $L/\nabla^{1/3}$ ratio, at a fixed S/L ratio, equal to the average S/L ratio of the population.

The graph in the lower left corner of Fig.6 illustrates the best, worst and average fitness values over the generations evaluated. The highest fitness values in the population are indicated by the dashed line (upper), the average fitness values by the solid line (middle), and the lowest fitness values by the dash-dot line (lower).

The initial oscillations in the fitness graph illustrate the exploration of the solution space using fitness sharing. As the three fitness plots meet it can be seen that the GA is converging to an optimal part of the solution space.

7. Future work

The work described here has used a GA to find the optimal solution in a problem space with four free parameters. More useful results could be achieved if additional catamaran hull parameters were included for optimisation, such as longitudinal centre of buoyancy position (LCB), prismatic or block coefficients (C_P or C_B), or midship area coefficient (C_M)

Another natural extension to the work described here is to extend the GA to perform multi-objective optimisation. Example objectives might be the simultaneous optimisation of both catamaran resistance and seakeeping qualities.

8. Conclusions

The work detailed in this paper has demonstrated that a sparse dataset with a relatively high degree of noise can be fitted effectively with a feed-forward neural network. In addition we have demonstrated that there may be benefits resulting from the investigation of networks with two hidden layers rather than one. In particular it may be possible to find network architectures that have fewer degrees of freedom combined with lower error levels.

The work detailed in this paper has also demonstrated that a combination of GAs and ANNs can be successfully used as an optimisation tool for catamaran design parameters. In this problem domain it appears that using tournament selection alone during the exploration phase, and tournament selection with elitism in the exploitation phase gives the best results.

This research has also shown that the success of a GA is closely related to the selection methods and other parameters used. However, a particular combination of parameters may not produce a successful optimiser in a different problem domain. In particular, if the solution space becomes more multi-modal in nature, the importance of selecting the correct GA parameters increases.

References

AKAIKE, H. (1974), *A New Look At The Statistical Model Identification.* IEEE Trans. Auto. Control., 19, 723

ALYUDA (2005), *Neurointelligence Manual*, http://www.alyuda.com

ANDRIS, P.; FROLLO, I. (2002), *Optimisation of NMR Coils by Genetic Algorithm,* Measurement Science Review 2/2

COUSER, P.R.; MASON, A.P.; MASON, G.; SMITH, C.R.; VON KONSKY, B.R. (2004), *Artificial Neural Networks for Hull Resistance Prediction*, 3rd COMPIT, Siguenza

DARPA (1998), *Neural Network Study*, AFCEA International Press

DOCTORS, L.J.; DAY, A.H. (2000), *The Survival of the Fittest – Evolutionary Tools for Hydrodynamic Design of Ship Hull Form,* Royal Inst. Naval Arch.

EIBEN, A.E.; SCHIPPERS, C.A. (1998), *On evolutionary exploration and exploitation,* Fundamenta Informaticae 35

GOLDBERG, D.E. (1989), *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley Inc.

GOLDBERG, D.E.; RICHARDSON, J. (1987), *Genetic algorithms with sharing for multimodal function optimization*, 2nd Int. Conf. Genetic Algorithm

GUYON, I. (1997), *A scaling law for the validation_set training_set size ratio*, Bell Laboratories

HEARN, G.E.; WRIGHT, P.N.H. (1998), *Michell wave resistance characteristics resulting from the optimisation of a catamaran hullform via genetic algorithms*, EMAC'98

INSEL, M.; MOLLAND, A. (1992), *An investigation into the resistance components of high speed displacement catamarans*, Trans. RINA 134(A)

JAIN, A.K.; MAO, J.C.; MOHIUDDIN, K.M. (1996), *Artificial neural networks: A tutorial*, Computer 29(3), pp.31-44

KOUSHAN, K. (2003), *Artificial neural networks for prediction of ship resistance and wetted surface area,* Marintek

MATLAB version 5.3: The MathWorks [Online] http://www.themathworks.com.

MOLLAND, A.F.; WELLICOME, J.F.; COUSER, P.R. (1994), *Resistance experiments on a systematic series of high speed displacement catamaran forms: variation of length-displacement ratio and breadth-draught ratio,* Ship Science Report 71, Dept. Ship Science, Univ. of Southampton

NEOCLEOUS, C.; SCHIZAS, C. (1995), *Artificial neural networks in marine propeller design*, ICNN'95, Int. Conf. on Neural Networks, Vol. 2, pp.1098-1102, New York, IEEE Computer Society Press

OEI, C.K.; GOLDBERG, D.E.; CHANG, S.-J. (1991), *Tournament selection, niching, and the preservation of diversity.* IlliGAL Techn. Report 91011, Univ. of Illinois at Urbana-Champaign

SARLE, W.S. (1994), *Neural networks and statistical models,* 19th Annual SAS Users Group Int. Conf.

SCHIZAS, C.N., NEOCLEOUS, C.C. (1995), *Artificial Neural Networks in marine propeller design*, Int. Conf. Neural Networks

TUCK, E.O.; LAZAUSKAS, L. (1996), *Unconstrained ships of minimum drag*, Dept. Applied Mathematics Techn. Report, The University of Adelaide

Appendix 1

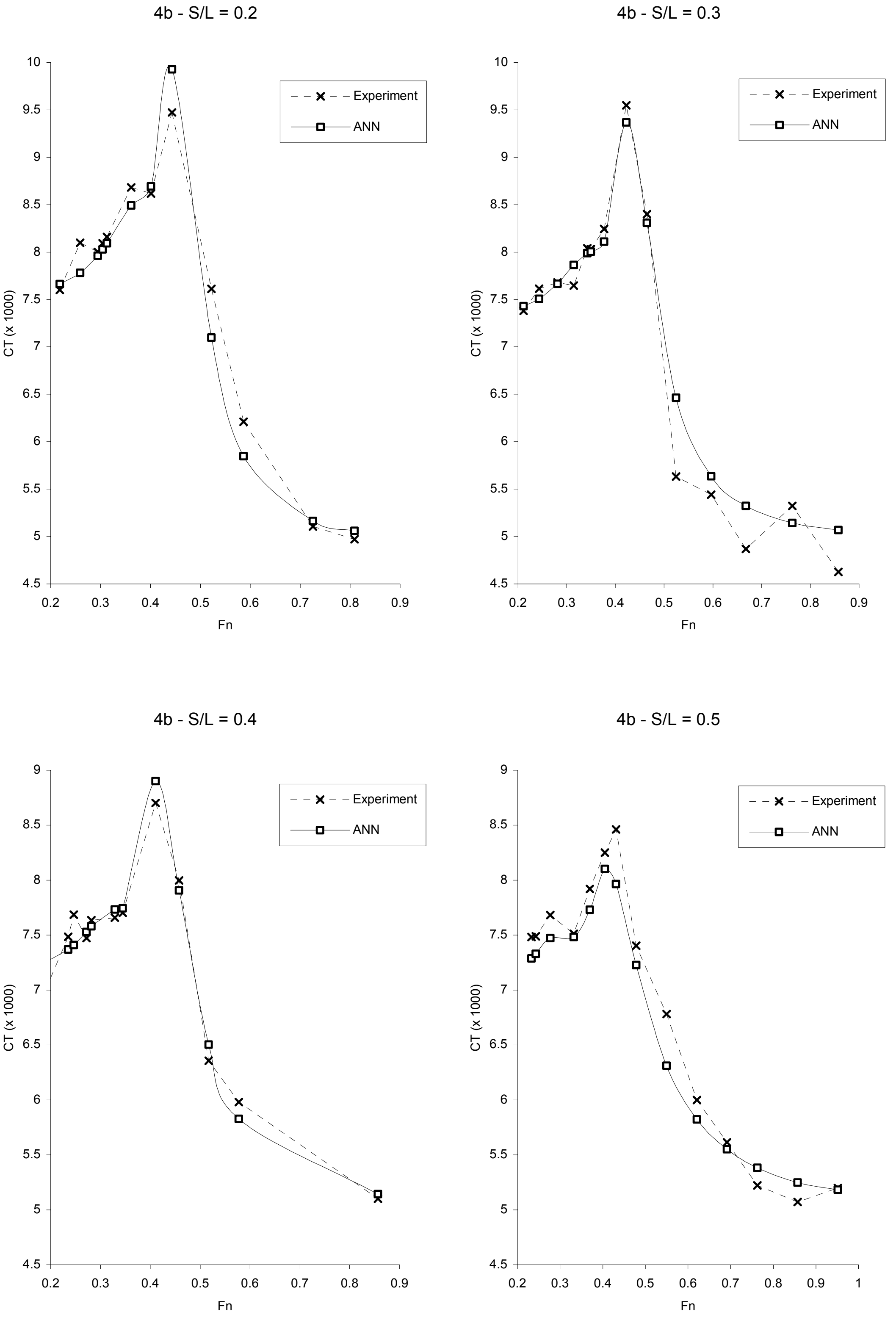


Fig.7: C_T curves for model 4b

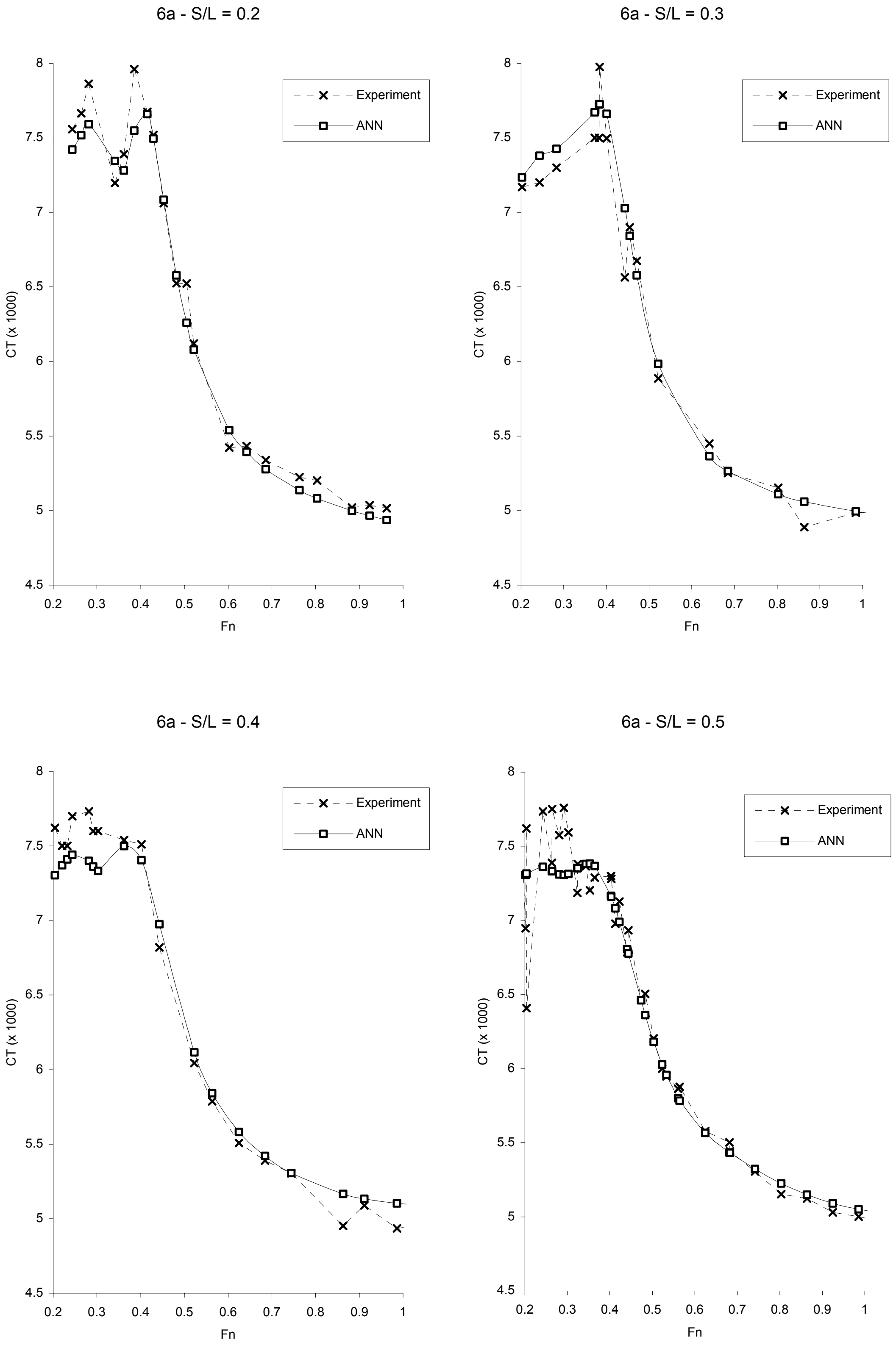


Fig.8: C_T curves for model 6a

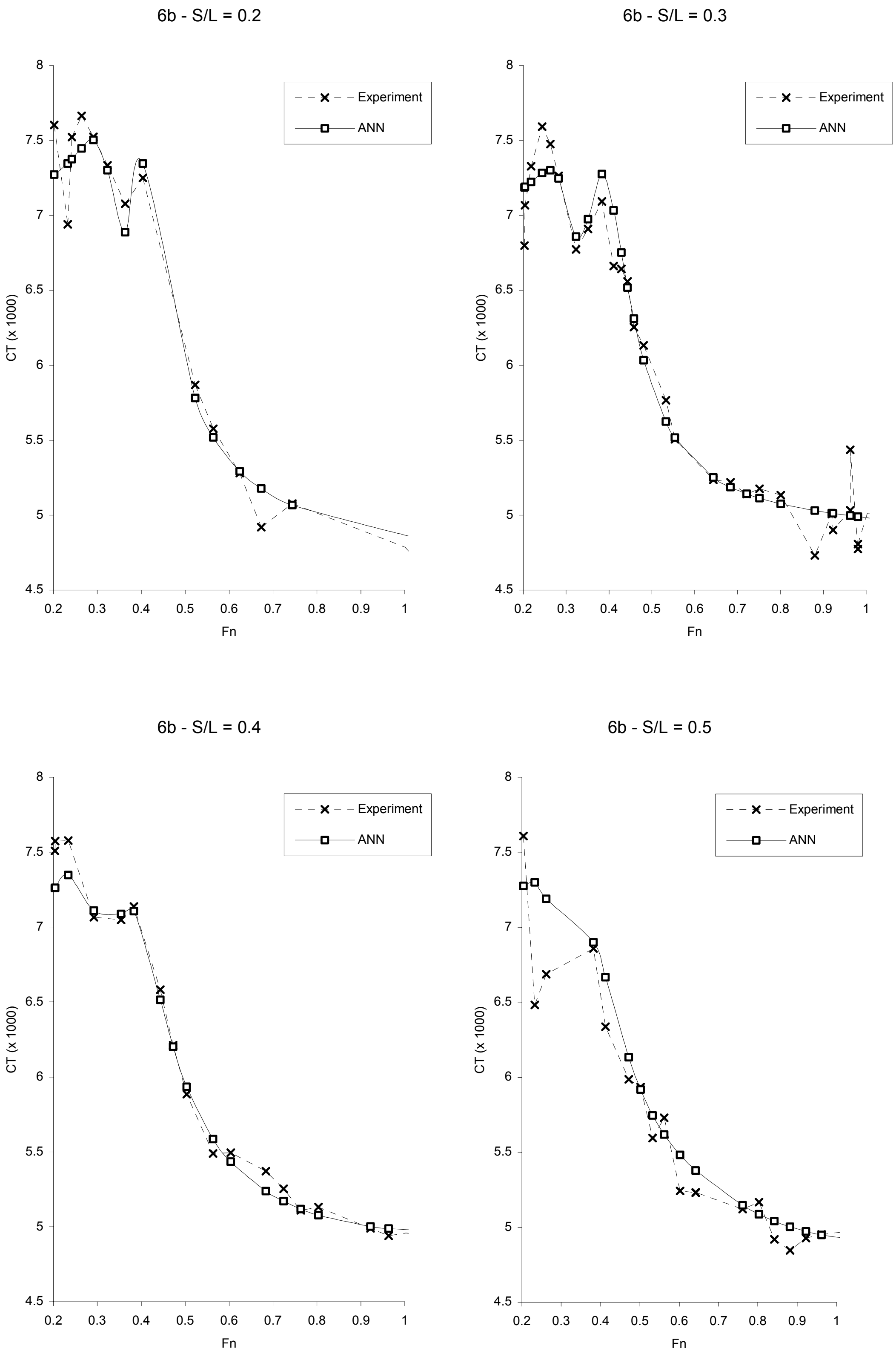


Fig.9: C_T curves for model 6b