PERGAMON

Computers & Operations Research 29 (2002) 849–868

computers & operations research

www.elsevier.com/locate/dsw

Estimation of all-terminal network reliability using an artificial neural network

Chat Srivaree-ratana, Abdullah Konak, Alice E. Smith*

Department of Industrial and Systems Engineering, 207 Dunstan Hall, Auburn University, Auburn, AL 36849-5346, USA

Received 1 June 1999; received in revised form 1 July 2000

Abstract

The exact calculation of all-terminal network reliability is an NP-hard problem, with computational effort growing exponentially with the number of nodes and links in the network. During optimal network design, a huge number of candidate topologies are typically examined with each requiring a network reliability calculation. Because of the impracticality of calculating all-terminal network reliability for networks of moderate to large size, Monte Carlo simulation methods to estimate network reliability and upper and lower bounds to bound reliability have been used as alternatives. This paper puts forth another alternative to the estimation of all-terminal network reliability — that of artificial neural network (ANN) predictive models. Neural networks are constructed, trained and validated using the network topologies, the link reliabilities, and a network reliability upperbound as inputs and the exact network reliability as the target. A hierarchical approach is used: a general neural network screens all network topologies for reliability followed by a specialized neural network for highly reliable network designs. Both networks with identical link reliability and networks with varying link reliability are studied. Results, using a grouped cross-validation approach, show that the ANN approach yields more precise estimates than the upperbound, especially in the worst cases. Using the reliability estimation methods of the ANN, the upperbound and backtracking, optimal network design by simulated annealing is considered. Results show that the ANN regularly produces superior network designs at a reasonable computational cost.

Scope and purpose

An important application area of operations research is the design of structures, products or systems where both technical and business aspects must be considered. One expanding design domain is the design of computer or communications networks. While cost is a prime consideration, reliability is equally important. A common reliability measure is all-terminal reliability, the probability that all nodes (computers or terminals) on the network can communicate with all others. Exact calculation of all-terminal reliability is an

* Corresponding author. Tel.: + 1-334-844-1400; fax: + 1-334-844-1381.
E-mail address: aesmith@eng.auburn.edu (A.E. Smith).

NP-hard problem, precluding its use during optimal network topology design, where this calculation must be made thousands or millions of times. This paper presents a novel computationally practical method for estimating all-terminal network reliability. Is shown how a neural network can be used to estimate all-terminal network reliability by using the network topology, the link reliabilities and an upperbound on all-terminal network reliability as inputs. The neural network is trained and validated on a very minute fraction of possible network topologies, and once trained, it can be used without restriction during network design for a topology of a fixed number of nodes. The trained neural network is extremely fast computationally and can accommodate a variety of network design problems. The neural network approach, an upper bound approach and an exact backtracking calculation are compared for network design using simulated annealing for optimization and show that the neural network approach yields superior designs at manageable computational cost. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Neural network; Network reliability; Network design; Simulated annealing

1. Introduction

Reliability and cost are two important considerations when designing communications networks, especially backbone telecommunications networks, wide area networks, local area networks and data communications networks located in industrial facilities. If the *nodes* (stations, terminals or computer sites) of the network are fixed, the main design decisions are selection of the type and routing of *links* (cables or lines) of the network to ensure proper and reliable operation while meeting cost objectives. The following typically define the problem assumptions:

1. The location of each network node is given.
2. Nodes are perfectly reliable.
3. Link costs and reliabilities are fixed and known.
4. Each link is bi-directional.
5. There are no redundant links in the network.
6. Links are either operational or failed.
7. The failures of links are independent.
8. No repair is considered.

There are many versions of the network design problem in the literature. Most involve reliability and cost, while sometimes capacity, delay, routing distance and connectivity are included. Mathematically, the design optimization problem for a minimum cost network that meets a minimum reliability requirement can be expressed as

$$\begin{aligned} &\text{Minimize} \quad Z(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} x_{ij}, \\ &\text{s.t.} \qquad\quad R(\mathbf{x}) \geqslant R_o, \end{aligned}$$

where N is the number of nodes; (i, j) the link between nodes i and j; x_{ij} the decision variable, e.g., $x_{ij} \in \{0, 1\}$ for networks with identical link reliability; $\mathbf{x}$ the link topology of $x_{12}, \ldots, x_{ij}, \ldots, x_{N-1,N}$;

$R(\mathbf{x})$ the reliability of $\mathbf{x}$; R_o the network reliability requirement; Z the objective function and c_{ij} the cost of (i, j).

The network topology design problem has been studied in the literature with both enumerative-based methods (usually a variation of branch-and-bound) [1] and heuristic methods [2–8]. A common aspect of these optimization methods is that network reliability must be calculated for each and every candidate topology identified, usually running to millions or greater. The search space size of possible network topologies is

$$k^{\frac{(|N| \times (|N|-1))}{2}}, \tag{1}$$

where k is the number of choices for the links (assuming the links have the same number of choices). There are always at least two choices — 0 for no link present and 1 for a link present — between any pair of nodes, i, j, in the network. In many applications k is greater than two. For example, a designer might choose between a single link connecting two nodes or two links (one in redundancy) connecting the two nodes. In this case $k = 3$ and the redundancy choice is more costly and more reliable. In other situations, a designer might choose between grades of fiber optic cable or between different sheathing options, also having cost/reliability trade offs. A 10 node network ($N = 10$) with identical links ($k = 2$) has 3.5×10^{13} possible designs. A 10 node network with five alternative link choices has 10^{35} possible designs. Clearly for networks of realistic size, a computationally expedient alternative to the exact network reliability calculation must be found to use during the design optimization procedure.

The network design problem is especially difficult when considering *all-terminal* network reliability (also called *uniform* or *overall* network reliability), defined as the probability that all nodes can communicate with all other nodes. (This is equivalent to all-terminal stationary availability when a mission time is not implicitly assumed.) Because each link has a reliability, p, and a non-zero unreliability $q = 1 - p$, at any time instant, only some links of $\mathbf{x}$ might be operational. A state of $\mathbf{x}$ is a sub-graph $(N, \mathbf{x}')$ where $\mathbf{x}'$ is the set of operational links such that $\mathbf{x}' \subseteq \mathbf{x}$. The overall network reliability of state $\mathbf{x}' \subseteq \mathbf{x}$ is

$$\mathrm{R}(\mathbf{x}) = \sum_{\Omega} \left[\prod_{j \in \mathbf{x}'} p(x_j) \right] \left[\prod_{j \in (\mathbf{x} \setminus \mathbf{x}')} q(x_j) \right], \tag{2}$$

where $\Omega =$ all operational states. However, the exact calculation of all-terminal network reliability is NP-hard, that is, computational effort increases exponentially with network size [9] because of the exponential growth in states. One way to calculate the exact network reliability is to enumerate all possible minimal cutsets of a network as in Ball and Van Slyke [10]. Other similar approaches to exactly calculating network reliability are given in Aggarwal and Rai [11], Cavers [12] and Rai [13]. These methods are not computationally practical for large networks since the fundamental step of enumeration of minimal cutsets is NP-hard [14]. Monte Carlo stochastic simulation methods can estimate network reliability very precisely [15,16], however, simulation must be repeated numerous times to ensure a good estimate. Therefore, the simulation approach also incurs significant computational effort when estimating the reliability of the network, especially for highly reliable networks where failures are rare.

2. Neural networks for reliability estimation

Neural networks were inspired by the power, flexibility and robustness of the biological brain. They are computational (mathematical) analogs of the basic biological components of a brain — neurons, synapses and dendrites. Artificial neural networks (hereafter referred to as ANN) consist of many simple computational elements (summing units — *neurons* — and weighted connections — *weights*) that work together in parallel and in series. Function approximation neural networks begin in a random state and "learn" using repeated processing of a training set, that is, a set of inputs with target outputs. Learning occurs because the error between the ANN output and the target output is calculated and used to adjust the weighted synapses. This continues until errors are small enough or no more weight changes are occurring. The ANN is then trained and the weights are fixed. The trained ANN can be used for new inputs to perform function approximation or classification tasks.

While the original inspiration was the biological brain, a function approximation ANN can also be regarded as a statistic, and there are many strong and important parallels between the field of statistics and the field of ANN [17,18]. The process of training the ANN using a data set is an analog to computing a vector-valued statistic from that data set. Just as a regression equation's coefficients (viz., slopes and intercepts) are calculated by minimizing squared error over the data set, ANN weights are determined by minimizing error over the data set. However, there are also important dissimilarities between statistics and ANN. ANN have many free parameters (i.e., weighted connections). An ANN with five inputs, an intermediate (*hidden*) layer of five neurons and a single output has 36 trainable weights, where a simple multiple linear regression would have six (five slopes and an intercept). ANN can accommodate redundant free parameters rather well, but there is significant danger in overfitting an ANN model [18]. An overfitted ANN would be strongly dependent on the data set (*sample*) used to build it, and may poorly reflect the underlying relationship (*population*). Therefore, thorough validation of ANN using data not used in training is essential.

An important property of a function approximation ANN, under certain conditions, is that it is a universal approximator [19–21]. This means that the bias associated with choosing a functional form, as is done in regression analysis when a linear relationship is selected, is minimized. This is a substantial advantage over traditional statistical prediction models, as the relationship between network topology and all-terminal reliability is highly non-linear with significant, but complex, interactions among the links.

In this paper, ANN are developed, or trained, based on the all-terminal reliability of a very small set of possible network topologies and link reliabilities, for a given number of nodes. The resulting ANN is used to estimate network reliability as a function of the link reliabilities and the topology during the search for the optimal topology. In this way, estimates of the reliability of numerous topologies are available without costly calculation or simulation. A disadvantage of using ANN as a reliability evaluator is that the reliability prediction is only an estimate that may be subject to bias and/or variance depending on the adequacy of the ANN. The functionality of using an ANN estimation of reliability during optimal network design is tested by comparing it to an easily calculated upperbound and a computationally expensive exact calculation. A similar approach was used for design of series-parallel systems when considering cost and reliability [22], however it had less practical utility because reliability of series-parallel systems can be exactly calculated quite easily with closed form mathematical expressions.

3. Training and validating the neural networks

A backpropagation training algorithm [23] was selected because of its powerful approximation capacity and it applicability to both binary and continuous inputs. The backpropagation algorithm minimizes the squared error between the ANN output and the target. A hyperbolic activation function was used in all neurons and a learning rate of 0.3 was set for hidden neurons and a learning rate of 0.15 was set for output neurons. Through experimentation, the number of hidden neurons was set identical to the number of input neurons and ANN were trained for 250,000 epochs, that is, 250,000 passes through the training set. The ANN were not sensitive to these values and the experimentation to identify a good set of ANN parameters was very brief. A standard ANN software package, neuralworks explorer [24], was used.

3.1. Networks with identical link reliability

Limiting the links chosen to be in a network topology to those with the same reliability (i.e., $k = 2$) simplifies the problem of estimating network reliability because the number of possible topologies grows exponentially with an increase in k (see Eq. (1)). In this case, if $x_{ij} = 1$, the link is chosen for the network topology and if $x_{ij} = 0$, no link is present. However, to make the ANN more applicable to a variety of design problems, five different values of link reliability were chosen to be included in a single ANN. For the problems studied, these link reliability values are 0.80, 0.85, 0.90, 0.95 and 0.99. To clarify, the ANN in this section would be appropriate for network design problems using any of these five link reliabilities, however, for a given design problem all links must have the same reliability. This is relaxed in the next section where networks with varying link reliabilities are considered.

The inputs to the ANN were:

1. The architecture of the network as indicated by a series of binary variables (x_{ij}).
   The length of the string of 0's and 1's is equal to $(N(N-1))/2$.
2. The link reliability (either 0.80, 0.85, 0.90, 0.95 or 0.99).
3. The calculated upperbound of network reliability using the method of [25].

The upperbound calculation, while adding computational effort of $O(N^3)$, significantly improved the estimation precision of the ANN. Without the upperbound as an input, the errors of the ANN reported in Section 4 were nearly doubled. Any upperbound could be used, such as that in Jan [26], however the one chosen is rather unique in that it is applicable to networks with links with different reliability values, and it has been shown to more precise than other bounds that could accommodate links of different reliabilities [25]. This property of accommodating links with different reliabilities will be important for the ANN described in the next section. The bound used is

$$R(\mathbf{x}) \leqslant 1 - \left[\sum_{i=1}^{n} \left(\left(\prod_{(k,i)\in E_i} (1 - p_{ki}) \right) \prod_{j=1}^{i-1} \left(1 - \frac{\prod_{(k,j)\in E_j}(1 - p_{kj})}{(1 - p_{ij})} \right) \right) \right], \tag{3}$$

where p_{ij} is the reliability of undirected link (i, j) and E_i is the set of links connected to node i.

The output of the ANN was the estimated all-terminal network reliability. For the training and validation sets, the target network reliability was the exact value as calculated using the backtracking technique of Ball and Van Slyke [10]. This procedure essentially enumerates the cutsets and calculates the network unreliability, $(1 - R(\mathbf{x}))$, as detailed below:

Step 0: (Initialization). Mark all links as free; create a stack that is initially empty.
Step 1: (Generate modified cutset)
(a) Find a set of free links that together with all inoperative links will form a network-cut.
(b) Mark all the links found in 1(a) inoperative and add them to the stack.
(c) The stack now represents a modified cutset; add its probability to a cumulative sum.
Step 2: (Backtrack)
(a) If the stack is empty, end.
(b) Take a link off the top of the stack.
(c) If the link is inoperative and if when made operative, a spanning tree of operative links exists, then mark it free and go to 2(a).
(d) If the link is inoperative and the condition tested in 2(c) does not hold, then mark it operative, put it back on the stack and go to Step 1.
(e) If the link is operative, then mark it free and go to 2(a).

A node size of 10 was chosen to investigate the approach of this paper. A set of 750 network topologies were randomly generated (ensuring each network formed at least a spanning tree, i.e., $R(\mathbf{x}) > 0$) with 150 observations of each link reliability. Remembering that the total number of network designs that could be handled by this ANN is 5×2^{45} or 1.7×10^{14}, 750 is an exceedingly small sample indeed. The upperbound of each network topology and the exact network reliability were calculated to use as an input and as the target output, respectively. After preliminary experiments, a network architecture of 47 inputs (45 possible arcs, the link reliability and the network upperbound), 47 hidden neurons in one hidden layer and a single output was used.

The data set was divided using a five-fold cross-validation technique so that five validation ANN were trained and one final application ANN was trained. The five validation ANN used 4/5's of the data set for training (600 observations) and the remaining 1/5 (150 observations) as testing, where the testing set changed with each validation ANN. Each training and testing set had equal proportions of each link reliability. The final application ANN was trained using all 750 members of the data set and its validation is inferred using the cross-validation ANN. This cross-validation approach provides an unbiased and quite precise estimate of ANN performance on the population of network topologies. While there are marginal improvements to be had by using a full cross validation (leaving one observation out each time), computationally it is quite burdensome. Other strategies such as bootstrapping are also computationally more intensive while suffering from bias. See Twomey and Smith [27] for a complete discussion of resampling validation techniques as used in backpropagation networks. The grouped cross-validation estimate of root mean squared error (RMSE) for the application ANN is

$$\mathrm{RMSE} = \sqrt{\frac{1}{750}\sum_{g=1}^{5}\sum_{h=1}^{150}\left(y_{(g-1)150+h} - \hat{f}[T_{(g)}, \mathbf{x}_{(g-1)150+h}]\right)^2}, \tag{4}$$

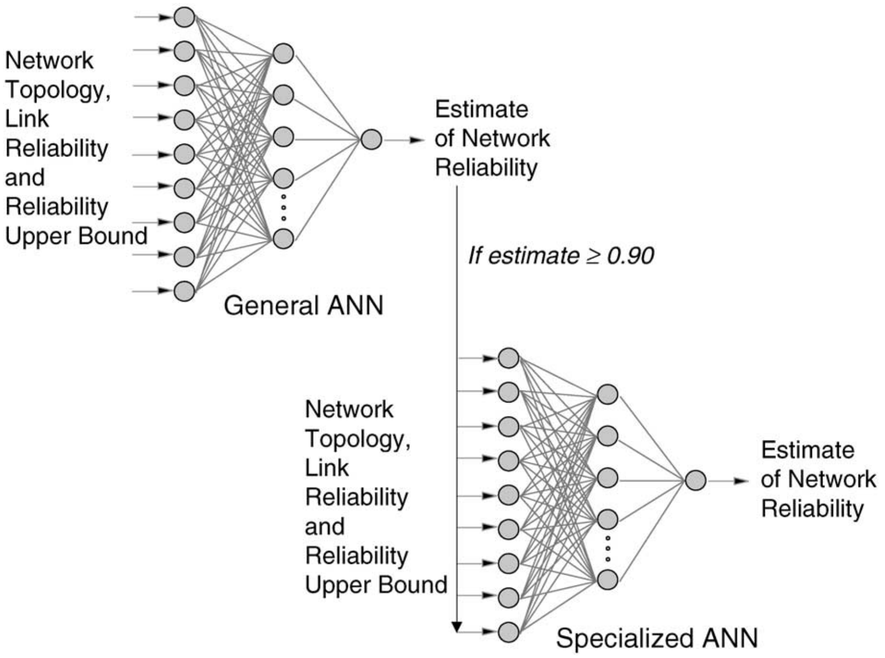
Network Topology, Link Reliability and Reliability Upper Bound
General ANN
Estimate of Network Reliability
If estimate ≥ 0.90
Network Topology, Link Reliability and Reliability Upper Bound
Specialized ANN
Estimate of Network Reliability

Fig. 1. The hierarchy of a general ANN and a specialized ANN.

where g indexes the group left out, h indexes the observations in the left out group, and the sample $T_{(g)} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{(g-1)150}, y_{(g-1)150}), (\mathbf{x}_{(g)151}, y_{(g)151}), \ldots, (\mathbf{x}_{750}, y_{750})\}$ is used to construct the ANN $\hat{f}[T_g, \mathbf{x}_{(g-1)150+h}]$.

A second strategy using a specialized ANN for highly reliable networks was also employed. Because most actual network topology designs will be highly reliable, it is important that the reliability estimation be precise when $R(\mathbf{x}) \geqslant 0.90$. If the first ANN (just described) estimated a reliability of 0.90 or greater, the network topology, the link reliability and the upperbound were input to the second, specialized ANN, as shown in Fig. 1. This ANN was trained on 250 randomly generated topologies (using the same five link reliabilities) that had actual all-terminal reliabilities of 0.90 or greater. As in the general ANN, there were equal number (50) observations of each link reliability in the data set. Also as in the general ANN, a five-fold grouped cross-validation procedure was used for ANN training and validation. The ANN architecture was the same as the first network. Using the network reliability estimate from the first ANN as an additional input to the specialized ANN was investigated, but this did not improve predictive performance of the specialized ANN.

3.2. Networks with varying link reliability

Allowing links of different reliability within a single network topology is an important real world consideration. It does greatly expand the number of possible network topologies, complicating both the network design problem and the estimation of all-terminal network reliability. Using the same set of five link reliabilities from Section 3.1, networks with mixes of these were randomly generated (using equal probability on each link type). For these networks, $k = 6$ (i.e., one of the five reliability values or 0, which indicates the link is not present in the network topology). To clarify, the ANN in this section would be appropriate for network design problems using any of these five link reliabilities in any combination. The binary topology inputs of Section 3.1 are no longer applicable. Instead, the reliability value of each link is input (0, 0.80, 0.85, 0.90, 0.95, 0.99) while the input of the single link reliability from Section 3.1 is eliminated, leaving 46 inputs for a 10 node network.

The inputs to the ANN were:

1. The architecture of the network as indicated by a series of real-valued variables (x_{ij}). The length of the string is equal to $(N(N-1))/2$.
2. The calculated upperbound of network reliability using the method of Konak and Smith [25].

As in Section 3.1, 750 randomly generated network topologies were used for training and validating the ANN and the exact all-terminal network reliability using backtracking [10] was used as the target. The ANN architecture used 46 hidden neurons in a single hidden layer and a single output.

Also, as in Section 3.1, the strategy of a general ANN for all network topologies and a specialized ANN for highly reliable ($\geqslant 0.90$) networks was used. The specialized networks were trained and validated using a set of 250 randomly generated topologies. Again, there were 46 inputs to the neural network, a single output and 46 hidden neurons in one hidden layer.

4. Computational results

4.1. Networks with identical link reliability

Tables 1 and 2 give the five-fold results in root mean squared error (RMSE) for the general ANN and the specialized ANN, respectively, for networks with identical link reliability (those described in Section 3.1). It can be seen that the ANN estimations always improve upon the upperbound

Table 1
Errors for the general neural network with identical link reliability

Fold	RMSE training	RMSE testing	RMSE upperbound
Set 1	0.03672	0.04260	0.08875
Set 2	0.03073	0.05004	0.08954
Set 3	0.03444	0.03067	0.07158
Set 4	0.03123	0.05666	0.07312
Set 5	0.03173	0.05131	0.08800
Average	0.03297	0.04626	0.08220

Table 2
Errors for the specialized neural network with identical link reliability

Fold	RMSE training	RMSE testing	RMSE upperbound
Set 1	0.00664	0.00688	0.01232
Set 2	0.00583	0.01271	0.01371
Set 3	0.00630	0.00892	0.00908
Set 4	0.00629	0.00795	0.00927
Set 5	0.00555	0.01125	0.01598
Average	0.00612	0.00954	0.01207

estimates, sometimes significantly. Furthermore, the errors of the specialized ANN are much less than that of the general ANN, allowing a more precise network reliability estimation for those topologies that are likely to be considered the best.

While RMSE is important because it is the mean squared error that the backpropagation algorithm minimizes, a network designer will be interested in the differences between the reliability estimate and the actual network reliability. Fig. 2 shows an example of one of the five-fold validations comparing estimation of the ANN on the test set with the actual reliability while Fig. 3 shows the same for the specialized ANN. It can be seen that the predictions of the ANN are unbiased and are quite precise. Where the general ANN is less precise (at $R(\mathbf{x}) \geqslant 0.90$), the specialized ANN does a much better job. The mean absolute error (MAE) of the application general ANN is 0.036 and the MAE of the application specialized ANN is 0.007. Of course, these errors may be positive or negative since an ANN is an unbiased estimator while the upperbound errors will always be positive.

Table 3 presents results of a statistical analysis comparing the reliability estimations of the ANN and the bound with the actual reliability calculated by the backtracking algorithm. The comparisons between all three methods were done with ANOVA and Tukey's test at $\alpha = 0.05$ and the two way comparisons were done with paired t tests. From the ANOVA analysis, the ANN and the actual reliabilities were statistically identical but the bound was different. Pairwise comparison of the three alternatives yielded statistical difference in every case at $\alpha = 0.05$, however the ANN is closer to the actual than is the bound.

4.2. Networks with varying link reliability

This problem is much harder than that described in the preceding section, however the ANN prediction performed well. Table 4 gives the results over the five-fold cross validation for the general purpose ANN while Table 5 gives the same results for the specialized ANN. It can be seen that the RMS error of the ANN is still significantly less than that of the upperbound. A graphic view shows that the ANN estimate is unbiased while the upperbound estimate is biased upwards, of course (Fig. 4). Figs. 5 and 6 show the absolute error of the ANN versus the error of the upperbound for the first fold of the test set for the general and specialized ANN, respectively. It can be easily seen that while the upperbound sometimes performs better than the ANN, the maximum

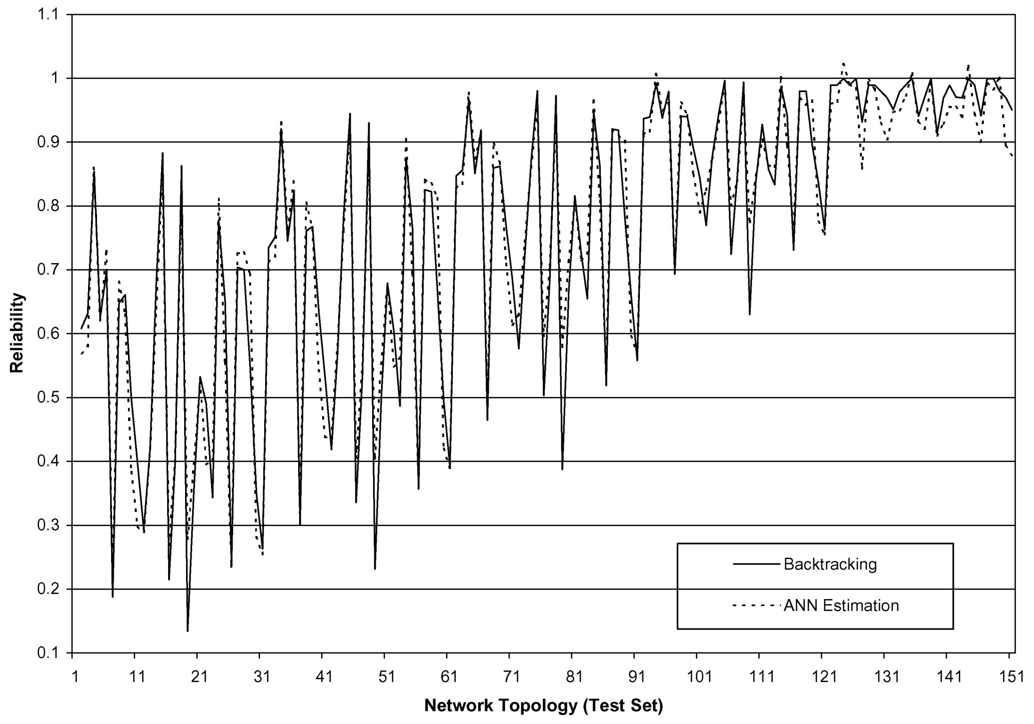

Fig. 2. General neural network estimation of reliability versus actual (backtracking) reliability on the fifth test fold for networks with identical link reliability.

errors of the ANN are much less. That is, the worst cases of the ANN are much better estimates than the worst cases of the upperbound. This has practical importance since a large error may badly mislead the optimization search where small errors will not. Therefore, the ANN can engender a more reliable search than the upperbound.

As in the preceding section, a statistical analysis of the estimations of the ANN and the upperbound with the exact network reliability are shown in Table 3. The statistical results are very similar to those of Section 4.1. From the ANOVA analysis at $\alpha = 0.05$, the ANN and the actual reliabilities were statistically identical but the bound was different. For the general ANN, with a pairwise comparison all three methods are statistically different, with the difference between the ANN and the actual smaller than that between the bound and the actual. For the specialized ANN, there is no statistical difference between the ANN and the actual however there is a statistical difference between the bound and the actual.

4.3. General remarks

Considering the relatively minute training set used for each ANN, the computational benefits of the approach become apparent. The ANN approach can now be used for any network design problem of 10 nodes with these five link reliabilities, in either an identical manner (Sections 3.1 and 4.1) or in a mixed manner (Sections 3.2 and 4.2). Considering the vast search spaces of each 10 node

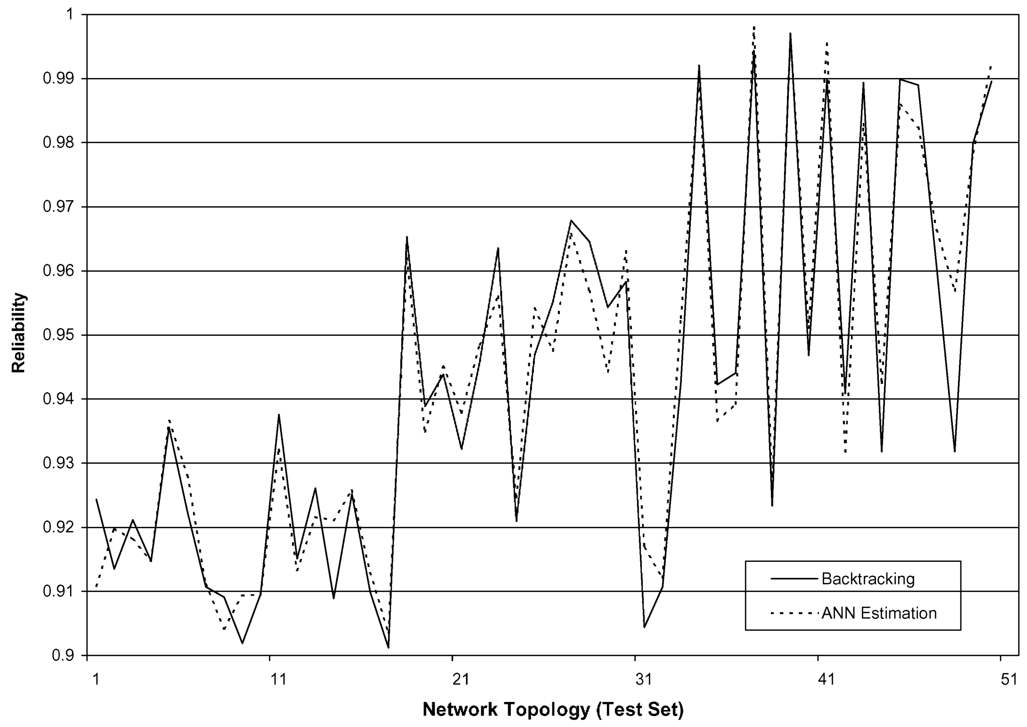

Fig. 3. Specialized neural network estimation of reliability versus actual reliability on the first test fold for networks with identical link reliability.

Table 3
Statistical comparisons on ANN test sets

Network type		Comparison	Difference	p value	Groupings
		ANN vs. Actual	– 0.0041	0.0183	
		Bound vs. Actual	– 0.0515	0.0000	
	General	ANN, Bound, Actual		0.0000	(ANN, Actual) (Bound)
Identical links		ANN vs. Actual	– 0.0012	0.0527	
		Bound vs. Actual	– 0.0082	0.0000	
	Specialized	ANN, Bound, Actual		0.0019	(ANN, Actual) (Bound)
		ANN vs. Actual	– 0.0054	0.0195	
		Bound vs. Actual	– 0.0511	0.0000	
	General	ANN, Bound, Actual		0.0000	(ANN, Actual) (Bound)
Varying links		ANN vs. Actual	– 0.00027	0.6682	
		Bound vs. Actual	– 0.00746	0.0000	
	Specialized	ANN, Bound, Actual		0.0032	(ANN, Actual) (Bound)

Table 4
Errors for the general neural network with varying link reliability

Fold	RMSE training	RMSE testing	RMSE upperbound
Set 1	0.04562	0.05869	0.08880
Set 2	0.04332	0.07042	0.10325
Set 3	0.04813	0.04758	0.06856
Set 4	0.04562	0.06454	0.07523
Set 5	0.04249	0.07177	0.09908
Average	0.04504	0.06260	0.08698

Table 5
Errors for the specialized neural network with varying link reliability

Fold	RMSE training	RMSE testing	RMSE upperbound
Set 1	0.00817	0.00823	0.01031
Set 2	0.00726	0.01211	0.01376
Set 3	0.00781	0.00927	0.01401
Set 4	0.00792	0.00902	0.01127
Set 5	0.00763	0.01037	0.01451
Average	0.00776	0.00980	0.01277

reliability design problem, an optimization procedure that examined only a very small fraction of possible designs would still require millions of network reliability calculations. The tiny number of network topologies needed for ANN training and validation gives an indication of the power of the method. Increasing the training and validation set size will almost certainly improve the estimation accuracy of the ANN while a decrease in size will worsen the estimates. For networks of larger size, where target (actual) reliabilities using backtracking or another exact method are not practical, Monte Carlo simulation could be substituted. Network reliability could be accurately estimated by using many replications of Monte Carlo simulation for each network topology in the data set available for training/testing. While this is still computationally burdensome, it is feasible, and need only be done for the relatively small training/testing data set.

5. Using the ANN estimate during optimal network design

To further examine the utility of the ANN approach for optimal network design, the three methods of reliability estimation — ANN, the upperbound of Konak and Smith [25] and the exact

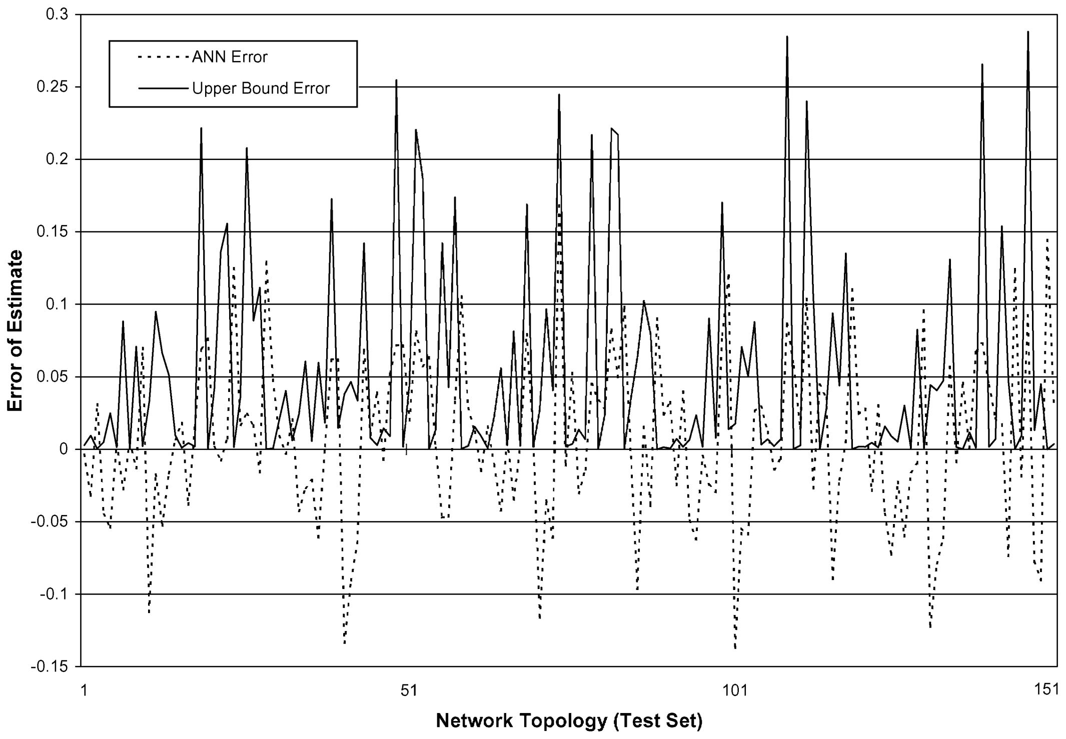

Fig. 4. General neural network estimation error versus upperbound error on the first test fold for networks with varying link reliability.

backtracking calculation — were compared. A straightforward simulated annealing optimization method was developed as described below and was tested on several design problems using the three methods. Two alternative objective functions were used; the first minimized cost subject to a minimum network reliability constraint and the second maximized network reliability subject to a maximum cost constraint. Networks of both identically reliable links and links of differing reliability were considered. While simulated annealing was chosen, other optimization methods such as tabu search or greedy heuristics might be employed.

5.1. Simulated annealing method

Simulated annealing (SA) is a straightforward global optimization method typically used for combinatorial problems. The basic idea is to iterate from a random solution to a near-optimal solution by making generally improving moves while allowing occasional non-improving moves. The method was inspired by the physical phenomenon of cooling metals where atoms move from a random state to a maximally organized state. The main parameter is the cooling schedule, including an initial temperature, a method for reducing temperature, a method for controlling the duration at a given temperature and a

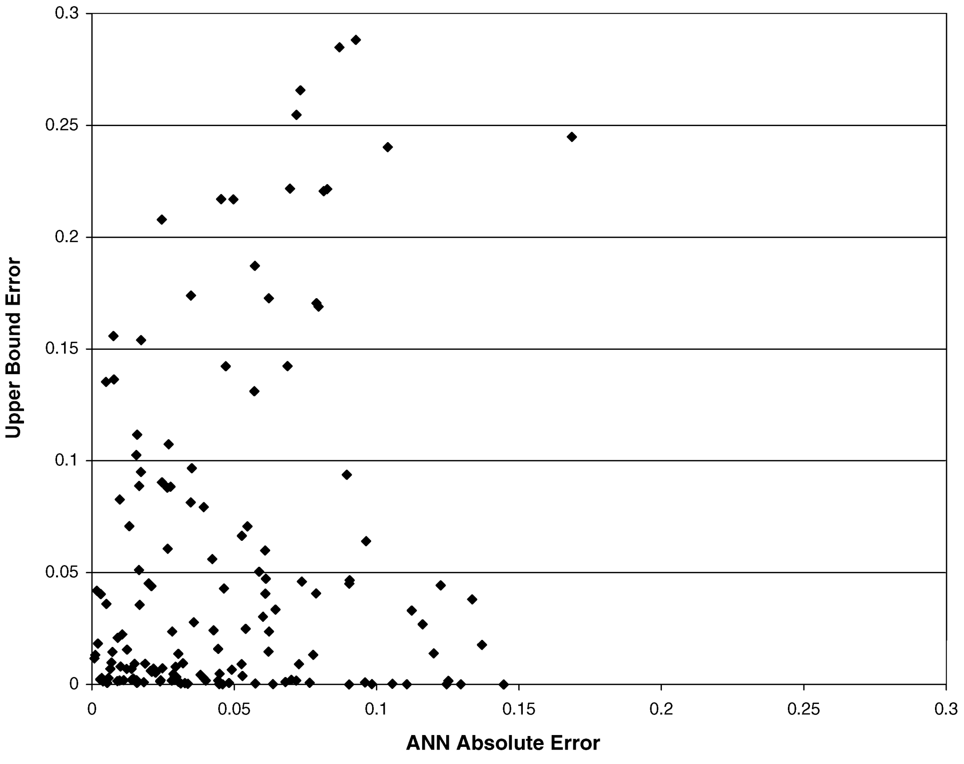

Fig. 5. General neural network absolute estimation error (x-axis) versus upperbound error (y-axis) on the first test fold for networks with varying link reliability.

termination criteria. Solutions are altered via a move operator, which usually transforms a solution to a neighboring solution randomly.

Notation

i	iteration counter
ii	maximum iterations
$S(i)$	solution at iteration i
r	cooling rate
$t(i)$	temperature at iteration i
$U(0, 1)$	uniform random number between 0 and 1
$\exp(-\Delta/t(i))$	acceptance probability of a non-improving move at iteration i.

For the first objective of minimizing cost for a given reliability constraint, the SA procedure works as follows:

Step 1. $i = 0$; $t(i) =$ Initial temperature;
Step 2. Generate initial feasible solution, $S(i)$
Step 3. $S(i+1) =$ Move $(S(i))$ and $\Delta = [S(i+1)\text{cost} - S(i)\text{cost}]$

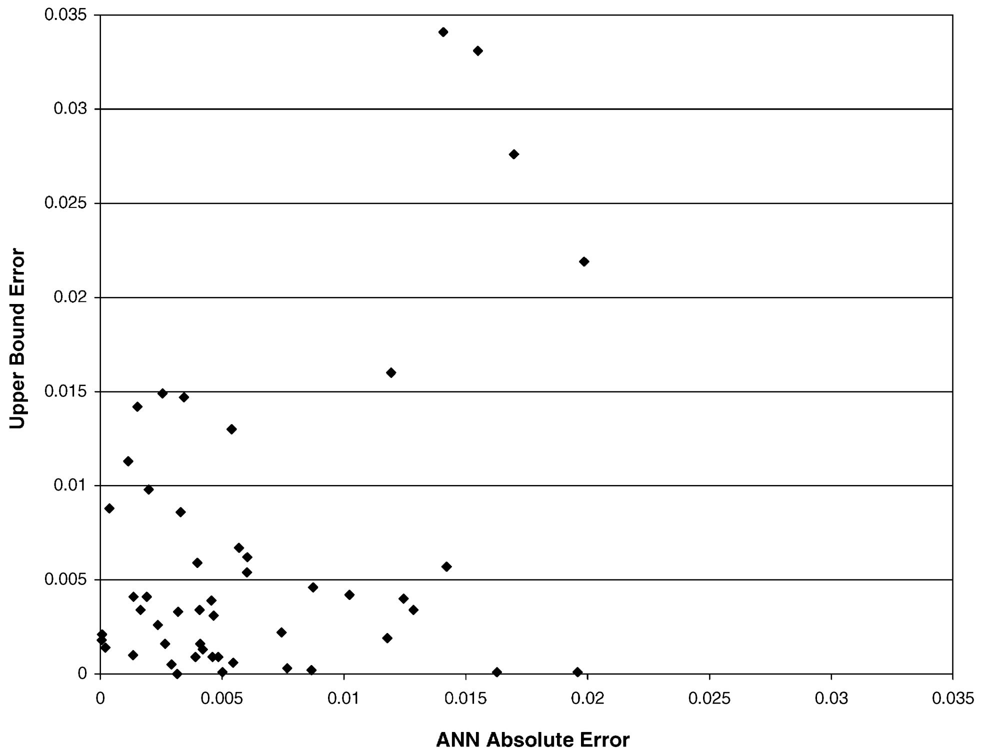

Fig. 6. Specialized neural network absolute estimation error (x-axis) versus upperbound error (y-axis) on the first test fold for networks with varying link reliability.

```
Step 4. If $\Delta \leqslant 0$, go to Step 6, else go to Step 5
Step 5. If $U(0,1) \leqslant \exp(-\Delta/t(i))$, then go to Step 6, else $S(i+1) = S(i)$
Step 6. $t(i+1) = t(i) \times r$
Step 7. If $i \leqslant ii$, then $i = i + 1$ and go to Step 3, else stop.
```

For the second objective of maximizing reliability for a given cost constraint, the search proceeds identically, except in Step 3 Δ is calculated as a reliability difference rather than a cost difference and the sign of the difference is reversed.

The move operator for uniform links begins by randomly choosing four nodes in the network, i_1, i_2, i_3, i_4, and considers the two links: (i_1, i_2) and (i_3, i_4). If one of these two links exists, but the other one does not, the existing one is deleted and the non-existing one is added. If both links exist, then one is randomly chosen and deleted. Finally, if both do not exist, then one is randomly added. Only moves to feasible solutions are taken; if the move operator generates an infeasible solution it is discarded and another move tried.

For networks with differing links, the move operator is somewhat changed. The four nodes are randomly chosen and the two links are considered as before. Upgrade link (i_1, i_2)'s type to the next available link type option (i.e., if it is type i, make it type $i+1$) and degrade (i_3, i_4)'s type to the next available link type option (i.e., if it is type i, make it type $i-1$). Recall that links are ordered from least reliable (not present) to most reliable. Once again, only moves to feasible solutions are taken.

Table 6
Link types for optimization problems

Link type	Link cost	Link reliability
1	100	0.80
2	120	0.85
3	135	0.90
4	155	0.95
5	175	0.99

5.2. Computational results

Several 10 node problems were solved using the ANN discussed in the earlier sections using the link reliabilities and costs listed in Table 6. The SA was run for 10 iterations using a different random number seed with a maximum of 1000 iterations. A geometric cooling schedule was used where the temperature is reduced by a small fraction each iteration. For each problem instance and random number seed, three alternative methods of reliability estimation were considered. The first was the upper bound, the second the ANN (which first requires calculation of the bound as an input to the ANN) and the third, an exact calculation by backtracking. This last method is computationally unwieldy for networks of even 10 nodes.

Fig. 7 shows the results for maximizing reliability with uniform links of reliability equal to 0.90 and a cost constraint of 1800 while Fig. 8 shows the results for maximizing reliability with links of mixed reliability and a cost constraint of 1500. While there is variability to random number seed, the ANN is close to the backtracking solution in at least a few of the replications. The bound performs worse in nearly all replications. Fig. 9 shows results of mixed link networks for the alternative objective of minimizing cost subject to a minimum network reliability of 0.95. It can be seen that the upper bound resulted in infeasible networks (failed to meet the minimum reliability constraint) in all replications while the network designs identified using ANN were feasible except for one instance. Due to the size of the search space, it is impossible to ascertain how close to optimal (minimum cost) these networks are.

These three figures are a typical sampling of the results; the ANN approach was usually superior to the upper bound approach, rarely worse than the upper bound and generally resulted in feasible network designs. While the ANN performance is inferior in quality to that of backtracking, it is not computationally practical to use an exact calculation during iterative design optimization. Summarizing the computational comparison of the upper bound versus the ANN, the ANN takes roughly 1.25 times that of the upper bound alone. This is a modest and manageable increase considering the speed of the procedure.[1]

[1] As an example, one SA replication (1000 iterations) of maximizing reliability given a cost constraint for networks of identical links takes about 4 CPU seconds with the ANN approach.

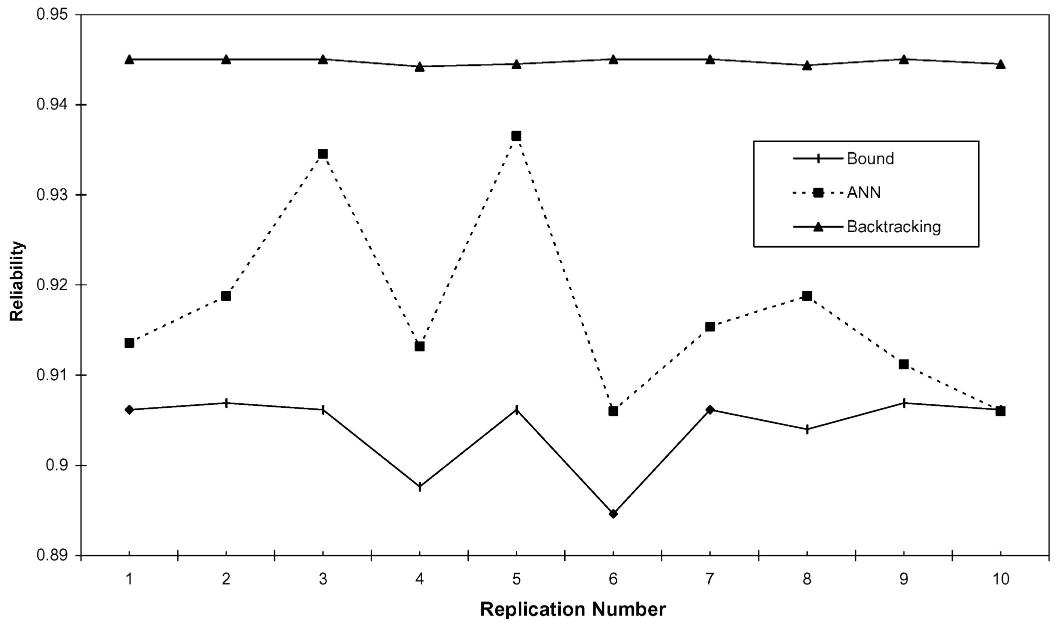

Fig. 7. Results of maximizing reliability subject to a maximum cost constraint of 1800 using links with reliability of 0.90. This graph shows the actual reliability (calculated with backtracking) of the best solution found for each method.

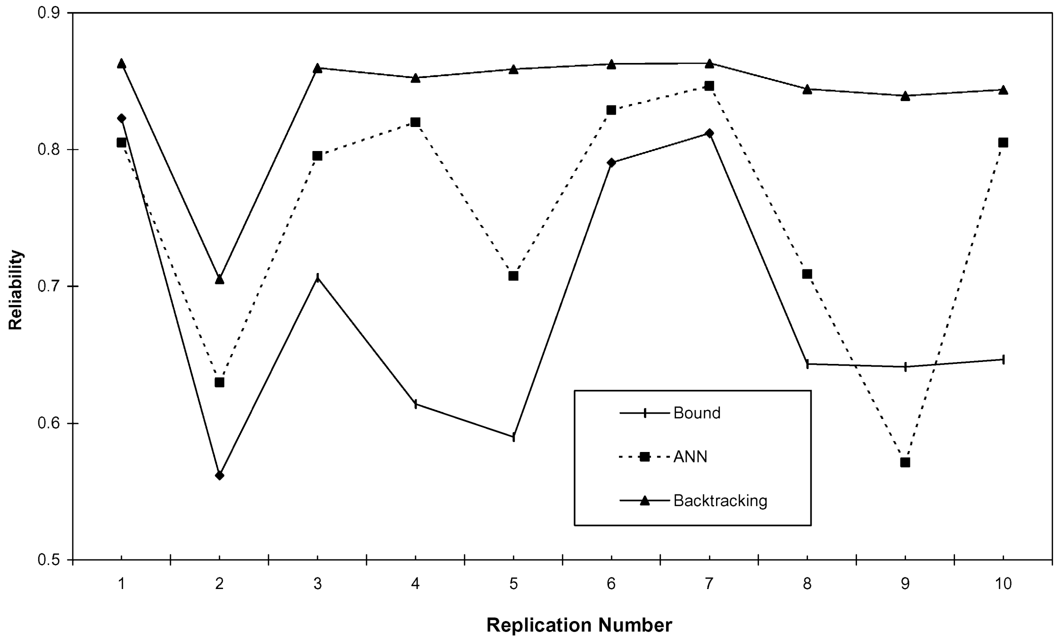

Fig. 8. Results of maximizing reliability subject to a maximum cost constraint of 1500 using mixed links. This graph shows the actual reliability (calculated with backtracking) of the best solution found for each method.

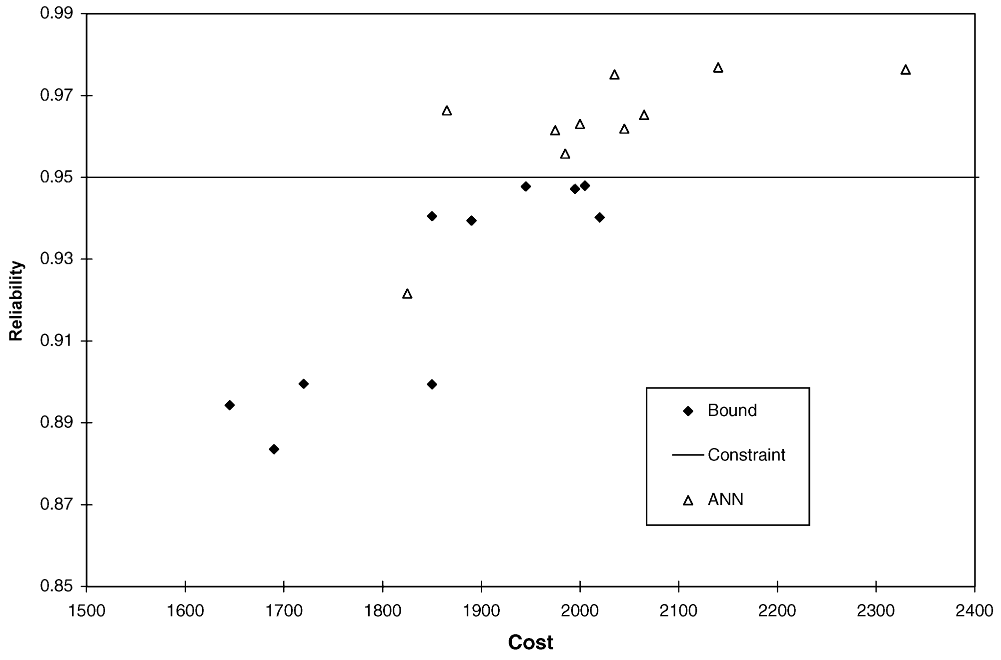

Fig. 9. Actual reliability (calculated by backtracking) and cost of best networks found by using the ANN and the upperbound during minimizing cost subject to a minimum network reliability of 0.95 using links of differing cost and reliability.

6. Conclusions and discussion

The ANN approach to estimating all-terminal reliability worked well. Using an extremely small fraction of the possible network topologies for a 10 node problem, a general ANN and a specialized ANN were trained and validated. While the computational effort of training and validating the ANN is orders of magnitude higher than a single backtracking calculation or a single set of Monte Carlo simulation iterations, subsequent use of the ANN during network design optimization is nearly computationally "free". Backtracking or Monte Carlo simulation must be repeated for each and every network architecture considered during design optimization, often running to the millions of designs for a single network design problem. For networks of larger size, as is typically found in telecommunications, the ANN approach yields greater computational benefits as the increase in computational effort with increase in network size is far less for ANN than for either backtracking or Monte Carlo simulation.

The recommended approach is to use the ANN estimation during the optimization for all topologies considered and then exactly calculate[2] the network reliability on only the best topology, or the few best topologies, periodically during search. In this way, most of the computational effort of reliability calculation is eliminated while maintaining a workable design optimization method.

[2] Or, alternatively, estimate it accurately with Monte Carlo simulation.

The ANN software package used, like others, allows generation of C code from a trained ANN, thereby making this approach practical to embed within general optimization or design algorithms, as was done with the SA design optimization in Section 5.

It is likely that confining the ANN for networks with identical link reliability to only a single link reliability would further improve its precision, however this would reduce flexibility during the design phase. Another alteration is to not randomly generate the network topologies for training and validation, but use a design of experiments to obtain a balance of topologies with different number of links, node degrees, reliabilities, etc. As a further extension, this methodology may work using function approximation methods other than neural networks, such as regression or spline fitting. Finally, this approach of substituting a computationally expedient approximation for the exact objective function calculation in iterative optimization can be feasible and effective for many problems. The important issues are to develop an approximation that is precise enough, especially in the search space regions of greatest interest, and saves sufficient computational effort to make the sacrifice of the exact objective function calculation worthwhile.

References

[1] Jan R-H, Hwang F-J, Chen S-T. Topological optimization of a communication network subject to a reliability constraint. IEEE Transactions on Reliability 1993;42:63–70.
[2] Aggarwal KK, Chopra YC, Bajwa JS. Topological layout of links for optimising the overall reliability in a computer communication system. Microelectronics and Reliability 1982;22:347–51.
[3] Chopra YC, Sohi BS, Tiwari RK, Aggarwal KK. Network topology for maximizing the terminal reliability in a computer communication network. Microelectronics & Reliability 1984;24:911–3.
[4] Deeter DL, Smith AE. Heuristic optimization of network design considering all-terminal reliability. Proceedings of the Reliability and Maintainability Symposium, 1997. p. 194–9.
[5] Deeter DL, Smith AE. Economic design of reliable networks. IIE Transactions 1998;30:1161–74.
[6] Dengiz B, Altiparmak F, Smith AE. Efficient optimization of all-terminal reliable networks using an evolutionary approach. IEEE Transactions on Reliability 1997;46:18–26.
[7] Dengiz B, Altiparmak F, Smith AE. Local search genetic algorithm for optimal design of reliable networks. IEEE Transactions on Evolutionary Computation 1997;1:179–88.
[8] Venetsanopoulos AN, Singh I. Topological optimization of communication networks subject to reliability constraints. Problem of Control and Information Theory 1986;15:63–78.
[9] Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP-completeness. San Francisco: W. H. Freeman and Co, 1979.
[10] Ball M, Van Slyke RM. Backtracking algorithms for network reliability analysis. Annals of Discrete Mathematics 1977;1:49–64.
[11] Aggarwal KK, Rai S. Reliability evaluation in computer-communication networks. IEEE Transactions on Reliability 1981;R-30:32–5.
[12] Cavers JK. Cutset manipulations for communication network reliability estimation. IEEE Transactions on Communications 1975;Com-23:569–75.
[13] Rai S. A cutset approach to reliability evaluation in communication networks. IEEE Transactions on Reliability 1982;R-31:428–31.
[14] Provan JS, Ball MO. The complexity of counting cuts and of computing the probability that a graph is connected. SIAM Journal of Computing 1983;12:777–88.
[15] Fishman GS. A Monte Carlo sampling plan for estimating network reliability. Operations Research 1986;34:581–94.
[16] Yeh M-S, Lin J-S, Yeh W-C. A new Monte Carlo method for estimating network reliability. Proceedings of the 16th International Conference on Computers & Industrial Engineering, 1994. p. 723–6.

[17] Cheng B, Titterington DM. Neural networks: a review from a statistical perspective. Statistical Science 1994;9:2–54.
[18] Geman S, Bienenstock E, Doursat R. Neural networks and the bias/variance dilemma. Neural Computation 1992;4:1–58.
[19] Funahashi K. On the approximate realization of continuous mappings by neural networks. Neural Networks 1989;2:183–92.
[20] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks 1989;2:359–66.
[21] White H. Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings. Neural Networks 1990;3:535–49.
[22] Coit DW, Smith AE. Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach. Computers & Operations Research 1996;23:515–26.
[23] Werbos PJ. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Unpublished PhD thesis, Harvard University, 1974.
[24] Neuralworks reference guide and software. NeuralWare, Pittsburgh, PA, various years.
[25] Konak A, Smith AE. A general upperbound for all-terminal network reliability and its uses. Proceedings of the Industrial Engineering Research Conference, Banff, Canada, May 1998, CD Rom format.
[26] Jan R-H. Design of reliable networks. Computers & Operations Research 1993;20:25–34.
[27] Twomey JM, Smith AE. Bias and variance of validation methods for function approximation neural networks under conditions of sparse data. IEEE Transactions on Systems, Man, and Cybernetics, Part C 1998;28:417–30.

Chat Srivaree-ratana has a B.E. in Mechanical Engineering from Chulalongkorn University, Thailand and an M.S. in Industrial Engineering from the University of Pittsburgh.

Abdullah Konak is a Ph.D. candidate in the Department of Industrial Engineering at the University of Pittsburgh and works as an instructor in the Department of Industrial and System Engineering at Auburn University. He received his M.S. in Industrial Engineering from Bradley University and B.Sc. from Yildiz Technical University, Turkey. His research interests include network optimization, heuristic optimization techniques, network reliability, and manufacturing system modeling using artificial intelligence techniques.

Alice E. Smith is Professor and Chair of Industrial and Systems Engineering at Auburn University. She has degrees in Engineering and Business from Rice University, Saint Louis University and University of Missouri - Rolla. Dr. Smith has authored articles in *IIE Transactions*, *IEEE Transactions on Reliability*, *INFORMS Journal on Computing*, *International Journal of Production Research*, *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *The Engineering Economist*, *and IEEE Transactions on Evolutionary Computation*.