Neural Network Approach for Solving the Train Formation Problem

DAVID R. MARTINELLI AND HUALIANG TENG

The train formation plan is one of the important elements of railroad system operations. Whereas mathematical programming formulations and algorithms are available for solving the train formation problem, the long CPU time required for convergence makes it difficult to solve the problems in a reasonably short time. At the same time, shorter decision intervals are becoming necessary, given the highly competitive operating climate of the railroad industry. A novel approach is presented for quickly obtaining good solutions to the train formation problem (TFP). A neural network model is developed for efficiently solving the TFP. Following a training process for neural network development, a testing process indicates that the neural network model will likely be both sufficiently fast and accurate in producing train formation plans under on-line conditions.

Railroad system operation plans are developed to perform the sequential process of car block decisions, train formation decisions, train schedule decisions, and empty car distribution decisions. This is performed under the consideration of engine power, maintenance, and service level requirements. There are two classes of models developed for railroad operation decisions: those that treat each decision level independently and those that treat two or more of the decision levels simultaneously. For example, Beckmann et al. (*1*), Thomet (*2*), Suzuki (*3*), Duvalyan (*4*), and Assad (*5*) developed models specifically for train formation decisions. On the other hand, Crainic (*6*) developed a model of both car block and train formation decisions, while Keaton (*7–9*) gives three models, each involving multiple decision levels.

Despite the substantial quantity and diversity of rail operating decision models, a common element exists among them, in that they all require a substantial investment of computational effort and subsequent implementation time. Experience with these models indicates that the CPU time required to obtain an optimal or near optimal solution varies with demand patterns and formulation constraints. For example, a longer CPU time is required for Assad's model for more restrictive demand patterns or if additional constraints are considered. Therefore, in general, more dynamic demand patterns tend to require more expensive modeling efforts.

A common approach for the industry in handling dynamic demands has been to shorten the time period between successive modeling updates. Unfortunately this introduces a trade-off between longer CPU time requirements for more realistic solutions and the added resources necessary to provide more frequent model updates. In light of this trade-off, artificial neural networks (ANNs) may prove quite fruitful if shorter implementation times can be achieved without substantial loss in solution integrity. Such an ANN model can also be integrated into a simulation model to analyze more operation alternatives. It is well known that the time-consuming validation and calibration process required by the neural network model is offset by the efficiency of its subsequent use, so this study will concentrate on the integrity of the solution provided by the neural network model.

Department of Civil Engineering, West Virginia University, P.O. Box 6103, Morgantown, W.Va. 26506-6103.

The remainder of this paper is organized as follows: first, a brief introduction to neural network tools and their use in formulating the train formation problem (TFP) is given, with consideration of alternative neural network architectures and problem representations. Next, an analysis is performed for training (or calibrating) the network given solutions from the conventional model of Assad (*5*). The ANN model is tested with respect to various criteria using additional data sets. Finally, several conclusions are derived.

ANN FORMULATION OF THE TFP

In the formulation, the train formation decisions are represented as the output from a neural network model of different topology. The related criteria for neural network model training control and testing are then discussed.

Architecture of the ANN

In this research, the back-propagation neural network (BPNN) model was used due to its wide use in solving the optimization problem. The BPNN model consists of three layers: the input layer, the hidden layer, and the output layer. The neurons in the input layer represent O-D demands, while neurons in the output layer represent the train consist that defines the assignment of demand to trains. There is only one hidden layer considered in this BPNN model.

The model is developed for a hypothetical railroad network having 30 O-D demand pairs, represented in Figure 1. However, this network is realistic in terms of topology and the complexity of the problem implied in this network. The corresponding demands are given in Table 1, where r_{ij} denotes the cars per period demanded from Origin i to Destination j. These demands are indexed 1 through 30 in the respective cells of Table 1. Thus, the number of neurons in the input layer is 30. For this network, 44 trains are considered for satisfying the O-D demands, and an overall 108 combinations of demand-train assignments are considered. The train routes are listed in Table 2, and the train consists considered are listed in Table 3, where $a_{i,j}$ denotes that O-D demand i is assigned to train j ($1 \leq i \leq 30$, $1 \leq j \leq 44$). Thus, the number of neurons in the output layer is 108. Finally, Figure 2 illustrates the structure of the BPNN model.

Of the 30 O-D demands, there are some links that directly connect their origin to their destination. As a normal practice, local train ser-

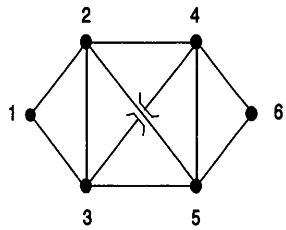

FIGURE 1 Example railroad network.

vices are provided for each link. Thus, these types of demands will be carried to their destination by local trains without passing by any intermediate yards. These demands are called short-distance demands. The remaining demands are those with no direct link between their origin and destination, which therefore must be carried by trains that pass at least 1 intermediate yard. These demands are called long-distance demands. In light of these typical operating characteristics, it is common to assume that short-distance demands will not fluctuate. At this point, it will be assumed that most long-distance demands will fluctuate between 100 and 150 cars per period, while all the short-distance demands are fixed.

TABLE 1 Demand for Example Railroad

O–D	1	2	3	4	5	6
1		r_{12} (1)	r_{13} (2)	r_{14} (3)	r_{15} (4)	r_{16} (5)
2	r_{21} (6)		r_{23} (7)	r_{24} (8)	r_{25} (9)	r_{26} (10)
3	r_{31} (11)	r_{32} (12)		r_{34} (13)	r_{35} (14)	r_{36} (15)
4	r_{41} (16)	r_{42} (17)	r_{43} (18)		r_{45} (19)	r_{46} (20)
5	r_{51} (21)	r_{52} (22)	r_{53} (23)	r_{54} (24)		r_{56} (25)
6	r_{61} (26)	r_{62} (27)	r_{63} (28)	r_{64} (29)	r_{65} (30)	

TABLE 2 Train Route for Supposed Demand

Yard	Train Number	Train Route	Yard	Train Number	Train Route
1	1 2 3 4 5 6 7 8 9 10	(1,2) (1,3) (1,2,4) (1,3,4) (1,2,5) (1,3,5) (1,2,4,6) (1,2,5,6) (1,3,4,6) (1,3,5,6)	4	23 24 25 26 27 28	(4,2,1) (4,3,1) (4,2) (4,3) (4,5) (4,6)
2	11 12 13 14 15 16	(2,1) (2,3) (2,4) (2,5) (2,4,6) (2,5,6)	5	29 30 31 32 33 34	(5,2,1) (5,3,1) (5,2) (5,3) (5,4) (5,6)
3	17 18 19 20 21 22	(3,1) (3,2) (3,4) (3,5) (3,4,6) (3,5,6)	6	35 36 37 38 39 40 41 42 43 44	(6,4,2,1) (6,4,3,1) (6,5,2,1) (6,5,3,1) (6,4,2) (6,5,2) (6,4,3) (6,5,3) (6,4) (6,5)

TABLE 3 Resulting Train Consist from BPNN

Yard	Train Number	Train	Train Consist Considered	Desired Consist	Recalled Consist
1	1	T12	$a_{5,1}$, $a_{4,1}$, $a_{3,1}$, $a_{1,1}$	64 20 73 2	64 17 72 2
	2	T13	$a_{5,2}$ $a_{4,2}$, $a_{3,2}$, $a_{2,2}$	94 91 60 0	94 91 60 0
	3	T114	$a_{5,3}$, $a_{4,3}$	10 0	10 0
	4	T214	$a_{5,4}$, $a_{4,4}$	0 0	0 0
	5	T115	$a_{5,5}$, $a_{4,5}$	0 0	0 0
	6	T215	$a_{5,6}$, $a_{4,6}$	17 0	17 0
	7	T116	$a_{5,7}$	0	0
	8	T216	$a_{5,8}$	148	148
	9	T316	$a_{5,9}$	0	0
	10	T416	$a_{5,10}$	0	0
2	11	T21	$a_{26,11}$, $a_{21,11}$, $a_{16,11}$, $a_{6,11}$	78 8 136 1	78 5 135 1
	12	T23	$a_{7,12}$	87	87
	13	T24	$a_{26,13}$, $a_{4,13}$, $a_{10,13}$, $a_{8,13}$	20 0 27 107	20 0 27 107
	14	T25	$a_{26,14}$, $a_{4,14}$, $a_{10,14}$, $a_{9,13}$	73 2 54 0	72 2 54 0
	15	T126	$a_{26,15}$, $a_{10,15}$	0 0	0 0
	16	T226	$a_{26,16}$, $a_{10,16}$	0 0	0 0
3	17	T31	$a_{26,17}$, $a_{21,17}$, $a_{16,17}$, $a_{11,17}$	72 13 0 0	72 13 0 0
	18	T32	$a_{12,18}$	95	95
	19	T34	$a_{5,19}$, $a_{15,19}$, $a_{4,19}$, $a_{13,19}$	91 0 4 10	91 0 4 10
	20	T35	$a_{5,20}$, $a_{4,20}$, $a_{15,20}$, $a_{15,20}$	60 0 14 104	60 0 14 104
	21	T136	$a_{5,21}$, $a_{15,21}$	14 0	14 0
	22	T236	$a_{5,22}$, $a_{15,22}$	22 0	22 0
4	23	T141	$a_{26,23}$, $a_{16,23}$	36 11	36 11
	24	T241	$a_{26,24}$, $a_{16,24}$	93 0	93 0
	25	T42	$a_{26,25}$, $a_{16,25}$, $a_{27,25}$, $a_{17,25}$	8 61 0 9	5 61 0 9
	26	T43	$a_{26,26}$, $a_{16,27}$, $a_{28,26}$, $a_{18,26}$	13 19 0 46	13 19 0 47
	27	T45	$a_{19,27}$	10	10
	28	T46	$a_{5,28}$, $a_{10,28}$, $a_{15,28}$, $a_{20,28}$	0 107 10 34	0 107 10 34
5	29	T151	$a_{26,29}$, $a_{21,29}$	0 0	0 0
	30	T251	$a_{26,30}$, $a_{21,30}$	0 0	0 0
	31	T52	$a_{26,31}$, $a_{21,31}$, $a_{27,31}$, $a_{22,31}$	136 38 0 64	135 38 12 64
	32	T53	$a_{26,32}$, $a_{21,32}$, $a_{28,32}$, $a_{23,32}$	0 89 0 0	0 89 0 0
	33	T54	$a_{24,33}$	82	82
	34	T56	$a_{5,34}$, $a_{15,34}$, $a_{10,34}$, $a_{25,34}$	2 0 104 99	2 0 104 99
6	35	T161	$a_{26,35}$	148	148
	36	T261	$a_{26,36}$	0	0
	37	T361	$a_{26,37}$	0	0
	38	T461	$a_{26,38}$	0	0
	39	T162	$a_{26,39}$, $a_{27,39}$	12 0	0
	40	T261	$a_{26,40}$, $a_{27,40}$	65 1	65 1
	41	T163	$a_{26,41}$, $a_{28,41}$	83 0	83 0
	42	T263	$a_{26,42}$, $a_{28,42}$	11 0	11 0
	43	T64	$a_{26,43}$, $a_{27,43}$, $a_{28,43}$, $a_{29,43}$	1 9 46 67	1 9 47 67
	44	T65	$a_{26,44}$, $a_{27,44}$, $a_{28,44}$, $a_{30,44}$	0 64 0 26	0 64 0 26

Determination of Training and Testing Criteria

The development of the BPNN model requires training of the network through numerous prepared training patterns. During training, weights associated with each link between neurons are adjusted in a manner that yields minimum error between the network output and the output of the training data set. This allows the network to "recognize" the underlying relationship between the input parameters (demand) and the output (train assignments). Following the training process, the adjusted connections between the input and output neurons can be used to generate train formation plans (train-demand assignments) for data sets beyond those used for training.

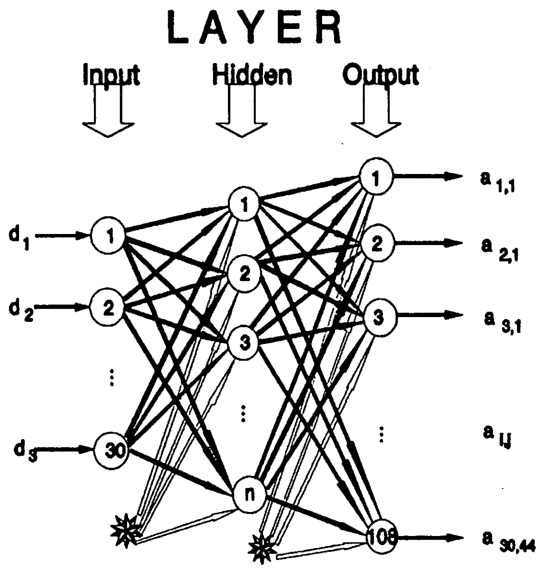

FIGURE 2 BPNN model structure.

The trained BPNN model will yield a TFP solution for a given data set almost instantaneously.

In the course of the training process, data sets are generated by a conventional model (*5*) in which a Monte Carlo algorithm was employed for solution. As is common with BPNN development, the training process was guided by the sum of square error (SSE) criterion, defined as follows:

$$\mathrm{SSE} = \left[\sum_{i=1}^{30}\sum_{j=1}^{44}(a'_{i,j} - a_{i,j})^2\right]^{1/2} \tag{1}$$

where $a'_{i,j}$ and $a_{i,j}$ denote the predicted and desired train assignments, respectively.

Following the training process, the resulting network is tested according to a set of criteria beyond that of simple calculation of output errors. Specifically, these more comprehensive criteria are used to investigate whether (*a*) the BPNN abides by the constraints and the objective function of the conventional model, and (*b*) the train-demand assignments produced from the BPNN model are adequately similar to those of the conventional model.

Three sets of constraints in the conventional model assure traffic flow balance, flow conservation, and minimal service flows. If the BPNN models are assumed to have recognized the flow balance constraints, the volume in the predicted demand-train assignment should be the same or close to those presented to the model. Thus, the first criterion, Average Percent Flow Deviation (APFD), measures the difference between the predicted and desired demand volumes. This criterion is explicitly defined as follows:

$$\mathrm{APFD} = \frac{1}{30M}\sum_{m=1}^{M}\sum_{i=1}^{30}\frac{\left|\sum_{j\subset O_{io}} a'_{ij} - \sum_{j\subset O_{io}} a_{ij}\right|}{\sum_{j\subset O_{io}} a_{ij}} \tag{2}$$

where M is the number of testing patterns, 30 is the number of O-D demands, and O_{io} is the set of trains going out of the origin of demand i.

In the same way, if the BPNN models are assumed to successfully recognize the flow conservation constraints, the flows into an intermediate yard should equal the flows out of the yard. Thus the second criterion—Average Flow Conservation at Intermediate Yards (AFCIY)—is explicitly defined as follows:

$$\mathrm{AFCIY} = \frac{1}{10\,MN}\sum_{m=1}^{M}\sum_{i=1}^{10}\left|\sum_{k=1}^{|IM_i|}\sum_{j\subset I_{i,k}} a'_{i,j} - \sum_{j\subset O_{i,k}} a'_{i,j}\right| \tag{3}$$

where $I_{i,k}$ and $O_{i,k}$ denote, respectively, the set of trains available for demand i going into and out of yard k. IM_i denotes the set of yards demand i will pass by, excluding its origin and destination yards. N denotes the number of the yards in IM_i.

In the conventional model, the train service minimum flow limit is that train service may be provided if the flow exceeds the minimum; otherwise no train service will be provided. In general the flows on the train services generated from the BPNN models are not guaranteed to satisfy the minimum flow limits. Thus, the extent to which the BPNN model satisfies this constraint must be investigated. Then the third criterion—average train length (ATL)—is defined as follows:

$$\mathrm{ATL} = \frac{1}{M\,|T_m|}\sum_{m=1}^{M}\sum_{j\subset T_m}\sum_{i\subset D_j} a'_{i,j} \tag{4}$$

where T_m denotes the set of trains in the mth testing pattern the flows on which are less than the minimal flow limit. D_j denotes the set of O-D demands which are supposed to be assigned to train j.

The aim of developing a BPNN is to obtain a more time-efficient model than the conventional model, while preserving the integrity of the solution. Therefore, it is important to assess the deviation between the BPNN solution and the conventional model. The BPNN may obtain some solutions superior to that of the conventional model. In such cases, conformity to the constraints must be insured. If a superior BPNN solution does not satisfy the problem constraints, then the predicted demand-train assignments must be adjusted to conform to the required constraints. Hence, a method was devised to recognize which train consists fully use the information in the predicted solution. Based on the adjusted demand-train-assignment, the fourth criterion involving the objective function is incorporated into the evaluation, namely the Percent Optimal Solution Difference (POSD):

$$\mathrm{POSD} = \sum_{m=1}^{M}|y'_m - y_m|/y_m \tag{5}$$

where y'_m and y_m denote the predicted and desired optimization solution, respectively.

The common practice of testing the predicted solution versus the desired solution based only on SSE may not suffice in testing the similarity between the predicted and desired train consist. Specifically, SSE is an aggregate measure of deviation, and a more disaggregate criterion is needed. Hence, the remaining two criteria measure first whether the predicted demands are actually assigned to the trains desired, and second, the difference in the number of cars between the BPNN solution and that of the conventional model. The fifth criterion—train consist difference in terms of demand assigned within the train (TCDDA)—is then calculated as follows:

$$\text{TCDDA} = \frac{1}{108 \times M} \sum_{m=1}^{M} \sum_{j=1}^{44} \frac{\sum_{i \subset D_j} |\delta(a'_{i,j}) - \delta(a_{i,j})|}{|D_j|} \quad (6)$$

where δ is a switch function. δ will be 0 when $a'_{i,j}$ or $a_{i,j}$ are less than or equal to 0; δ will be 1 otherwise. The number 108 is the sum of O-D demands considered in all the trains. The sixth criterion—train consist difference in terms of cars assigned (TCDCA)—is calculated as follows:

$$\text{TCDCA} = \frac{1}{108 \times M} \sum_{m=1}^{M} \sum_{j=1}^{44} \frac{\sum_{i \subset D_j} |[\delta(a'_{i,j}) - \delta(a_{i,j})]\,(a'_{i,j} - a_{i,j})|}{\sum_{i \subset D_j} [\delta(a'_{i,j}) - \delta(a_{i,j})]} \quad (7)$$

The difference between criteria stated in Equations 6 and 7 is the addition of $(a'_{i,j} - a_{i,j})$ cars in Equation 6 based on the difference of demand-train assignment.

There are different changing patterns for the six criteria, given a variation in the volume of training patterns. For the criteria given in Equations 2, 3, and 4; the smaller the values obtained, the better the recognition of restriction on flow conservation, balance, and train length for the train service provision. For criteria given in Equation 5, the smaller the values, the better the acknowledgment of objective function. For criteria given in Equations 6 and 7, the smaller the values, the more similar will be the predicted train consists to the desired train consists.

BPNN MODEL TRAINING PROCESS

The BPNN training process consists of two phases: training and testing. First, the BPNN models are trained by sets of training patterns until the control criteria SSE reach a certain value. It is well understood that the characteristics of data used as training patterns have an impact on the effectiveness and efficiency of the training process. Therefore, some of the data characteristics are varied in this BPNN training. Two groups of training patterns were employed. In the first group, a portion of the patterns have long-distance fluctuating demands that take on the boundary values 100 or 150, and a portion have long-distance demands that take on the random values uniformly generated between 100 and 150. In the second group, all of the sets of patterns have long-distance fluctuating demands that take on the random values generated between 100 and 150.

Five sets of training patterns are in the first training group. For each set, the number of patterns are, respectively, 20, 40, 64, 114, and 164. In the sets of 20, 40, and 64 patterns, all the long-distance fluctuating O-D demands are restricted to boundary values only. The sets of 114 and 164 patterns are composed of the set of 64 with an additional 50 and 100 patterns. For these additional patterns, all the long-distance demands take on random values. There are four sets of training patterns in the second group. In all four, the long-distance demands take on the random values. The number of patterns in each set are, respectively, 40, 60, 80, and 100.

Following the training process, the resulting BPNN model is tested by one set of testing patterns so as to investigate the extent to which the model can recognize the relationships imbedded in the training patterns. Generally, neural network models are tested according to the SSE criteria. Different from this practice, however, the train consists predicted by the BPNN models here are tested by the extended set of criteria derived from the constraints in the conventional model.

Training

The following are two examples used in training the BPNN. In the first example, the BPNN model is trained by a boundary set consisting of 64 patterns. In the second example, the BPNN model is trained by a boundary set consisting of 114 patterns.

Training by the Boundary Set of 64 Patterns

The trainings are carried out by setting the number of neurons in the hidden layer to 30, 40, 50, and 60 respectively, and the SSE at 0.5. The reduction of the SSE over the course of the training process is represented by Figure 3. The SSE decreases dramatically as the training iteration increases from 0 to 2,000, whereas after Iteration 2,000 the rate of decrease is significantly less. Table 3 gives the desired and recalled train consists predicted by the resulting BPNN model. A strong similarity may be observed between the desired and the recalled train consists.

Training by the Boundary Set of 114 Patterns

Different from the training of the first example, the training for this second example is conducted in three steps. First, due to the increase in the number of training patterns, a longer time is required to reach the initial SSE. After several trials, the SSE was set at 1.5. Second, for the same reason, the number of hidden nodes was increased to 60, 70, 80, and 90, respectively, for training. Finally, only the BPNN of 90 hidden nodes reached the required SSE, and the CPU time consumed approximately 11 hr. This training process is shown in Figure 4.

Third, to reduce the required CPU time based on the 90-hidden-node model, several learning rate combinations were tried. The results from this experiment are given in the following table, where α_1 and α_2 denote the learning rates for the input and output layers respectively.

Learning Rate α_1, α_2	*Number of Iterations*	*CPU Time*	*CPU Time Each Iteration*
$\alpha_1=1.0$ $\alpha_2=1.0$	4,600	9:07:05.36	7.14
$\alpha_1=1.0$ $\alpha_2=0.8$	6,800	13:41:52.00	7.25
$\alpha_1=0.8$ $\alpha_2=1.0$	3,800	9:40:41.46	9.17

The learning processes are illustrated in Figure 5. Observe in the table just displayed that the BPNN model of $\alpha_1=1$ and $\alpha_2=1$ dominates those of other learning rate combinations. This may demonstrate that the variation of the learning rate may not give an opportunity to quicken the training process. This point of view takes exception to the popular concept that the training process can be accelerated by setting a learning rate for the first layer than for the second layer.

Finally, the CPU times are summarized in Figure 6 for each set of training patterns. For the models trained by boundary sets with 20, 40, and 64 patterns, the CPU times are significantly shorter than those trained by sets of 114 and 164 patterns. After the set of 114 patterns, CPU times increase rapidly as the number of training patterns increase. For the models trained by random sets, the CPU

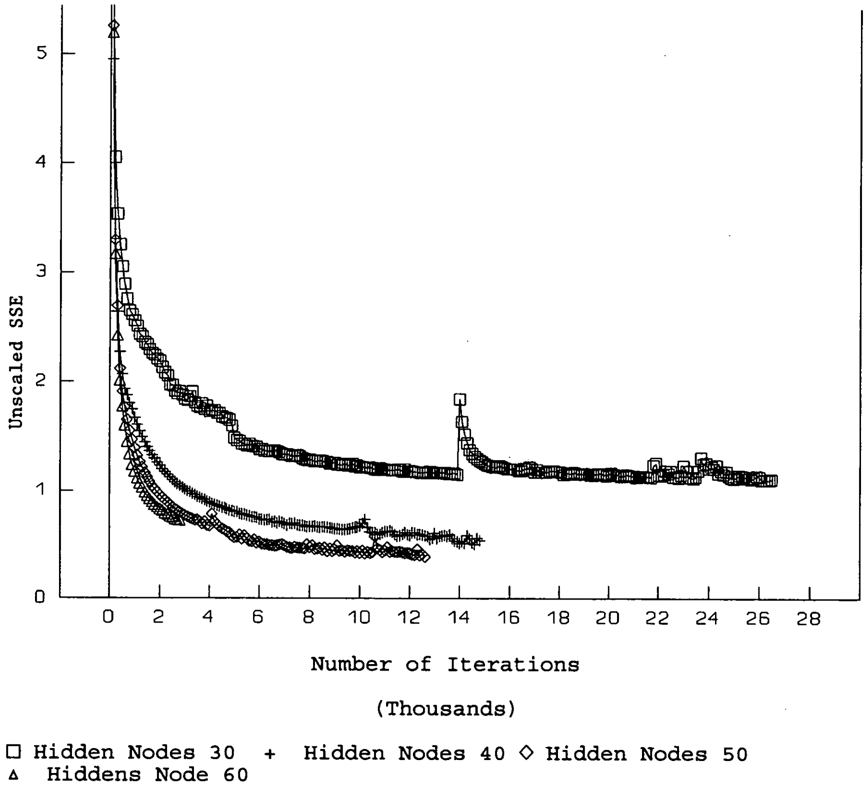

FIGURE 3 Training processes by boundary set of 64 patterns.

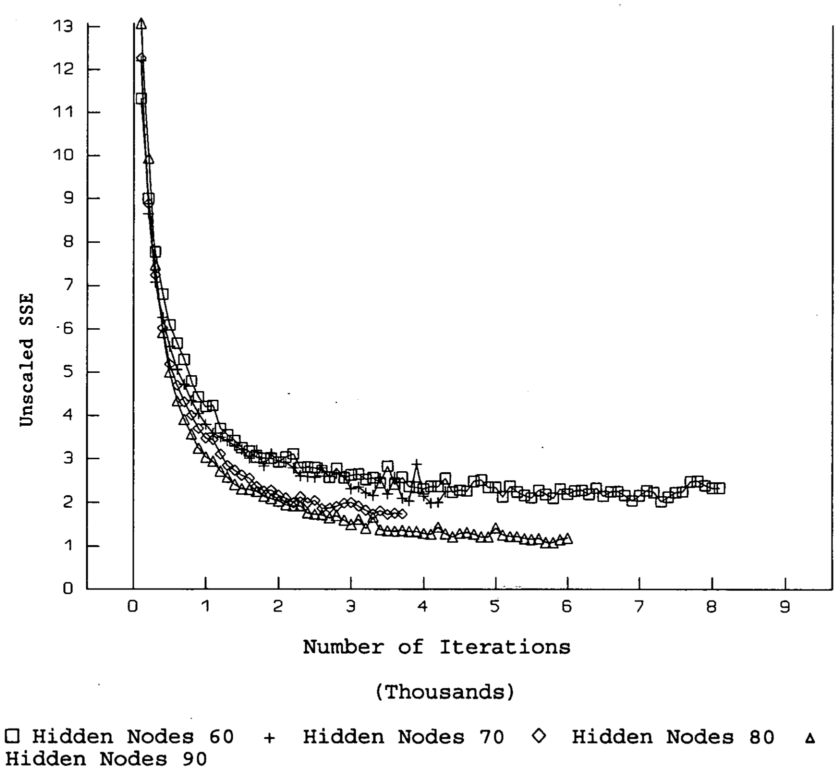

FIGURE 4 Training processes by boundary set of 114 patterns.

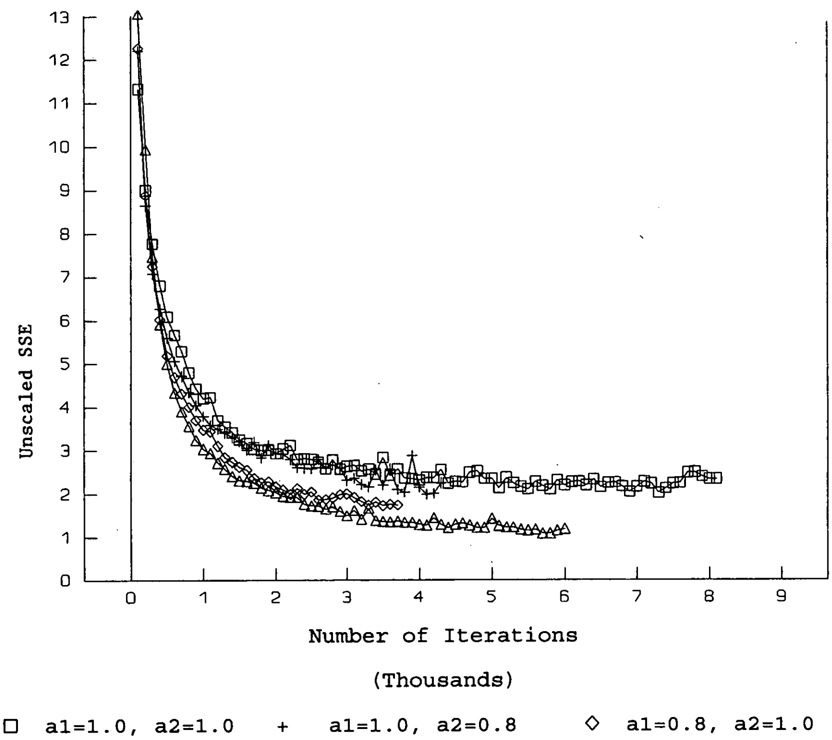

FIGURE 5 Learning processes by varying the learning rates.

times change in the same manner as those trained by boundary sets. Comparing the CPU times consumed by the BPNN models trained by the two different sets of patterns, it is observed that for the same amount of training pattern inputs, the BPNN models trained by random sets consume greater CPU times than those trained by boundary sets. This difference is not large.

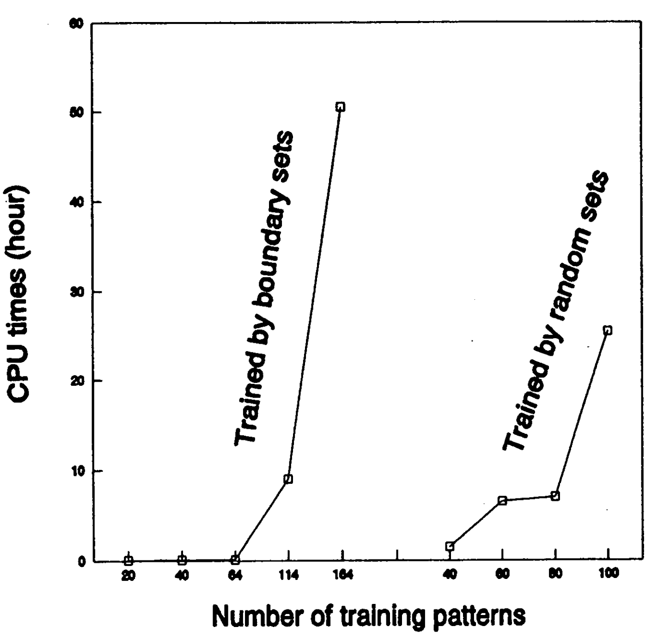

FIGURE 6 CPU times consumed.

Testing

The neural network models were tested according to the six criteria previously stated. The results are given in Tables 4 and 5. The six evaluation criteria are briefly repeated below.

Referring to Table 4, for the BPNN models trained by boundary sets, Criterion 1 achieves a minimal value when the number of patterns is 64. Therefore, inputting more than 64 patterns will not produce a smaller value. For the remaining five criteria, however, smaller values could be obtained as more training patterns are used.

Referring to Table 5 for the BPNN models trained by random sets, there are three different patterns for the six criteria. Criteria 2 and 5 obtain smaller values as more training patterns are input. For Criteria 1 and 4, there is no obvious trend for the variation of the volume of training patterns. Criteria 3 and 6 obtain the smallest values when 60 training patterns were used.

The BPNN models trained by random sets dominate those trained by boundary sets for low volumes of input sets, according to Criteria 1, 3, 4, and 6. The reverse is the case according to Criterion 5, and there is no significant difference according to Criterion 2. However, regardless of the criteria, the two models will converge to the same value as the number of patterns used increases.

Finally, train consists produced by the BPNN models are acceptable, and some are better than those obtained from the conventional model.

To further explain Criterion 3, the train length distributions are compiled for trains predicted to be less than the minimum train length limit. These trains are divided into 10 groups according to their length (number of cars): 0–1, 1–2, 2–3, 3–4, 4–5, 5–6, 6–7, 7–8, 8–9, and 9–10. The train length distribution could be represented by

TABLE 4 Test Results for BPNN Models Trained by Boundary Sets

Patterns\Criteria	1	2	3	4	5	6
20	0.14	3.58	1.89	0.065	0.46	4.68
40	0.14	3.27	2.24	0.072	0.43	4.75
64	0.10	2.41	2.23	0.070	0.43	4.83
114	0.16	1.65	1.54	0.056	0.37	3.75
164	0.20	1.63	1.40	0.048	0.38	3.67

TABLE 5 Test Results for BPNN Models Trained by Random Sets

Patterns\Criteria	1	2	3	4	5	6
40	0.21	2.75	1.39	0.050	0.67	4.37
60	0.21	2.61	1.25	0.040	0.40	3.48
80	0.19	1.73	1.30	0.049	0.36	3.82
100	0.20	1.83	1.42	0.040	0.36	3.90

$$p_k = \frac{t_k}{\sum_{m=1}^{M} |T_m|} \quad (8)$$

where t_k represents the number of trains falling into each group, $k = 1, 2, \ldots, 10$, and p_k represents the percent of trains in each group. The statistics are represented in Figures 7 and 8. Observe that, as the training patterns for both training sets increase, a greater percent of trains are predicted in the interval between 0 and 3. However, trains having less than three cars may be viewed as 0 (i.e., no service provided). In other words, fewer trains are predicted with less than minimum length.

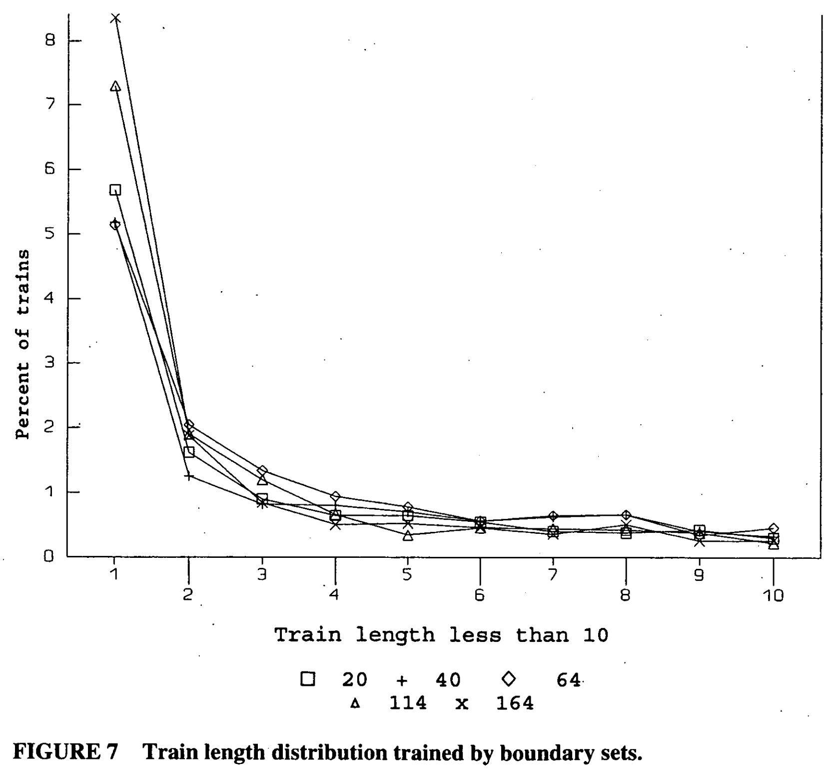

FIGURE 7 Train length distribution trained by boundary sets.

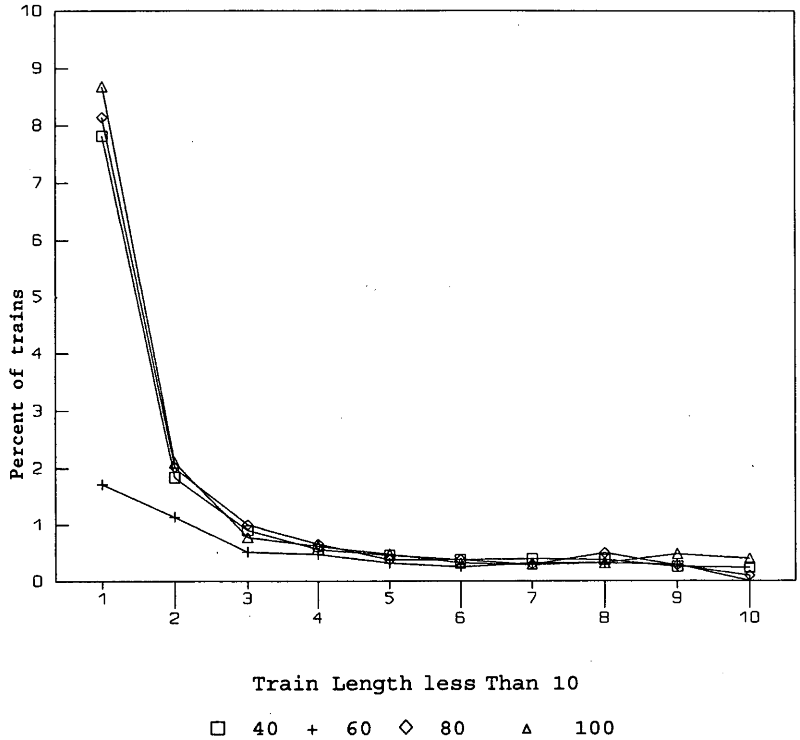

FIGURE 8 Train length distribution trained by random sets.

CONCLUSIONS

On the basis of the analysis presented here, several conclusions can be derived:

1. The neural network model was successful in recognizing the constraints of the conventional model as implied by the training patterns. Hence, to some extent, this recognition ensures that the predicted train consist satisfies the practical requirements and constitutes a feasible solution.
2. The neural network model successfully recognized the objective function of the conventional model. This ensures that the predicted train consists are optimal or near-optimal. In several cases the neural network models produced better solutions than the conventional model.
3. Because more than one possible optimal solution exists, the train consist predicted may be different from those obtained by the conventional model. More similar optimal solutions could be obtained as more training patterns are input (as measured by Criteria 5 and 6). For this reason, the popular practice of testing whether the solutions from the neural network model are similar to those desired using the SSE criteria is not appropriate to this type of optimization problem.
4. The two alternative strategies for generating training patterns had no significant impact on the effectiveness (CPU time) of the solution generated from the neural network model. This observation is reinforced due to the convergence of the two different neural network models as the number of training patterns enlarged.
5. Further research on this topic might be extended in any of several directions. First, to quicken the training process, multi-layer back-propagation neural network models could be tried, and the learning rates assigned could be different for neurons which connect different O-D demand-train assignments. Second, the train consist could be represented in 0-1 strings. This representation might prove easier for training and analysis.

REFERENCES

1. Beckmann, M., C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation.* Yale University Press, New Haven, Conn., 1965.
2. Thomet, M. A. A User-Oriented Freight Railroad Operating Policy. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1 No. 4, Oct. 1971, pp. 349–356.
3. Suzuki, S. A Method of Planning Yard Pass Trains in a General Network. In *Operation Research '72* (M. Ross, ed.). North Holland Publishing Company, Amsterdam, The Netherlands, 1973, pp. 353–361.
4. Duvalyan, S. V. Working Out Optimal Freight Train Make-Up on Electronic Computer and Distribution of Classification Work on Large Railway Network. *Rail International*, October, 1973, pp. 1069–1071.
5. Assad, A. A. Modelling of Rail Networks: Toward a Routing/Makeup Model. *Transportation Research*, Vol. 14B, 1980, pp. 101–114.
6. Crainic, T., J. Ferland, and J. Rousseau. A Tactical Planning Model for Rail Freight Transportation. *Transportation Science*, Vol. 18, No. 2, May, 1984, pp. 165–182.
7. Keaton, M. H. *Optimizing Railroad Operations.* Ph.D. thesis. University of Wisconsin, Madison, 1985.
8. Keaton, M. H. Designing Optimal Railroad Operating Plans: Lagrangian Relaxation and Heuristic Approaches. *Transportation Research*, Vol. 23B, No. 6, 1989, pp. 415–431.
9. Keaton, M. H. Designing Railroad Operation Plans: a Dual Adjustment Method for Implementing Lagrangian Relaxation. *Transportation Science*, Vol. 26, No. 4, November, 1992, pp. 263–279.

Publication of this paper sponsored by Committee on Artificial Intelligence.