Proceedings of the 12th IEEE
International Symposium on Intelligent Control
16-18 July 1997, Istanbul, Turkey

Training Recurrent Neural Networks for Dynamic System Identification Using Parallel Tabu Search Algorithm

D.Karaboga and A. Kalinli
ISYS Research Laboratory, Department of Electronic Engineering,
Engineering Faculty, Erciyes University
38039, Kayseri/ Turkey

Abstract

There are several modern heuristic optimisation techniques such as neural networks, genetic algorithms, simulated annealing and tabu search algorithms. Of these algorithms, tabu search is quite new promising search technique for numeric problems, especially for non-linear problems. However, the converging speed of the standard tabu search to the global optimum is initial solution dependent since it is a form of iterative search. In this paper, a new model of tabu search which has been proposed by the authors to overcome the drawback of a standard tabu search is tested for training a recurrent neural network to identify dynamic systems.

Keywords: Parallel tabu search, recurrent neural networks, genetic algorithms, system identification.

1. Introduction

Tabu search is a form of iterative search and based on intelligent problem solving principles. The modern version of the algorithm was developed by Glover [1, 2]. Tabu search has a flexible memory to keep the information about the past steps of the search and uses it to create and exploit the new solutions in the search space.

Tabu search is quite promising for numeric optimisation problems, especially for non-linear problems due to its features such as working iteratively (good for local converging), getting out of a local minima in the search space (good for the global optimisation) and using a history record of search (good for the expensive evaluations) [3, 4]. Tabu search is good at local converging due to its iterative nature, but it can have problem with reaching the global optimum solution in a reasonable computation time when the initial solution is far away from the region where the global optimum solution exists.

The dependence of the convergence speed of tabu search on the initial solution can be weakened by introducing a parallel structure into the algorithm [5]. The parallelism helps the tabu search find the promising regions of the search space very quickly. In this work, a parallel model proposed by the authors in [6] is tested for training a recurrent neural network for dynamic system identification. Section 2 gives the basic principles of a tabu search algorithm. Section 3 presents the proposed model by the authors. Section 4 explaines the application of tabu search for training recurrent neural networks and compares the performance of the proposed model to those of the standard tabu search and back-propagation algorithms.

2. A standard tabu search

The flowchart of a standard tabu search is presented in Figure 1. In the first unit, a random feasible solution $x_{initial} \in X$ for the problem is generated, and the tabu list and other parameters such as iteration number are initialised. In the second unit, a feasible set of solutions Q* are produced from the present solution x_{now} according to the tabu list and aspiration criteria. The evaluation unit evaluates the each solution x* produced from the present solution x_{now}. After the next solution x_{next} is selected in the selection unit with respect to the evaluation values, the last unit modifies the history record of the search. If the next solution is better than the best solution found so-far x_{best}, the next solution is replaced with the best solution. This loop is repeated until a stopping criteria is satisfied.

0-7803-4116-3 / 97 / $10.00

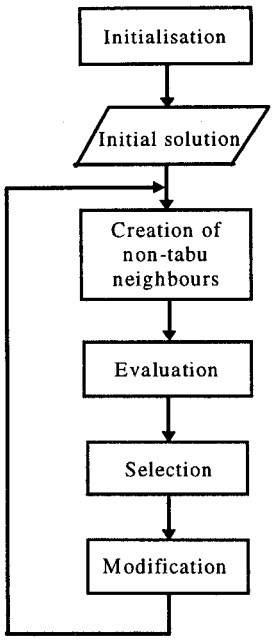

Figure 1. Flowchart of a standard tabu search

Standard tabu search employed in this work uses the following two tabu restrictions to decide whether a move is tabu or not:

recency(x*) >= recency limit (1)

frequency ratio(x*) <= frequency limit

If one of these restrictions is not satisfied, the move x^* is classified as tabu. The recency of a move is the difference between the current iteration number and the last iteration number at which that move has been changed. Frequency measure is equal to the counts of changes of the move.

In this work, an aspiration criteria is introduced when all available moves are classified tabu. The tabu move which loses its tabu status by the least increase in the value of current iteration is freed from the tabu list. The highest evaluation move is selected as the next solution.

3. Proposed model for tabu search

The flowchart of the proposed model is depicted in Figure 2. Sub-blocks 1 to n are each identical to the flowchart shown in Figure 1. These blocks represent n TSs executing independently of one another. The initial solutions for the tabu searches at the first cycle are created using a random number generator with different seeds.

After a given number of iterations, maxit1, the execution of these tabu searches is stopped. maxit1 is normally chosen to be sufficiently large to allow the tabu search to complete the local searching. An operation is applied to the solutions found by n parallel TSs at the first cycle to create the initial solutions for TSs at the second cycle. Various methods could be used for this purpose, for example, the best solution found by the TSs at the first cycle can be always selected as one of the initial solutions, and the others can be produced by applying a suitable crossover operator to the solutions obtained at the first cycle or all the initial solutions can be created using crossover operator. Another procedure could be the following: the best solution is selected directly, some can be created by applying crossover operator to the solutions obtained at the first cycle and the rest can be generated by a random number generator. In this work, the best solution is directly selected and the rest are created by applying the crossover operation explained in the following section.

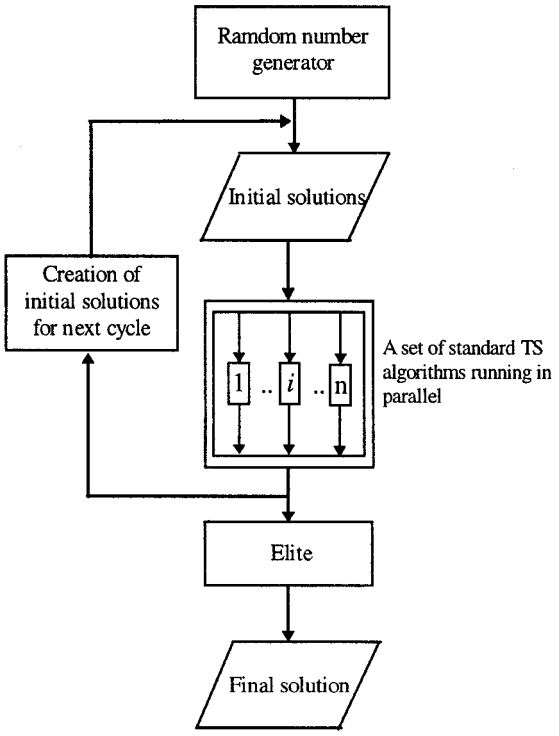

Figure 2. Flowchart of the proposed tabu search

The TSs are executed until a predetermined number of cycles and the final solutions are obtained. The best of the final solutions is chosen by "Elite" unit as the optimal solution for the problem. Each standard TS can have different parameters.

3.1. Crossover operator used

Genetic algorithms are optimisation algorithms based on natural evolution [7]. They simulate the natural selection and basic genetic operations such as crossover and mutation. The crossover operator is used to create new solution(s) from two existing solutions picked from a set of solutions by a selection operation. Depending on the representation way of the problem in the string form, a proper crossover operator must be applied [8, 9]. This work uses the following intermediate crossover operation: If two existing solutions are

Present solution 1: | W_1 | W_2 | .... | W_i | ..... | W_n |

Present solution 2: | V_1 | V_2 | .... | V_i | ..... | V_n |

a new solution is obtained as

New solution:

| $0.5(W_1+V_1)$ | $0.5(W_2+V_2)$ | .... | $0.5(W_i+V_i)$ | .... | $0.5(W_n+V_n)$ |

4. Training Elman recurrent neural network using tabu search

There are two main types of neural networks (NNs.): feedforward and recurrent [10]. Connections that allow information to loop back to the same processing element are called recursive and NNs having this type connections are named recurrent neural networks (RNNs). RNNs are more suitable than feedforward neural networks (FNNs) for representing a dynamic system since they have a dynamic mapping between its output(s) and input(s). A special type of RNNs is the Elman network (see Figure 3) [11, 12]. The Elman net has feedforward and feedback connections. However, so that it can be trained essentially as feedforward networks by means of the simple backpropagation (BP) algorithm, the feedback connections have to be kept constant. For the training to converge, it is important to select the correct values for the feedback connections. However, finding these values manually can be lengthy trial-and-error process.

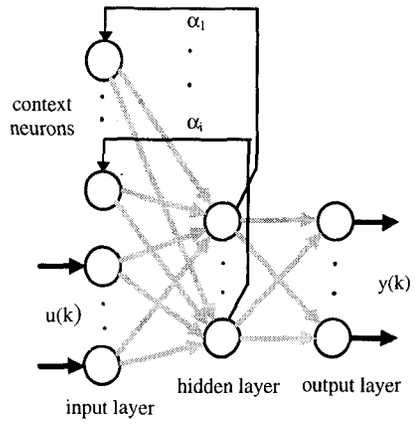

Figure 3. Structure of the Elman network

In this work, the performance of the proposed TS is tested for training the Elman network to identify dynamic plants. The structure of the net employed in this work is selected as in [10, 12] to compare the results. Since the plants to be identified are single-input single-output, the number of the external input neurones and the output neurones is one. The number of neurones at the hidden layer is equal to six. Therefore, the total number of connections is fifty-four of which six are feedback connections.

A solution is represented as a string of trainable weights of which each is a numeric number. When all connections are trainable, feedback connections have weight values in the range 0.0 to 1.0 while feedforward can have positive or negative weights between 1.0 and -1.0. When the feedback connection weights are constant, their values are taken as 1.0 (α_i=1). A neighbour solution is produced from the present solution by adding a randomly generated number between -1.0 and 1.0 to a trainable weight which is not tabu. Note that from the point of view of TS algorithm, there is no difference between the feedback and feedforward connections and training one type of connections is carried out identically to training other, unlike the case of the commonly used BP training algorithm.

In the training stage, first a sequence of the input signals u(k), (k=0,1....) is fed to both the plant and the recurrent network designed with the weights obtained from a neighbour. Second the rms error value between the plant and recurrent network outputs is computed. Next, according to the rms error values computed for neighbours, the next solution is selected. The neighbour solution which

produces the minimum rms error value is selected as the next solution.

In this work, the recency and frequency limits in eq.(1) are taken as

recency limit = number of trainable weights*0.5 (2)

frequency limit = average frequency * 1.5

5. Simulation results and discussion

Simulations were conducted to study the ability of RNN trained by the standard and the proposed TS algorithms to identify a linear and non-linear plant.

Linear plant: This is a third-order linear system with the following discrete-time equation;

$$y(k) = A_1y(k-1) + A_2y(k-2) + A_3y(k-3) + B_1u(k-1) + B_2u(k-2) + B_3u(k-3) \quad (3)$$

where A_1=2.627771, A_2=-2.333261, A_3=0.697676, B_1=0.017203, B_2=-0.030862, B_3=0.014086

The Elman net with all linear neurons was tested. Training input signal, u(k), k=0,1,...99, was random and varied between -1.0 and 1.0. First the results were obtained by assuming that only the feedforward connection weights trainable. Second the results were produced for the Elman net with all connection weights variable. For each case, experiments were repeated six times for different initial solutions. The results obtained by using the standard BP, TS and the proposed TS are given in Figure 4. The responses of the plant and the network designed by the standard and the proposed TSs are presented in Figures 5 and 6, respectively.

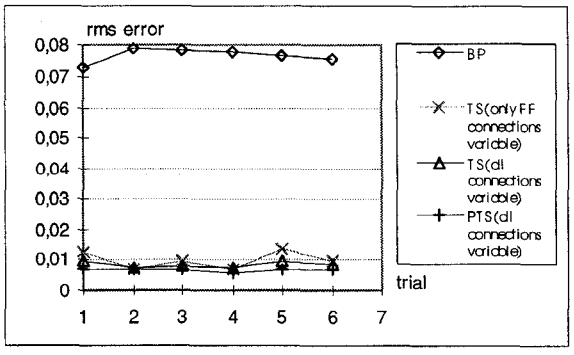

Figure 4. rms error values obtained for the linear plant for six runs with different initial solutions

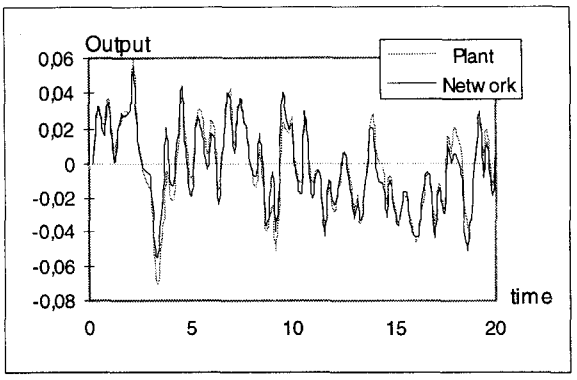

Figure 5. Responses of the plant and the network trained by the standard tabu search (third-order linear plant) (rms error = 0.0093218)

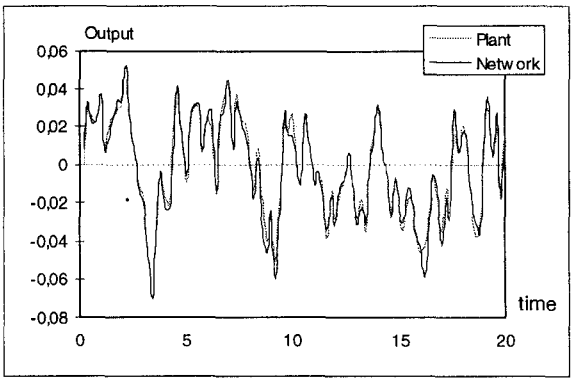

Figure 6. Responses of the plant and the network trained by the proposed tabu search (third-order linear plant) (rms error = 0.0054352)

Non-linear plant: The second plat model adopted for the simulations is that of a simple pendulum swinging through small angles [12]. The discrete-time description of the plant is:

$$y(k) = A_1y(k-1) + A_2y(k-2) + A_3y^3(k-2) + B_1u(k-2) \quad (4)$$

where A_1=1.04, A_2=-0.824, A_3=0.130667, B_1 =-0.16

The Elman net with non-linear neurons in the hidden layer was employed. The hyperbolic tangent function was adopted as the activation function of non-linear neurons. The neural networks were trained using the same sequence of random input signals as mentioned above. As in the case of linear plant, the results were obtained for six different runs with different initial solutions. The rms error values obtained by the standard BP, the standard TS and the proposed TS are presented in Figure 7. The responses of the non-linear plant and the recurrent network with the weights obtained by the standard

and the proposed TSs are shown in Figures 8 and 9, respectively.

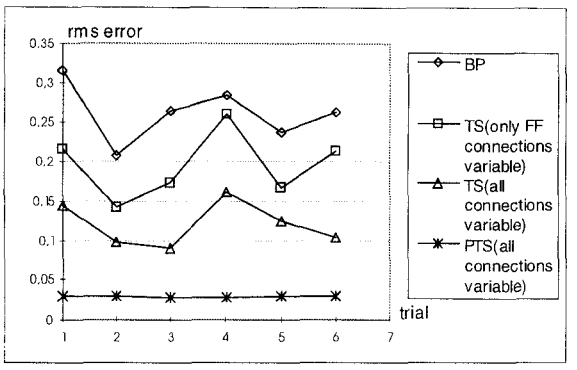

Figure 7. rms error values obtained for non-linear plant for six different runs with different initial solutions

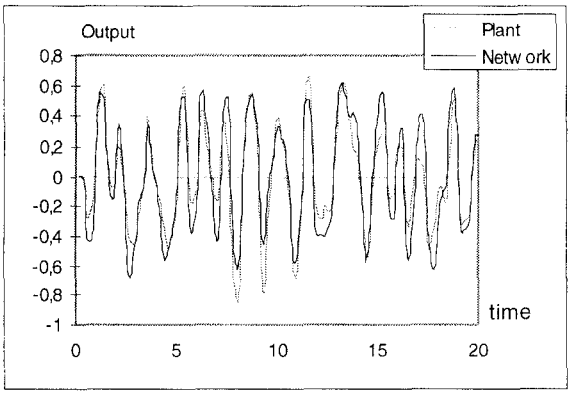

Figure 8. Responses of the non-linear plant and the network trained by the standard tabu search (rms error = 0.141587)

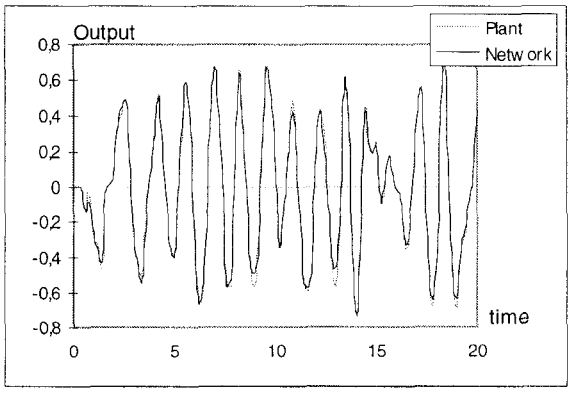

Figure 9. Responses of the non-linear plant and the network trained by the proposed tabu search (rms error = 0.02714256)

The proposed TS was executed during seven cycles. The number of standard TSs running in parallel was four (n=4). The parameters of all standard TSs were the same. The total evaluation number for each cycle was 24000 since each standard TS was run during 6000 evaluations.

The Elman net trained by TSs could identify the third-order linear plant successfully. Note that an original Elman net with an identical structure to that adopted in this work and trained using the standard backpropagation algorithm had failed to identify even second order linear plant [12]. Moreover, when the Elman net had been trained by the standard genetic algorithm, the third-order plant could not be identified although the second-order plant had been identified successfully [10]. It is clear from Figures 4-9 that, for both the net structures (with all connection weights variable and with only feedforward connection weights trainable), the proposed tabu search trained the nets better than the standard BP and the standard TS. The performance of the proposed TS is more evident in the case of identifying non-linear plant although the performance of both TSs is similar for identifying linear plant.

6. Conclusion

In this paper, a parallel tabu search model proposed by the authors was tested for training Elman recurrent neural network to identify linear and non-linear plants. From the simulation results obtained, it is concluded that the proposed TS can be used for training Elman network for identifying linear and non-linear plants efficiently.

7. References

[1] Glover, F., "Future paths for integer programming and links to artificial intelligence", *Computers and Operations Research*, 5, pp.533-549, 1986.

[2] Glover, F. and Grenberg, H.J., "New approaches for heuristic search: a bilateral link with artificial intelligence", *EJOR*, 39, pp.119-130, 1989.

[3] Karaboga, D., Kaplan, A., "Optimising Multivariable Functions Using Tabu Search Algorithms", *The Tenth Int. Symp. on Computer and Information Sciences*, Vol. II, Kusadası, Turkey, pp. 793-799, 1995.

[4] Karaboga,D., Kalinli, A., "Training Recurrent Neural Networks Using Tabu Search Algorithm", *5th Turkish Symposium on Artificial Intelligence and Neural Networks*, pp.293-298, Türkiye, 1996.

[5] Malek, M., Guruswamy, M., Pandya, P. and Owens, H., "Serial and parallel simulated annealing and tabu

search algorithms for the travelling salesman problem", *Annals of Operations Research*, 21, pp.59-84, 1989.

[6] Karaboga,D., Kalinli, A., "A New Model for Tabu Search Algorithm", *IMS'96-The First Turkish Symposium on Intelligent Manufacturing Systems*, pp.168-175, 1996.

[7] Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975.

[8] Michalewicz,Z., "Genetic Algorithms+Data Structures = Evolution Programs", Springer-Verlag, USA, 1992.

[9] Goldberg, D.E., "Genetic Algorithms in Search, Optimisation", and Machine Learning, Addison-Wesley, Reading, Mass, 1989.

[10] Pham, D.T. and Karaboga, D., "Training Elman and Jordan networks for system identification using genetic algorithms", *Journal of Artificial Intelligence in Engineering*, (to appear), 1997.

[11]Elman, J.L., "Finding structure in time", *Cognitive Science*, Vol.14, pp.179-211, 1990.

[12]Liu, X., "Modelling and Prediction Using Neural Networks", PhD Thesis, University of Wales College of Cardiff, Cardiff, UK, 1993.