Defence Science Journal, Vol. 56, No. 4, October 2006, pp. 627-636
© 2006, DESIDOC

Solving Battle Management/Command Control and Communication Problem using Modified BIONET

S. Thamarai Selvi and R. Malmathanraj
Madras Institute of Technology, Chennai - 600 044

ABSTRACT

This paper proposes and implements a neural architecture to solve the weapon allocation problem in the multi-layer defense scenario using modified BIONET neural network architecture. The presynaptic layer of the modified BIONET reduces the dimensionality of the principal state equation by partitioning the state space. The post-synaptic layer of the modified BIONET includes the perceptron Q-learning rule. The cortical layer incorporates L-learning scheme to provide better exploration over action space. Thus, action selection is effectively made with quicker convergence of training. The reward scheme in the reinforcement learning is obtained by calculating the measure of probability of survival. The decision module has been enhanced by incorporating the features corresponding to the battle weapons for effective representation of the environment. Thus, the modified BIONET neural architecture is used to increase the efficiency of assets saved in the simulation and the time complexity is reduced due to the state-space partitioning scheme involved in the neural network. The proposed modified BIONET is implemented in MATLAB and the percentage of assets saved is increased. Also, the training time is drastically reduced. Thus, the modified BIONET resulted in saving more assets with faster convergence of learning.

Keywords: Reinforcement learning, modified BIONET, radial basis function neural network, fuzzy inference system, multi-layer defence, battle management

1. INTRODUCTION

The weapon allocation problem in a multi-layer defence is a combinatorial optimisation problem. In general, the modern technology era has its impact on the battlefield scenario. In particular, precision guided weapons and improved nuclear arsenal add depth to the attacking capability of nations. There is a need for adaptive and automated decision-making process for effective allocation of defence resources. The process of effectively allocating defence resources against a perceived enemy threat is known[1] as battle management/command, control, and communication (BM/C^3) problem. Reinforcement learning has been used in constructing several decision modules. Several mathematical models[2,3] have been developed for theatre missile defence (TMD) and BM/C^3 problems.

The solution for these problems is constrained by various factors such as hyper dimensional state space, memory, and time requirements. Hence, the problem is solved using neural network model with adaptive learning technique. This paper proposes a new solution methodology for the BM/C^3 problem by including six different priority-attacking weapons

Revised 24 October 2005

and six different priority-defending weapons in the environment. The simulation includes the number of assets as 100. This study uses the mathematical model of the BM/C^3 problem for exact modelling of the environment. It adapts the probability of survival calculation to overcome the attacking weapon for accurate reward accumulation.

The reinforcement learning is the learning technique in which the agent tries to learn a policy. The policy defines how to select an action in a given state of the environment. The aim of the reinforcement learning is to maximise the reward when interacting with the environment. The major problem present in the reinforcement learning domain is the state-space explosion. This paper uses the modified BIONET network architecture and a reinforcement learning technique to learn from examples. The neural network structure is capable of searching the larger regions of the solution space roughly and globally. In the reinforcement learning, an optimal policy is inductively learned by the Q value function. The Q-learning algorithm takes the state action pair as input and yields the quality of the action selected in that state as output. During initial stages of learning, an action is selected by the exploration scheme. The optimal action in a given state is the action with the largest Q value. Thus in reinforcement learning algorithms, the knowledge acquired is encoded in *q* values.

The reinforcement learning is used for a wide range of problems such as game playing[4,5], robotics[6], scheduling and inventory control[7]. The constraint is due to the exponential increase in the number of admissible states with the dimensionality of state space. This limitation is called as the Bellman's curse of dimensionality. The Q-learning converges only after every state has been visited many times. Both the problems are solved using the modified BIONET neural architecture. This neural architecture provides an efficient generalisation over state space. The generalisation allows to estimate the *q* value of a state action pair without even visiting the state. The partitioning approach used overcomes the state explosion problem. The partitioning can be categorised into two viz., hard partitioning and soft partitioning.

The soft partitioning allows the data to lie simultaneously in multiple regions[8-10]. The hard partitioning splits allow the data to be present unique in any one of the regions[11-13]. The advantages of partitioning are that it facilitates quicker convergence and better learning.

2. PROBLEM STATEMENT

This problem is cast as a constrained optimisation problem involving six priority assets with the number of assets for each priority as maximum as 100. The allocation of defence resources to overcome the multi-priority attack is generally called BM/C^3 (Fig. 1).

2.1 Assumptions

The BM/C^3 problem is solved by considering the following assumptions:

- The defence resources are of surface-to-air type.
- The attacking resources are of air-to-surface type.
- These resources are mobile in nature.
- There are six different priority-attacking weapons and six different priority-defending weapons.

The developed mathematical model depicts the problem as follows:

$$prob(s)=\left[\prod_{a=1}^{A}\left[1-\left\{\prod_{d=1}^{D}\left(1-k_{dsa}\right)^{x_{dsa}/n_{sa}}\right\}g_{sa}\right.\right]^{n_{sa}} \tag{1}$$

where

$prob(s)$ Probability of survival

d Types of defending weapons available

s Number of assets

a Types of attacking weapons

k_{dsa} Probability of successful interception by one defending weapon of type d deployed to defend an asset s against an attacking weapon of type a

n_{sa} Number of attacking weapons of type a aimed at asset s

x_{dsa} Number of defending weapons of type d at asset s allotted to overcome the attack wave of type a

g_{sa} Damage probability

A Maximum priority number for attacking weapon

D Maximum priority number for defending weapon.

In the problem setup, A_1-A_6 denotes the priority of the attacking weapons and x_{ijk} denotes the defending weapons k allotted for the i^{th} priority attacking weapons to safeguard the j^{th} asset. The mathematical optimisation model is used for calculating the reward function and state-space transition. The design goal of the modified BIONET neural architecture with reinforcement learning is to train the neural architecture to achieve a survivability of up to 75 per cent of the assets. The simulation results of the problem setup provides the output for one asset at a time instant, and the same action can be applied to all the assets that perceive threat. The problem is considered to be dynamic with the presence of 100 different assets with varying attacking/ defending scenarios for the assets.

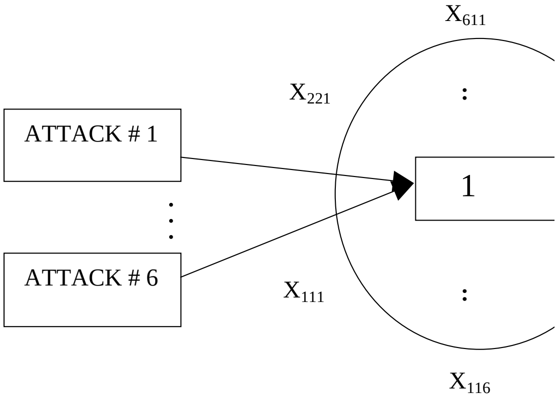

Figure 1. Battle management/command control and communication problem setup.

The numerical features regarding the state of the environment is extracted and these are described using fuzzy linguistic variables. The state variable (S) is fed as input into the modified BIONET neural architecture.

$$S = (F_1,\ F_2,\ F_3 \ldots\ldots\ldots F_{11});$$

where

F_1 Total operating cost

F_2 Manpower required

F_3 Precision of weapon and effectiveness/ kill ratio

F_4 Area required for operation

F_5 Accuracy of tracking radars

F_6 Accuracy of radars for guiding missiles

F_7 Number of assets remaining in the top 3 priority regions

F_8 Precision of weapon and effectiveness of the opponent weapons

F_9 Speed of execution/time factor involved

F_{10} Ability to launch successive warheads

F_{11} Availability of auxiliary vehicles

3. MODIFIED BIONET ARCHITECTURE

The modified BIONET architecture provides a new framework to perform learning on aggregated states. The state action space is partitioned into a set of disjoint smaller subset states X_i. The q value for all states in one subset is the same. The ANN model BIONET[14] is proposed based on the neurophysiology of the human nervous system. BIONET is a four-layered feedforward neural network which overcomes the limitations of the conventional multi-layer feedforward neural networks such as slower convergence and difficulty in fixing the number of neurons in the hidden layer. The modified BIONET neural network architecture ensures quicker convergence and state-space partitioning is performed by one of the two layers of modified BIONET. The problem can be solved with drastic time reduction by partitioning the state space. The states are grouped considering the measure of closeness of certain features among the states in the environment. The features depict the battle readiness of the defending nation and the armour capabilities of the attacking nation. After every attacking wave, the time taken by the defending nation to respond is constrained by the practical difficulties involving the machineries. In this study, the mean feature set (mF_1, mF_2, mF_3,………,mF_{11}) of the environment is used as the current state of the environment which is obtained from subsequent attack waves. The difference is used as a feature space for solving this problem.

The *q* learning is a reinforcement learning method where the learner builds incrementally a Q function, which attempts to estimate the discounted future rewards for taking actions from any initial state. The output of the Q function for state *x* and action *a* is denoted by Q(*x*,*a*). When action *a* has been chosen and applied, the system moves to a new state, *y*, and a reinforcement signal, *r*, is received. Q(*x*,*a*) is updated by

$$Q(x,a) = [(1-\alpha)\ Q(x,a) + \alpha\{r + \gamma V(y)\}] \quad (2)$$

where

$V(y)$ is the value of the state y, defined by

$$V(y) = \max_{b\varepsilon A(y)} Q(y,b)$$

where $A(y)$ is the possible action set, α is the learning rate and g is a discount factor.

The Q value function is approximated by perceptron *q* learning rule in the postsynaptic layer. The time complexity of learning the Q function is reduced using the modified neural architecture.

The study further addresses state-space partitioning, used to group the homogenous states into separate regions. Further, partitioning can improve learning to a degree. The convergence times of the partitioning algorithm are often faster than gradient-based algorithms. The *L*-learning exploration technique is used to select actions from the Q value network.

In this study, the partitioning of state space is effectively performed by the modified BIONET neural network architecture. The network learns to cluster the situations based on the features extracted from the environment than clustering by considering the numerical closeness of the state equations. The architecture is shown in Fig. 2. The input are fed to the stimulus layer. Connections to the receptor layer are made in such a way that the neurons in the receptor layers receive input from the stimulus layer. The receptor layer is partitioned into *M* groups of neurons where *M* is the number of patterns to be classified. Each group in the receptor layer consists of the same number of neurons same as the stimulus layer required to state the facts about the environment. The cortical layer consists of total number of neurons the same as the number

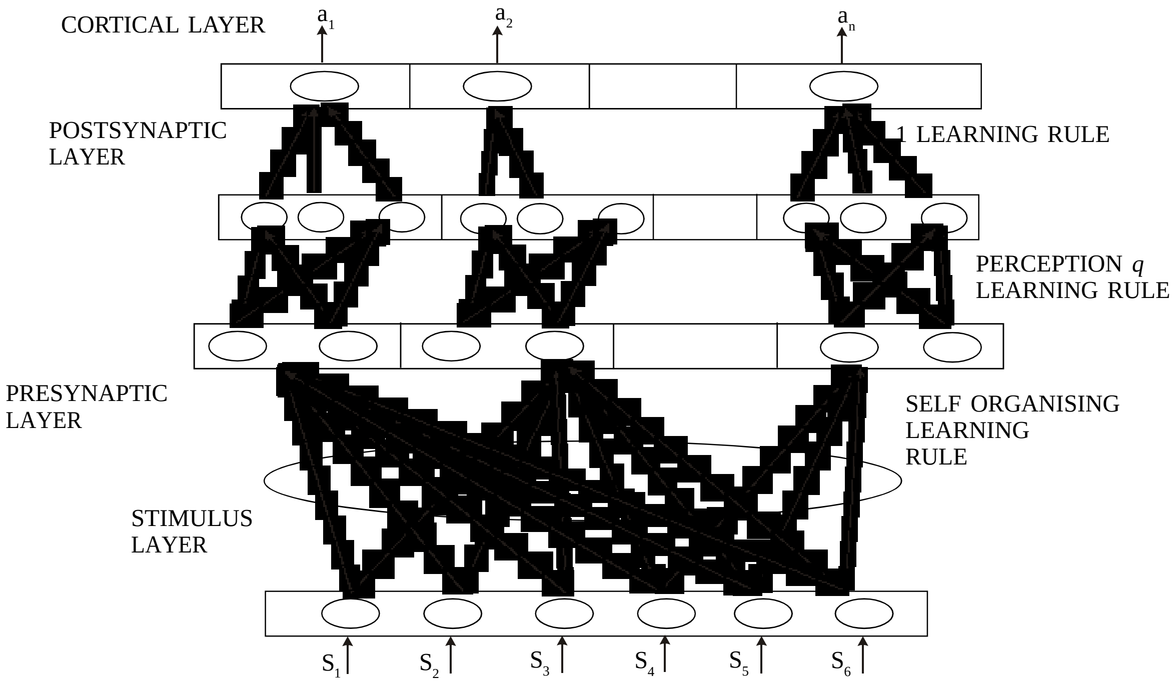

Figure 2. Modified BIONET architecture.

of actions available with the decision module. Based on the hypotheses[14], the architecture has four layers, viz., stimulus layer, presynaptic receptor layer, postsynaptic receptor layer, and cortical layer.

The stimulus layer stimulates the input signals. These signals are received by the receptor layer as done by the sensory organs in the human beings. The presynaptic layer learns to partition the state space into states without ambiguity. The output of the presynaptic layer is a single integer $i(z)$, which denotes the index of the aggregated state to which z belongs.

4. LEARNING ALGORITHM

The modifications made in the learning algorithm include the self-organising learning rule for the presynaptic layer, the perceptron q-learning rule for the postsynaptic layer, and the L-learning rule for the cortical layer. The stimulus layer receives the input state information. The state input present in the interceptor allocation problem results in a combinatorial explosion of state space. The presynaptic layer consists 10 cells and each cell provides the representative state information for that cell, viz., (10, 10, 10,…, 10), (20, 20,…, 20). The post-synaptic layer consists of 10 cells. Each cell contains 18 perceptron, with each perceptron denoting the index of the action used in the simulation. The q values for every action is performed using the perceptron q-learning rule.

4.1 Presynaptic Receptor Layer

The presynaptic receptor layer employs self-organising learning rule for the clustering of states into regions. The principal goal of self-organising map is to transform the input signal of arbitrary dimension into a one or two-dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion. This mapping of input vectors into topological order is motivated by the topological ordering of different sensory input such as visual and auditory features separately in the human cortex system. The output of presynaptic layer is topologically arranged in the sense that the closer input vectors are represented by closer neurons. In the self-organising learning rule, the transformation of weight vector towards input vector constitute the learning process.

For a network to be self-organising, the synaptic weight vector w_j of neuron j should change in relation to the input vector. The training process defines the quantity of change to be performed.

$$\Delta w_j = \eta\, y_j x - g\left(y_j\right) w_j \qquad (3)$$

where

η is the learning rate parameter

w_j is the synaptic weight vector of neuron j

y_j is the output of the neuron j

g(y_j) is the positive scalar function of the response y_j. A condition is implemented in g(y_j) such that g(y_j) = 0 for y_j = 0. To satisfy the above condition, g(y_j) = 0.

To maintain the topology of output, the weight vector of few neurons surrounding the winner neuron are also modified.

$$y_j = h_{j,i(x)}$$

By substituting all the above equations in the first equation, one gets

$$\Delta w_j = \eta\, h_{j,i(x)} (x - w_j) \qquad (4)$$

Now using the discrete time formulae,

$$w_j(n+1) = w_j(n+1) + \eta(n)\, h_{j,i(x)} (x - w_j(n)) \qquad (5)$$

where $i(x)$ denotes the index of the winner neuron, and $h_{j,i(x)}$ denotes the neighborhood surrounding the winner neuron.

The value $w_j(n+1)$ gives the updated weight vector. This equation has the effect of minimising the Euclidean distance between weight vector and input vector. Upon repeated presentations of the training data, the synaptic weight vector tends to follow the distribution of the input vectors due to neighborhood updating. The algorithm therefore leads to the topological ordering of the feature map in the input space in the sense that neurons that are adjacent in the lattice will tend to have similar synaptic weights. The output of the presynaptic receptor provides the representative states for learning by reinforcement. These representative states parameterise the complete state space over a group of states. The output of the presynaptic receptor groups similar states as according to the prevailing rule base, and the output of this module provide a single digit numeric value (z). The value (z) serves as the approximate indicator denoting the minimum number of resources to be used for overcoming the incoming attack. The actions with number of resources lesser than z are forbidden from competing in the S_2 layer.

4.2 Postsynaptic Receptor Layer

The value function approximation in the reinforcement learning is performed by the postsynaptic receptor layer. The function approximation in the reinforcement learning schemes allows the decision module to generalise from the states it has visited to the states it has not visited. The delta rule is modified to adapt the Q-learning procedure. In the reinforcement learning schemes, the parameters are updated after each trial. The rate of change of error wrt each parameter θ_i is $\partial E_j / \partial \theta_i$. The parameters θ_i have to be updated towards the direction of minimum error.

$$\theta_i < - \; \theta_i - \alpha \partial E_j / \partial \theta_i \tag{6}$$

where α is the learning rate.

By combining with temporal difference learning, the above equation is modified into the following equations:

$$\theta_i < \; - \; \theta_i + \alpha \lfloor R(s) + \gamma \max_{a'} Q_\theta(a', s') - Q_\theta(a, s) \rfloor \; \delta Q_\theta(a', s') / \partial \theta_i \tag{7}$$

where $R(s)$ denotes the reward function, $Q_\theta(a', s')$ denotes the q value of the (s, a) pair updated by performing the action a selected by exploration technique.

In the postsynaptic receptor layer the tan$h$ activation function is used as its antisymmetric and facilitates learning process. The action selection from value function is performed using the L-learning technique[15].

5. IMPLEMENTATION ISSUES

The simulation scheme was performed with a test setup of 100 assets, 6 priority of attacking weapons and 6 priority of defending weapons. The maximum number of defending weapons available for all priority weapons was 100. An action was selected to maximise the survival ratio of assets.

The weapon selection matrix used in the simulation is

```
A = [1 2 3 4 5 6;
     4 5 6 7 8 9;
     7 8 8 9 2 3;
     3 4 5 2 1 1;
     2 3 4 5 6 7;
     4 5 6 7 8 1;
     6 7 7 7 8 2;
     5 6 6 6 6 2;
     7 8 8 8 8 8;
     2 3 4 5 5 5];
```

where every element a_{ij} ε A denotes the row index of the weapon repository. Let one suppose that action index selected is i = 3; This implies that the 3rd row of the repository is selected. Now the weapon repository is denoted by

$$\begin{array}{llllllll} w = [& w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & w_{16}; \\ & w_{21} & w_{22} & w_{23} & w_{24} & w_{25} & w_{26}; \\ & w_{31} & w_{32} & w_{33} & w_{34} & w_{35} & w_{36}; \\ & w_{41} & w_{42} & w_{43} & w_{44} & w_{45} & w_{46}; \\ & w_{51} & w_{52} & w_{53} & w_{54} & w_{55} & w_{56}; \\ & & . & . & . & & \\ & w_{n1} & w_{n2} & w_{n3} & w_{n4} & w_{n5} & w_{n6}]; \end{array}$$

then the number of defence resources allotted to overcome the first priority weapons is [w_{71},…,w_{7n}]; and the weapons allotted to overcome the second priority weapons is [w_{81},…,w_{8n}]; The presynaptic layer performs the partitioning of input state space. This layer uses the winner take all algorithm and any one of the cells present in the presynaptic layer is fired. This cell provides the representative state input to the postsynaptic layer as explained in Section 3. The Q function is approximated by the perceptron Q-learning technique[16]. The tan*h* activation function is used due to the antisymmetric property and it facilitates learning better.

This study implements the *L*-learning scheme for action selection. The *L*-learning scheme is implemented with the value of γ_l as 0.95 and value of γ_q as 0.95. The action space explored by the *L*-learning technique is shown in Fig. 3. The better exploration in the initial stages of learning helps the modified BIONET to acquire quicker convergence. During each episode, the agent begins at the initial state *s* and is allowed to perform actions until it reaches the goal state. The modified BIONET architecture performs various learning trials with different initial states. In the first episode of learning, maximum number of steps is required to reach the goal. As the number of iteration increases, the steps to goal are minimised. The Modified BIONET architecture achieves faster convergence in the learning trial as shown in Figs 4 and 5.

The goal of the reinforcement learning is set to attain the maximum survivability of assets as 75 per cent. The multiple reward values are obtained from the environment using probability of survival calculation. Figure 4 denotes the learning trial with the total number of assets used in simulation as 100. Figure 5 denotes the learning trial with the total number of assets used in simulation as 200. The inference from the graphs show quicker convergence of modified BIONET.

Table 1 shows the number of defending weapons allocated to overcome the multiple priority attack. The output of the cortical layer is an action index which indicates the row number of the defending weapon selection matrix. From the graphs, the quicker convergence of the modified BIONET can be observed.

The neural network structure is capable of searching the larger regions of the solution space roughly and globally. The defence plan showing the number of weapons allocated to overcome the multiple priority attack weapon for the first asset with the model probability value g_{sa} and k_{dsa} as given in the simulation is shown in Table 1.

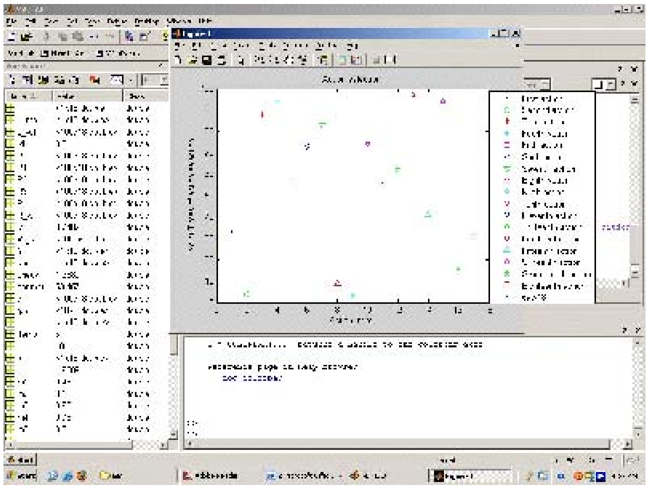

Figure 3. Plot to show the action space exploration by *L* learning exploration.

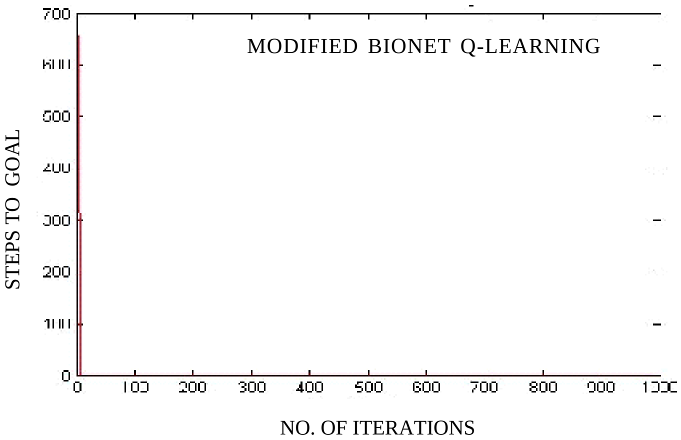

Figure 4. Plot to show the learning trial with total number of assets used in simulation as 100.

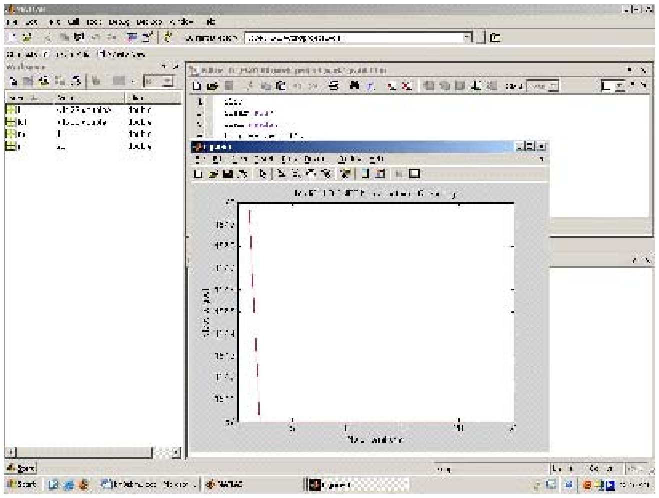

Figure 5. Plot to show the learning trial with total number of assets used in simulation as 100.

Table 1. Optimal defence plan developed using the modified BIONET neural architecture

Defending weapon type	Asset	Attacking weapon type	k_{dsa}	g_{sa}	Defence plan
1	1	1	0.8	0.231	12
2	1	1	0.2	0.531	10
3	1	1	0.3	0.215	14
4	1	1	0.4	0.123	6
5	1	1	0.5	0.215	8
6	1	1	0.25	0.313	1
1	1	2	0.231	0.231	12
2	1	2	0.531	0.531	10
3	1	2	0.215	0.215	14
4	1	2	0.123	0.123	8
5	1	2	0.215	0.215	8
6	1	2	0.313	0.313	4
1	1	3	0.231	0.231	13
2	1	3	0.531	0.531	10
3	1	3	0.215	0.215	14
4	1	3	0.123	0.123	7
5	1	3	0.215	0.215	8
6	1	3	0.313	0.313	2
1	1	4	0.8	0.8	12
2	1	4	0.2	0.2	14
3	1	4	0.3	0.3	10
4	1	4	0.4	0.4	9
5	1	4	0.5	0.5	8
6	1	4	0.25	0.25	5
1	1	5	0.8	0.8	13
2	1	5	0.2	0.2	10
3	1	5	0.3	0.3	14
4	1	5	0.4	0.4	7
5	1	5	0.5	0.5	8
6	1	5	0.25	0.25	3
1	1	6	0.8	0.8	12
2	1	6	0.2	0.2	10
3	1	6	0.3	0.3	14
4	1	6	0.4	0.4	10
5	1	6	0.5	0.5	2
6	1	6	0.25	0.25	6

6. CONCLUSION

This paper proposes an efficient solution for the BM/C^3 problem involving the sequential allocation of defence resources over a period of time sequences. The proposed system facilitates learning better due to the partitioning of state space and the adaptation of L-learning exploration technique to have maximum exploration over action space. The learning is performed with ease using modified BIONET neural network. The partitioning of state space concept is used along with Q value function to learn from experience. Further, the radial basis function (RBF) network is used for Q function approximation. The graphs obtained from the simulation results show the efficiency of the decision module in terms of the number of trials.

REFERENCES

1. Bisht. Hybrid genetic algorithm for optimal weapon allocation. *Def. Sci. J.*, July 2004, **54**(3).

2. Bertsekas, Dimitri P.; Homer, Mark L.; Logan David A. & Patek, Stephen D. Missile defence

and interceptor allocation by neuro dynamic programming. *IEEE Trans. Syst, Man Cybern.*, January 2000, 3**0**(1).

3. Mantle, Peter, J. The missile defence equation. AAAI Press, 2005.

4. Tesauro, G.J. Practical issues in temporal difference learning. *Machine Learning*, 1992, **8**, 257-77.

5. Tesauro, G.J. TD-Gammon, a self-teaching backgammon program achieves master-level play. *Neural Computation*, 1994, **6**(2), 215-19.

6. Reidmiller, M. Application of sequential reinforcement learning to control dynamic systems; *In* Proceedings of IEEE International Conference on Neural Networks, 1996, pp. 167-72.

7. Mahadevan, S.; Marchellak, N.; Das, K.T. & Gosavari, A. Self-improving factory simulation using continous time average reward reinforcement learning. *In* Proceedings of the 14th International Conference on Machine Learning, 1997. pp. 202-10.

8. Bridle, J. Probabilistic interpretation of feedforward classification network output with relationship to the statistical pattern recognition. *In* Neuro computing: Algorithms, architectures and Applications, edited by Joulie F. Fogelman and J. Herault. New York, Springer Verlag, 1989.

9. Nowlan, S.J. Soft competitive adaptation, neural network algorithms based on fitting statistical mixtures. Technical Report CMU-CS-91-126, CMU, Pittsburgh, P.A. 1991.

10. Wahba, G.; Gu, C.; Wang,Y. & Chappell, R. Soft classification, a.k.a, risk estimation via penalised log likelihood and smoothing spline analysis of variance, 6, Technical Report , University of Wisconin, Madisson. 1993.

11. Breiman, L.; Friedman, J.H.; Olshen, R.A. & Stone, C.J. Classification and regression trees. Wadsworth International Group, Belmont, CA, 1984.

12. Friedman, J.H. Multivariate adaptive regression splines. *Annals of Statistics*, 1991, **19**, 1-41.

13. Quinlan, J.R. Induction of decision trees. *Machine Learning*, 1986, **1**, 81-106.

14. Thamarai Selvi, S.; Arumaugam, S. & Ganesan, L. BIONET: An artificial neural network model for diagnosis of diseases. *Pattern Recog. Lett.*, 2000, **21**, 721-40.

15. Iwata, K.; Ikeda & Saka, H. A new criterion using information gain for action selection strategy in reinforcement learning. *IEEE Trans. Neural Networks*, July 2004, **15**(4), 792-99.

16. Russell, Stuart & Norvig, Peter. Artificial intelligence: A modern approach. PHI 2003.

17. Kaelbling, L.P.; Littman, M.L. & Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.*, **4**, 237-85.

18. Jouffe, L. Fuzzy inference system learning by reinforcement methods. Technical Report INSA 96081, 1996.

19. Glorennec P.Y. Fuzzy Q-learning and dynamical fuzzy Q learning. Proceedings of the 3rd IEEE International Conference on Fuzzy Systems, Orlando, June 1994.

20. Berenji H. & Khedkar P. Learning and tuning fuzzy logic controllers through reinforcement. *IEEE Trans. Neural Networks*, September 1992, **3**(5).

21. Castro, J.L. Fuzzy logic controllers are universal approximators. *IEEE Trans. SMC*, April 1995, **25**(4).

22. Jang, J.S.R.; Sun, C.T. & Mizutani, E. Neuro fuzzy and soft computing. PHI 1997.

23. Wyatt, Jereme. Exploration and inference in learning from reinforcement. PhD Dissertation, University of Edinbourgh, 1997.

24. Thrun, S. Efficient exploration in reinforcement learning. Technical Report CS-92-102, Carnegie Melon University.

25. Ivan, S.K.; Lee, Henry & Lau, Y.K. Adaptive state-space partitioning for reinforcement learning, *Engg. Appli. Arti. Intell.*, 2004, **17**, 577-88.

Contributors

Dr (Ms) S. Thamarai Selvi received BE (Mech Engg) from the Madhurai Kamaraj University, ME (Computer Science & Engg) from the Bharathiar University, and PhD (Computer Science & Engg) from the Manonmaniam Sundaranar University. She is currently Professor and Head, Dept of Information Technology at MIT Campus of Anna University, Chennai. Her areas of research include: Artificial neural networks, grid security, and semantic grid services.

Mr R. Malmathanraj received BE (Electronics and Communication Engg) from the Manonmaniam Sundaranar University, and ME (Communications Systems) from the Madhurai Kamaraj University. He is currently pursuing his doctoral studies at the Anna University. His research interests include: Neural networks and image processing.