A stochastic dynamic local search method for learning Multiple-Valued Logic networks

メタデータ	言語: English 出版者: 公開日: 2017-10-03 キーワード (Ja): キーワード (En): 作成者: Cao, Qiping, Gao, Shangce, Zhang, Jianchen, Tang, Zheng, Kimura, Haruhiko メールアドレス: 所属:
URL	http://hdl.handle.net/2297/19141

PAPER

A Stochastic Dynamic Local Search Method for Learning Multiple-Valued Logic Networks

Qiping CAO[†], **Shangce GAO**[††], **Jianchen ZHANG**[††], ***Nonmembers***, **Zheng TANG**[††a)], ***and*** **Haruhiko KIMURA**[†††], ***Members***

SUMMARY In this paper, we propose a stochastic dynamic local search(SDLS) method for Multiple-Valued Logic (MVL)learning by introducing stochastic dynamics into the traditional local search method. The proposed learning network maintains some trends of quick descent to either global minimum or a local minimum, and at the same time has some chance of escaping from local minima by permitting temporary error increases during learning. Thus the network may eventually reach the global minimum state or its best approximation with very high probability. Simulation results show that the proposed algorithm has the superior abilities to find the global minimum for the MVL network learning within reasonable number of iterations.

key words: *Multiple-Valued Logic, Local search, stochastic dynamic, temporary, iteration*

1. Introduction

Multiple-Valued Logic (MVL) has been the subject of much research over many years [1–3]. Multiple-valued logic circuits and systems, which contain multiple-valued logic circuits, multiple-valued numeric logic configuration and logic design, together with multiple-valued logic switch theory, are the main domain of the MVL [4–6]. There are several kinds of methods to design Multiple-Valued numeric system, such as the algebraic method [7–11], the circuit method [12], the theorem-proving techniques [13, 14], the modular design approach [15], and the hyperplanes method [16], etc.

Recently, the ability of MVL networks to accumulate knowledge about objects and processes using learning algorithms makes their application in pattern recognition very promising and attractive [11, 17–20]. In particular, different kinds of neural networks are successfully used for solving the image recognition problem [21]. Neural networks based on multi-valued neurons have been introduced in [22] and further developed in [23–26]. Multi-valued neural element(MVN) is based on the ideas of multiple-valued threshold logic [27]. Its main properties are ability to implement arbitrary mapping between inputs and outputs described by partially defined multiple-valued function. They have been proposed for solving the image recognition problems. Different models of associative memory have been considered in [23, 24, 27–29].

Although several error back-propagation based [30–33] and global searching algorithms (such as Genetic Algorithms) [34–37] have been proposed to emulate MVL functions, many nodes and therefore many parameters (weights and thresholds) are usually necessary to approximate an MVL function and these techniques cannot use any knowledge which is available prior to training. Furthermore, although Genetic Algorithms provide an alternative method to problems that are different to solve with traditional optimization techniques, they suffer from poor convergence properties and difficulties to reach high-quality solutions in reasonable time [38]. In [39, 40], the authors proposed a learning MVL network that uses the prior knowledge we have on MVL network while constructing an MVL network and conducts learning in a manner analogous to neural back-propagation. In these techniques, derivatives of the node functions are required, but they generally do not exist. Furthermore, since derivatives of these piece-wise functions are zero for most inputs, learning cannot be performed efficiently. A metaheuristic [41], such as local search [42] and simulated annealing [43] may provide a good solution to this problem. The authors have therefore, been directed towards a learning MVL network which performs learning by a non-back-propagation manner and proposed a new learning method for Multiple-Value Logic (MVL) networks using the local search method [18]. It is a "non-back-propagation" learning method which constructs a layered MVL network based on canonical realization of MVL functions, defines an error measure between the actual output value and teacher's value and updates a randomly selected parameter of the MVL network if and only if the updating results in a decrease of the error measure.

However, due to their inherent local minimum problems, all these learning algorithms, either the error back-propagation based algorithms or the non error back-propagation based algorithms often converged to

Manuscript received September 13, 2006.
Manuscript revised November 17, 2006.
Final manuscript received January 5, 2006.

[†]Tateyama Institute of System, Toyama-shi, 930-0016 Japan
[††]Faculty of Engineering, Toyama University, Toyama-shi, 930-8555 Japan
[†††]Graduate School of Natural Science, Kanazawa University, Kanazawa-shi,920-1192 Japan
a) E-mail: ztang@eng.u-toyama.ac.jp

a local minimum solution that is far from the optimal solution. In this paper, we propose a stochastic dynamic local search method for MVL learning by introducing stochastic dynamics into the local search. The constructed system maintains some trend of quick descent to local minima, and at the same time has some chance of escaping from them by permitting temporary error increases during learning. So the system may eventually reach the global minimum state or its best approximation with very high probability.

This paper is organized as follows: in the next section, the multiple-valued logic network is briefly described. The original local search method is given in section 3. In section 4, we propose a new local search method by incorporating stochastic dynamics, and the simulation results are shown in section 5. Finally we give some general remarks to conclude this paper.

2. Multiple-Valued Logic (MVL) NETWORK

The values of the signal used by radix R are most commonly the extension of the positive integer binary notation. Given the set $Q = \{0, 1, ..., R-1\}$ for any R-valued system, we find the multiple-variable MAX and MIN operators, together with many variants of single-variable literal operators widely employed.

(1)MAX and MIN operators:

$$x_1 + x_2 + ... + x_n = MAX(x_1, x_2, ..., x_n) \quad (1)$$

$$x_1 \cdot x_2 \cdots x_n = MIN(x_1, x_2, \ldots, x_n) \quad (2)$$

(2)Literal operators:

$$x(a,b) = \begin{cases} \text{R-1} & a \le x \le b \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The operators mentioned above together with a constant operator ,for example, the literal operator $x(a,b)$, structure the algebra for functional completeness of R radix. In this condition, any R-valued function can be synthesized in a sum-of-products form.

$$F(x_1, x_2, ..., x_n) = \sum_{e_1, e_2, ..., e_n} F(e_1, e_2, ...e_n) \cdot x_1(e_1, e_1) x_2(e_2, e_2) \cdots x_n(e_n, e_n) \quad (4)$$

where $x_1, x_2, ..., x_n$ are R-valued variables, $e_i \in 0, 1, 2, ...R-1$, $i = 1, 2, ..., n$, $F(e_1, e_2, ..., e_n) \in 0, 1, 2, ..., R-1$

We consider an R-valued logic system of n inputs $x_1, x_2, ..., x_n$,and one output $F(x_1, x_2, ..., x_n)$. Fig.1 shows the general realization topology for the R-valued combinatorial function, using MAX, MIN and literal operators. Node functions in the same layer are of the same type, as described below:

Layer 1: Each node in this layer is a literal function and its node function is given by Eq.4. The window parameters of the i-th input to the j-th MIN gate are defined as $a_{ij}, b_{ij} (a_{ij}, b_{ij} \in 0, 1, 2, ..., R-1 \quad and \quad a_{ij} \le b_{ij} (i = 1, 2, ..., n \quad and \quad j = 1, 2, ..., R^n - 1))$.The literal function for the node function is shown is Fig.2. As the values of the a_{ij} and b_{ij} change, the literal function varies accordingly, thus exhibiting various forms of literal functions.

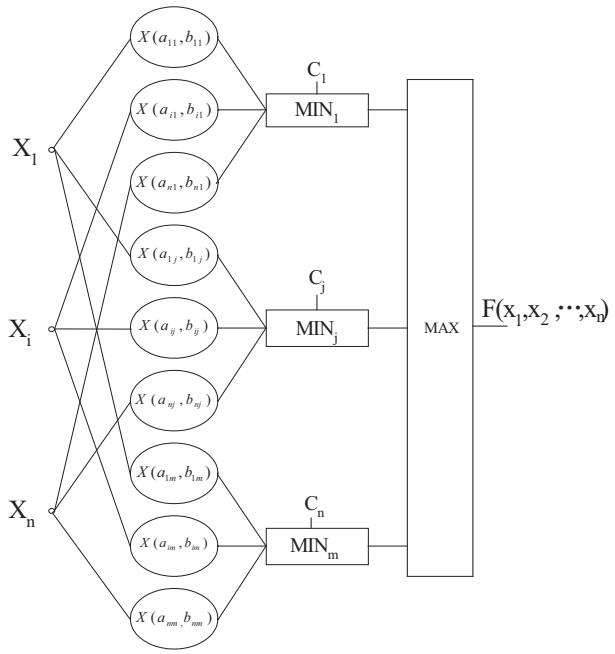

Fig. 1 A learning MVL architecture based on a canonical realization of MVL function

Layer 2: A node in layer 2 corresponds to the MIN operation. Each node selects a particular area of the function and defined its function value 1 or 2 or ... or (R-1) by means of the logic signal 1 or 2 or ... or (R-1) included within the MIN term. Thus, it corresponds to m MIN nodes, maximally $(R^n - 1)$ MIN nodes. The function is given by

$$MIN_j = MIN(c_j, x(a_{1j}, b_{1j}), ..., x(a_{ij}, b_{ij}), ..., x(a_{nj}, b_{nj})) \quad (5)$$

where c_j is biasing parameter of MIN_j with the logic signal 1 or 2 ... or (R-1).

Layer 3: This node gives a MAX operator between the product terms:

$$O = MAX(MIN_1, MIN_2, ..., MIN_m) \quad (6)$$

The learning MVL network described above is a multi-layered feed-forward network in which each node performs a particular function (a node function) on incoming signals using a set of parameters specific to this node. The form of the node function varies from layer to layer, and each node function can be defined by prior knowledge on the network. Unlike the traditional neural networks, the MVL networks give the maximal numbers of the nodes needed for any MVL functions.

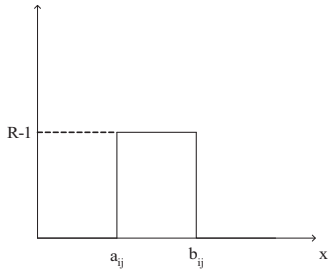

Fig. 2 The definition of a literal function

3. Local Searching Method for MVL Network

As mentioned above, any multiple-valued logic function can be synthesized in a network as shown in Fig.1 with appropriate window parameters and biasing parameters. Thus, during the learning process, these parameters can be adjusted in order to optimize an error function. For simplicity, we consider the one output MVL problem. The error function is given by Eq.7 where O_p and T_p represent the *p-th* actual output value and teacher's value corresponding the *p-th* input pattern $(x_1, x_2, ..., x_n)_p$, respectively.

$$E = \sum_p (O_p - T_p)^2 \tag{7}$$

where p is the number of the total input patterns. The variable space of the MVL network consists of the window parameters a, b and the biasing parameter c $(a, b, c \in 0, 1, 2, ..., R-1)$. Thus, if we define a vector V whose elements include all these parameters:

$$V = \{a_{11}, a_{12}, ..., a_{ij}, b_{12}, ...b_{ij}, ..., c_1, c_2, ...\}^T \tag{8}$$

the error function E can be expressed as:

$$E = E(V) \tag{9}$$

Then, we can iteratively adjust V to minimize the error function $E(V)$. First, the search starts at an initial point and moves along one of d directions. d denotes the number of elements of the vector V. Then the *l-th* direction e^l can be defined as:

$$e^l = (0, ..., 0, \overset{l}{1}, 0, ..., 0)^T \tag{10}$$

The sequence of iterations V_0, V_1 can be described as follows. For the $k-th$ iteration, ($k \geq 0$ when $K = 0$, the initial iterate V_0 and the step size Δ_0 are given) a positive change Δ_k^+ in a direction e_k^l with a positive step Δ_k

$$\Delta V_k^+ = \Delta_k e_k^l \tag{11}$$

when $V_k^l \neq R-1$, (V_k^l is the $j-th$ element in V), results in a change ΔE^+ in E as:

$$\Delta E^+ = E(V_k + \Delta_k e_k^l) - E(V_k) \tag{12}$$

Similarly, a negative change ΔV_k^- in a direction e_k^l with a positive step Δ_k

$$\Delta V_k^l = -\Delta_k e_k^l \tag{13}$$

when $V_k^l \neq R-1$, results in a change ΔE^- in E as:

$$\Delta E^- = E(V_k - \Delta_k e_k^l) - E(V_k) \tag{14}$$

where Δ_k is a positive constant and usually $\Delta_k = 1$ for all k. Then the following learning rule:

$$V_{k+1} = \begin{cases} V_k + \Delta_k e_k^l & \text{if } \Delta E^+ < 0 \\ & \text{and } \Delta E^+ < \Delta E^- \\ V_k - \Delta_k e_k^l & \text{if } \Delta E^- < 0 \\ & \text{and } \Delta E^- < \Delta E^+ \\ V_k & \text{otherwise} \end{cases} \tag{15}$$

will lead the network to the local minimum of E , and hence, to a solution of the MVL problem. Namely the error function E is always decreased by any parameter change produced in the method.

4. Stochastic Dynamic Local Search for MVL Network

Both the back-propagation-based learning algorithm [17] and the local search-based learning algorithm [18] may lead a convergence to a local minimum. However there is not an efficient way for the learning to reach the global minimum from the local minimum. We propose an improved local search method of solving the local minimum problem and apply it to MVL network learning.

Fig.3 is a conceptual graph of the error landscape with a local minimum and a global minimum. The X-coordinate denotes the state of the network and the Y-coordinate denotes the value of error function. For example, if the network is initialized onto point A. Because of the mechanism of the local search method, the state of network moves towards decrease direction and reaches the local minimum (Point B). If we change the dynamics of the MVL at point A to increase the value of error temporarily, point A can become a new point C. From point C, the network returns to move towards decrease direction and reaches the global minimum point D. By incorporating stochastic dynamics into the original local search method, we propose a new algorithm that permits error increase temporarily, which helps MVL escape from the local minimum. The learning rule (Eq.(15)) is modified as follows:

$$V_{k+1} = \begin{cases} V_k + \lambda(t)\Delta_k e_k^l & \text{if } \Delta E^+ < 0 \\ & \text{and } \Delta E^+ < \Delta E^- \\ V_k - \lambda(t)\Delta_k e_k^l & \text{if } \Delta E^- < 0 \\ & \text{and } \Delta E^- < \Delta E^+ \\ V_k & \text{otherwise} \end{cases} \tag{16}$$

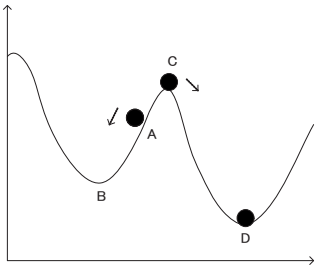

Fig. 3 Conceptual graph of proposed method

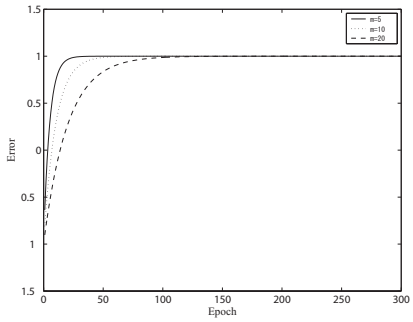

Fig. 4 The characteristic graph of $h(x)$

where $\lambda(t)$ is given by:

$$\lambda(t) = \lfloor random(h(t), 1) \rfloor \tag{17}$$

and $h(t) = 1 - 2e^{-t/m}$, the function $random(a, b)$ returns a stochastic value between a and b. The function $\lfloor x \rfloor$ removes the fractional part of x and returns the resulting integer value. Furthermore, if x is negative, $\lfloor x \rfloor$ returns the first negative integer less than or equal to x. And the parameter t denotes the iteration number. Fig.4 shows the value of changes with m for $m = 5, 10, 20$ respectively.

As can be seen in the graph above, we find that the value of function will reach the maximum slower and slower with m increases. At the beginning, $\lambda(t)$ appears randomly as 1, 0, -1. Then with calculating time goes, $\lambda(t)$ can only be 1.

Case 1: When $\lambda(t)$ is equal to -1, the sequence of iterations will be selected in the contrary direction of the original local search method. In this condition, the value of the error function will ascend.

Case 2: When $\lambda(t)$ is equal to 0, the selected sequence will stay at the original value. And the value of the error function will be unchanged.

Case 3: When $\lambda(t)$ is equal to 1, the algorithm is the same as the original MVL algorithm.

Thus, the proposed algorithm always permits descent of the error, but ascent of the error is allowed initially and becomes less likely as time goes. Furthermore, the algorithm has further feasibility to increase the error with m becomes larger. So, if we select an appropriate m, the proposed algorithm will have powerful ability to reach the global minimum.

Generally, the proposed algorithm can be described as follows:

Step 1. Assign MVL network;

Layer the network into literal, MIN and MAX, a three layered network as in Fig.1.

Step 2. Initialize all the parameters;

Set all window and biasing parameters to random values in $0, 1, 2, ..., R-1$;

Set the maximum iteration times(Max_Epoch), the number of cells in the second layer(MAX_MIN) and m.

Step 3. Present input and desired outputs;

Present all possible multiple-valued input patterns $(x_1, x_2, ..., x_n)_p$ and specify their corresponding desired outputs T_p, where $p = 1, 2, ..., P$.

Step 4. Calculate actual outputs;

Use the multiple-valued operators and formulas to calculate every actual output $(O_1, O_2, ..., O_p$ corresponding to every input vector $(x_1, x_2, ..., x_n)_p$, where $p = 1, 2, ..., P$

Step 5. Adapt windows and biasing parameters;

Use the proposed learning rule $Eq.(16)$ to adapt the parameters.

Step 6. Repeat by going to step 3, until the window and biasing parameters stabilize.

5. Simulation Results

In this section, we present simulation results from the application of the proposed learning algorithm and traditional algorithms to a number of multiple-valued logic functions.

The first example [18] is a two-variable (x_1, x_2), 4-valued function shown in Table 1. A canonical realization of the function can be the summation of the 11 terms:

$$\begin{aligned} F(x_1, x_2) = {} & 1 \cdot x_1(0,0)x_2(0,0) + 1 \cdot x_1(0,0)x_2(1,1) \\ & + 1 \cdot x_1(1,1)x_2(1,1) + 1 \cdot x_1(3,3)x_2(0,0) \\ & + 1 \cdot x_1(3,3)x_2(1,1) + 1 \cdot x_1(3,3)x_2(2,2) \\ & + 1 \cdot x_1(1,1)x_2(3,3) + 2 \cdot x_1(3,3)x_2(3,3) \\ & + 3 \cdot x_1(0,0)x_2(2,2) + 3 \cdot x_1(1,1)x_2(2,2) \\ & + 3 \cdot x_1(2,2)x_2(2,2) \end{aligned} \tag{18}$$

We constructed an MVL network with 40 Literal gates, 20 MIN gates and 1 MAX gate as shown in Fig.5. The window parameters and biasing parameters were initialized randomly from 0 to 3. We used the stochastic dynamic local search algorithm(SDLS) to train the network to learn the MVL function of Table 1. The parameters were set as $Max_Epoch = 1000$,$m = 10$ and simulations were implemented in Visual Basic 6.0 on a Pentium4 2.8GHz (1GB)). The learning curve is shown in Fig.7. And target function which can be got by algebraic methods [44] is given in Equation 19. As can be seen, the MVL network could learn the MVL

Table 1 Example of a quaternary function

X_2 \ X_1	0	1	2	3
0	1	0	0	1
1	1	1	0	1
2	3	3	3	1
3	0	1	0	2

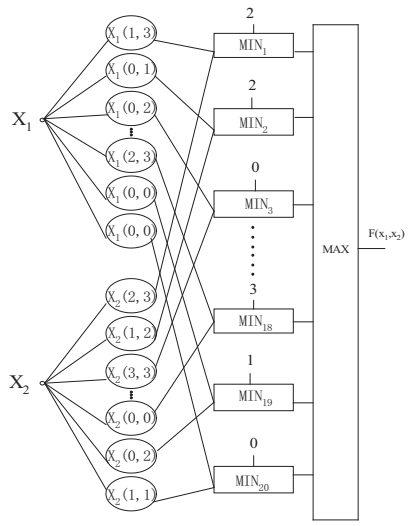

Fig. 5 The MVL network of Table 1 before learning

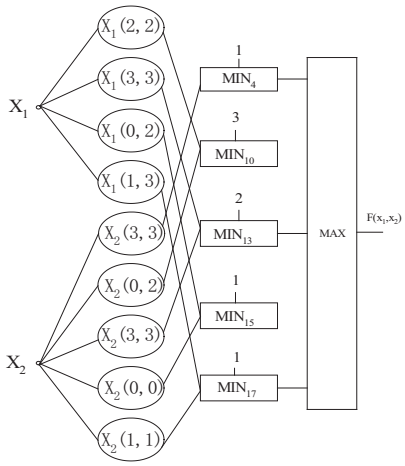

Fig. 6 The final MVL network after learning

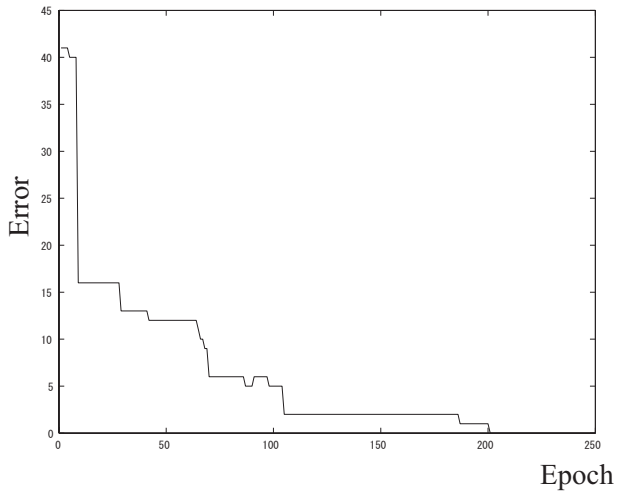

Fig. 7 Learning curve of 2-variables 4-valued function

function successfully. After learning, the network was adapted to a small network (Fig.6) from the initialized network(Fig.5). That is to say that the function(Eq.18) can be simplified to

$$\begin{aligned} F(x_1, x_2) &= 1 \cdot x_2(3,3) + 3 \cdot x_1(2,2)x_2(0,2) \\ &+ 2 \cdot x_1(3,3)x_2(3,3) + 1 \cdot x_1(0,2)x_2(0,0) \\ &+ 1 \cdot x_1(1,3)x_2(1,1) \end{aligned} \tag{19}$$

There was a reduction of 40 literal gates to 9, and 20 MIN gates to 5. It is due to those nodes whose biasing parameters $c_j = 0$ or the window parameters $a_{ij} > b_{ij}$ do not contribute to the output, and therefore can be deleted as shown in Fig. 6.

To compare to the performance of the stochastic dynamic local search(SDLS) algorithm with other well-known back propagation(BP) and local search(LS) algorithms, we used them to learning the same MVL functions. The back propagation was performed on a three layered feedforward neural network. The input size, hidden layer size and output layer size were set to 2, 20, and 1, respectively. The network outputs with 0, 0.33, 0.66 and 1 were considered as 0, 1, 2 and 3 for 4-valued function. The weights and thresholds were initialized randomly from -1 to 1 and sigmoid function was used as the neuron's transfer function. The SDLS and LS used the completely same networks and initial conditions. The maximum iterations were 1000. Fig.8 shows the typical convergence performance of the algorithms. For the BP algorithm, the network gave a large initial error, decreased as the learning was processed and finally converged to a local minimum. The large initial error is due to the BP algorithm's inherent disadvantages of completely randomly generated network and its initial parameters in which no knowledge of MVL function is included. Compared to the BP algorithm, both SDLS and LS algorithm produced a relatively small error initially. This is because some knowledge on MVL functions, at least the network, node function and ranges of the initial parameters can be incorporated into the construction of initial MVL network. For the LS algorithm, we can find its error function decreased continuously, and eventually got stuck in a local minimum. Different from the tradition LS algorithm, the SDLS introduced stochastic dynamics into the local search that permits temporary increase of error function, thus resulting in escape from local minima and possible convergence to global minimum or a better solution.

Furthermore, we performed 100 simulations using

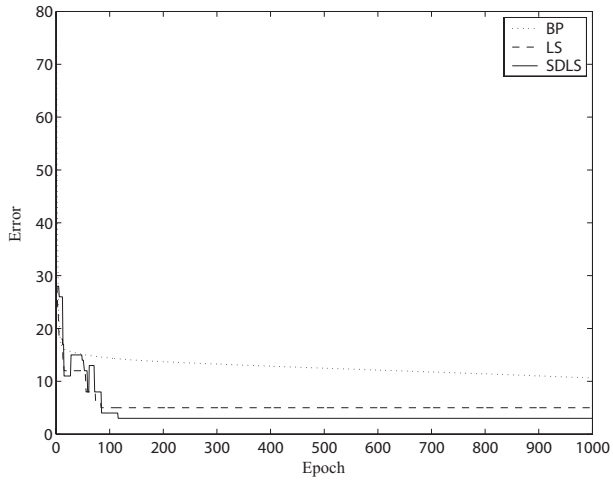

Fig. 8 The comparison of convergence performance among the proposed method($m = 10$), the original local search method and the neural network method on 2-variable 4-valued problem

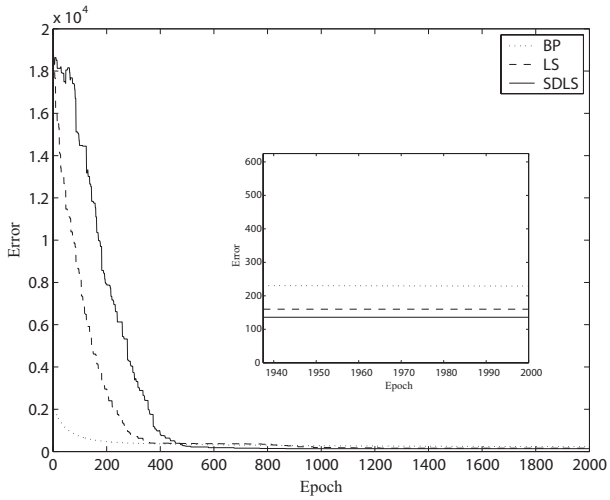

Fig. 9 The comparison of convergence performance among the proposed method, the original local search method and the neural network method on 2-variable 16-valued problem

Table 2 Results of 100 simulations

	Av. Initial	Av.Final	Min.	Av. Time(S)
BP	80	10.91	10.62	25.56
LS	28	4.74	1	3.05
SDLS	28	3.40	0	3.25

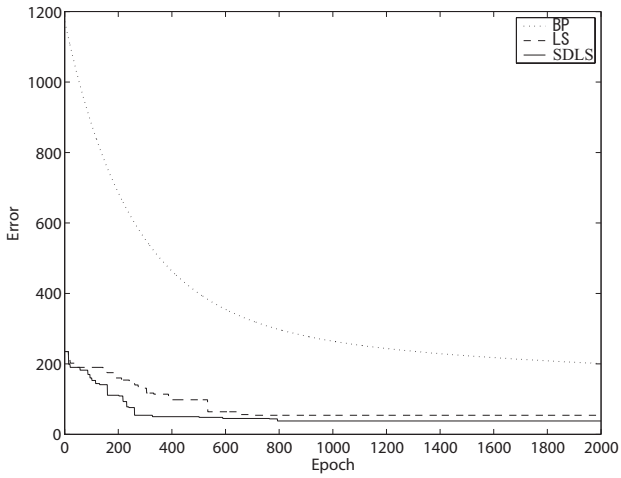

Fig. 10 The comparison of convergence performance among the proposed method, the original local search method and the neural network method on 4-variable 4-valued problem

the three algorithms and compared their performance in Table 2, where "Av." denotes "Average", "Min." denotes "Minimum" and "Av.Time" denotes "Average time (Seconds)". And we can obviously find that both the LS and SDLS algorithms consume less computational time than the BP algorithm.

We have also simulated some different classes of functions [18], such as 2-variable 16-valued functions and 4-variable 4-valued functions and compared them with the back-propagation networks and local search method. Fig.9 and Fig.10 show some typical simulation results of these problems respectively. All the results indicate that our algorithm performed better than the back-propagation algorithm and the original local search method on most of problems.

6. Conclusions

We proposed a local search method with stochastic dynamics for MVL. The proposed method can produce a simplification procession of a multiple-valued function. Furthermore, due to the introduction of stochastic dynamics, the proposed algorithm permits temporary error increases so that it has ability to escape from local minima and eventually reaches the global minimum state or its approximation with very high probability. The learning capability of the MVL network was presented and confirmed by simulations on a large number of 2-variable 4-valued problems, 4-variable 4-valued problems and 2-variable 16-valued problems. The simulation results also showed that the proposed method performed satisfactorily and exhibited good properties and had superior ability for MVL within reasonable number of iterations. In future we plan to investigate the characteristics of a hybrid system of combining Local Search algorithm and Chaos and its application to learning Multiple-Valued Logic Networks.

References

[1] G. Epstein, G. Frieder, and D.C. Rine, "The Development of Multiple-Valued Logic as Related to Computer Science", Computer, vol.7, pp.20-32,1974.

[2] S.L. Hurst, "Multiple-Valued Logic-its status and its future", IEEE Trans. Computers, vol.c-33,no.12,pp.1160-1179,Dec.,1984.

[3] K.C. Smith, "Circuits for multiple-valued logic-A tutorial and appreciation", in Proc.4rd IEEE International Sympo-

sium on Multiple-Valued Logic,pp.30-43,May,1976,.

[4] H.R. Grosch,Signed ternary arithmetic, Digital Computer Lab,MIT Cambridge, Msss, Memorandum M-1496,1954.

[5] M.Stark, Two bits per cell ROM, Digest of Papers of COMPCON Spring 81, pp. 209-212,1982.

[6] S.P. Onnewee and H.G.Kerlhoof, Current-mode CMOS high-radix circuits. Proc, 16th Int. Symp, on Multiple-Valued Logic, pp.60-69, 1986.

[7] E.L Post, "Introduction to a general@theory of elementary propositions," Amer. J. Math., vol.43, pp.163-185,1921.

[8] Z.G. Vranesic, E.S. Lee and K.C. Smith, A many-valued algebra for switching systems, IEEE Trans, Comput. Vol. C-19, No.10, pp.964-972, 1970.

[9] C.M. Allen and D.D. Givone, A minimization technique for multiple-valued logic systems, IEEE Trans. Comput., Vol.C-17, No.2, pp.182-184,1968.

[10] B.A. Bernstein, Modular representation of finite algebras, Proc. 7th Int. Cong. Math, 1924. Vol.1, pp.207-216, Univ. Toronto Press, 1928.

[11] C.Y Lee and W.H Chen, Several Valued combinational switching circuit, Trans, AIEE. Vol.75, pp.278-283,1956.

[12] E.J. McCluskey, Logic design of multiple-valued logic circuits, IEEE Trans, Comput. Vol. C-28, No.8, pp. 546-559,1979.

[13] W.C. Kabat and A.S. Wojcik, Automated synthesis of combinational logic using theorem-proving techniques, IEEE Trans, Comput. Vol. C-34, No.7, pp.610-632, 1985.

[14] W. Wojciechowski and A.S. Wojcik, Automated design of multiple-valued logic circuits by automatic theorem proving techniques, IEEE Trans, Comput. Vol. C-32, No.9, pp.785-798,1983.

[15] K-Y Fang and A.S. Wojcik, synthesis of multiple-valued logic functions based on a modular design approach, Proc, 13th Int. Symp. Multiple-Valued Logic, pp.397-407,1983.

[16] T. Watanabe and M. Matsumoto, Design of multiple-valued logic asynchronous sequential circuits using hyperplanes, Proc, 14th Int. Symp. Multiple-valued logic, pp.251-256,1984.

[17] D.E. Rumelhart, G.E. Hinton and R.J.Williams, "Learning representations by back-propagating errors," Nature, vol.323,pp.533-536,1986.

[18] Q.P. Cao, Z. Tang, R.L. Wang and W.G. Wang, "Local Search Based Learning Method for Multiple-Valued Logic Networks", IEICE Trans. Fundamentals, Vol.E86-A,No.7,pp.1876-1884,July 2003.

[19] I.N.Aizenberg and N.N.Aizenberg "Pattern recognition using neural network based on multi-valued neurons", Lecture Notes in Computer Sciense, 1607-II (J.Mira, J.V.Sanches-Andres - Eds.), Springer-Verlag pp.383-392,1999.

[20] I.Aizenberg,E.Mysnikova,M.Samsonova and J.Reintz," Application of the neural networks based on multi-valued neurons to classification of flu images of gene expression patterns", Fuzzy Days, Spriner-Verlag Berlin,pp.29-304,2001.

[21] I.Aizenberg, N.Aizenberg, C.Butakoff and E.Farberov "Image Recognition on the Neural Network Based on Multi-Valued Neurons", Proceedings of the 15-th International Conference on Pattern Recognition, Barcelona, Spain September pp.3-8, 2000, IEEE Computer Society Press, pp.993-996,2000.

[22] N.N.Aizenberg and I.N.Aizenberg "CNN based on multi-valued neuron as a model of associative memory for gray-scale images", Proc. of the 2-d IEEE International Workshop on Cellular Neural Networks and their Applications, Munich, October pp.12-14,1992

[23] N.N.Aizenberg ,I.N. Aizenberg and G.A.Krivosheev "Multi-Valued Neurons: Learning, Networks, Application to Image Recognition and Extrapolation of Temporal Series", Lecture Notes in Computer Science, pp.389-395,1995.

[24] N.N.Aizenberg, I.N.Aizenberg and G.A.Krivosheev "Multi-Valued Neurons: Mathematical model, Networks, Application to Pattern Recognition", Proc. of the 13 Int.Conf. on Pattern Recognition, Vienna, August pp.25-30, 1996, Track D, IEEE Computer Soc. Press, pp.185-189,1996.

[25] I.N.Aizenberg and N.N.Aizenberg "Application of the Neural Networks Based on Multi-Valued Neurons in Image Processing and Recognition", SPIE Proceedings, pp.88-97,1998.

[26] I.N.Aizenberg "Neural networks based on multi-valued and universal binary neurons: theory, application to image processing and recognition", Lecture Notes in Computer Sciense, Springer-Verlag pp.306-316,1999.

[27] I.N.Aizenberg ,N.N. Aizenberg and J.Vandewalle "Multi-valued and universal binary neurons: theory, learning, applications", Kluwer Academic Publishers, Boston/Dordrecht /London ,2000.

[28] S.Jankowski , A.Lozowski and M.Zurada "Complex-Valued Multistate Neural Associative Memory", IEEE Trans. on Neural Networks, 7, pp.1491-1496,1996.

[29] H.Aoki and Y.Kosugi "An Image Storage System Using Complex-Valued Associative Memory", Proceedings of the 15 Th International Conference on Pattern Recognition, Barcelona, Spain September 3-8, 2000, IEEE Computer Society Press, vol.2, pp.626-629, 2000.

[30] Z.Tang, O.Ishizuka, Q.Cao and H.Matsumoto, "Algebraic properties of a learning multiple-valued logic network," in Proc.23rd IEEE Internation Symposium on Multiple-Valued Logic,Sacramento,California,pp.196-201,May,1993.

[31] Q.Cao,O.Ishizuka,Z.Tang,H.Matsumoto,"Algorithm and implementation of a learning multiple-valued logic network", in Proc.23rd IEEE Internation Symposium on Multiple-Valued Logic,Sacramento,California,pp.202-207,May,1993.

[32] T.Watanabe,et.al.,"Layered MVL neural networks capable of recognizing translated characters,"in Proc.22th IEEE Internation Symposium on Multiple-Valued Logic, Sendai,Japan,pp.88-95,May,1992.

[33] Y.Hata,T.Hozumi and K.Yamato,"Gate model networks for minimization of multiple-valued logic functions,"in Proc.23th IEEE Internation Symposium on Multiple-Valued Logic,Sacramento,Califonia,pp.29-34,May,1993.

[34] W.Wang and C.Moraga, "Design of multivalued circuits using genetic algorithms", Proc. ISMVL'96, pp.216-221, 1996.

[35] E.Zaitseva, A.Shakirin, "Genetic Algorithms to Minimize a Multiple-Valued Logic Function Represented by Arithmetical Polynomial Form", Proc. of the 3rd Int.Conference on Application of Computer Systems, Szczecin, Poland, pp.397-404,1996.

[36] E.N.Zaitseva, T.G.Kalganova, E.G.Kochergov, "The Logical not-Polynomial Forms to represent Multiple-Valued Functions", Proc. of the 26th Int.Symp. on Multiple-valued Logic, Santiago de Compostela, Spain, pp.302-307,1996.

[37] E.Zaitseva, A.Shakirin, D.Popel, G.Holowinski, "Optimization of Incompletely Specified MVL functions using Genetic Algorithms", Proc. of the Int. Workshop on Design Methodogies for Signal Processing, Zakopane, Poland, pp.101-108, 1996.

[38] C.Erick, "Efficient and Accurate Parallel Genetic Algorithms", Kluwer Academic Pub. 2000

[39] Z.Tang,O.Ishizuka and K.Tanno "A learning Multiple-Valued Logic Network that can Explain Reasoning,"T.IEE Japan,vol.119-C,no.8/9,1999.

[40] Z.Tang,Qi Ping Cao and O.Ishizuka,"A Learning Multiple-Valued Logic Network:Algorithm and Applications,"IEEE Trans.on Computers,Vol.47,No.2,pp.247-250,Feb,1998.

[41] F.Glover and G.A. Kochenberger (Eds.), "Handbook of

Metaheuristics", International Series in Operations Research & Management Science (57), Kluwer, 2003.

[42] E.Aarts and J.K. Lenstra(Eds.), "Local search in combinatorial optimization", Wiley, 1997.

[43] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing", Science, Vol.220, Number.4598, pages 671-680, 1983.

[44] Z.K. Luo, M.Hu and T.H.Chen, "The Theory of Multiple-Valued Logic and Its applications", Scientific Pub.(Bei Jing),1992. (in chinese)

Qi Ping CAO received the B.S. degree form Zhejiang University, Zhejiang , China, an M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China and a D.E. degree from Kanazawa University, Kanazawa, Japan in 1983, 1986 and 2005, respectively. From 1987 to 1989, she was an assistant professor in the Institute of Microelectronics at Shanghai Jiaotong University, Shanghai, China. From 1989 to 1990, she was a research student in Nagoya University, Nagoya, Japan. From 1990 to 2000, she was a senior engineer in Sanwa Newtech Inc., Japan. In 2000, she joined Tateyama Systems Institute, Japan. Her current research interests include multiple-valued logic, neural networks and optimizations.

Shangce GAO received the B.S. degree from Southeast University, Nanjing, China in 2005. Now, he is working toward the M.S. degree at Toyama University, Toyama, Japan. His main research interests are multiple-valued logic and artificial neural networks.

Jianchen ZHANG received the B.S. degree from Toyama University, Toyama, Japan in 2006. Now, she is working toward the M.S. degree at Toyama University, Toyama, Japan. Her main research interests are multiple-valued logic and artificial optimizations.

Zheng TANG received the B.S. degree from Zhejiang University, Zhejiang, China in 1982 and an M.S. degree and a D.E. degree from Tshinghua University, Beijing, China in 1984 and 1988, respectively. From 1988 to 1989, he was an Instructor in the Institute of Microelectronics at Tshinhua University . From 1990 to 1999, he was an associate professor in the Department of Electrical and Electronic Engineering, Miyazaki University, Miyazaki, Japan. In 2000, he joined Toyama University, Toyama, Japan, where he is currently a professor in the Department of Intellectual Information Systems. His current research interests include intellectual information technology, neural networks, and optimizations.

Haruhiko KIMURA received the B.S. degree from Tokyo Denki University, Japan in 1974, a D.E. degree from Tohoku University, Japan in 1979. From 1980 to 1984, he was an associate professor in Kanazawa Women University, Kanazawa, Japan. In 1984, he joined Kanazawa University, Kanazawa, Japan, where he is currently a professor in the Graduate School of Natural Science. His current research interests include optimal code conversion, production system and automaton theory.