퍼지 및 지능시스템학회 논문지 2001, Vol. 11, No. 7, pp. 680-688

Synthesis of GBSB-based Neural Associative Memories Using Evolution Program

Hyuk Cho*, Joo-young Park**, Jong-sub Moon***, and Dai-hee Park*

* H. Cho and D. Park is with the Department of Computer Science, Korea University, Chochiwon, Chungnam, 339-700, Korea.

** J. Park is with the Department of Control and Instrumentation Engineering, Korea University, Chochiwon, Chungnam, 339-700, Korea.

*** J. Moon is with the Department of Electronic and Information Engineering, Korea University, Chochiwon, Chungnam, 339-700, Korea.

Abstract

In this paper, we propose a reliable method for searching the optimally performing generalized brain-state-in-a-box (GBSB) neural associative memory using an evolution program (EP) given a set of prototype patterns to be stored as stable equilibrium points. First, we exploit some qualitative guidelines necessary to synthesize the GBSB model. Next, we parameterize the solution space utilizing the limited number of parameters to represent the solution space. Then, we recast the synthesis of GBSB neural associative memories as two constrained optimization problems, which are equivalent to finding a solution to the original synthesis problem. Finally, we employ an evolution program (EP), which enables us to find an optimal set of parameters related to the size of domains of attraction (DOA) for prototype patterns. The validity of this approach is illustrated by a design example and computer simulations.

Key words : Associative memory, Generalized brain-state-in-a-box (GBSB), Parameterization, Evolution program (EP), GBSB(generalized BSB), Domains of attraction (DOA)

1. Introduction

Brain-state-in-a-box(BSB) has been known as the best performance among the various kinds of promising neural associative models. This model was originally proposed by Anderson et al. in 1977 [anderson77], and has been regarded as particularly suitable for implementing associative memories. Its theoretical aspects, especially stability issues, are now well analyzed: Cohen and Grossberg [cohen83] proved a theorem on the global stability of the continuous-time continuous-state BSB with real symmetric weight matrices. Golden [golden86] showed that all trajectories of the discrete-time continuous-state BSB with real symmetric weight matrices approach a set of equilibrium points under certain conditions. Marcus and Westervelt [marcus89] also reported a related result for a large class of discrete-time continuous-state BSB type systems. Perfetti [perfetti95] analyzed qualitative properties of the BSB model, and formulated the design of the BSB-based associative memories as a constrained optimization in the form of a linear programming with an additional nonlinear constraint. He also proposed an ad hoc iterative algorithm to solve the constrained optimization.

An advanced BSB model, which is called GBSB, was proposed and studied by Hui and Zak [hui92], and is now considered to be more appropriate for realizing associative memories than the BSB model in several respects [lillo94][chan97]. The GBSB model has been studied extensively as an effective tool for realizing associative memories: Lillo et al. [lillo94] analyzed the dynamics of the GBSB model, and presented a novel synthesis procedure for GBSB-based associative memory. Their procedure utilizes a decomposition of an interconnection matrix, which results in an asymmetric interconnection structure, asymptotic stability of the desired memory patterns, and small numbers of spurious states. Zak et al. [zak96] incorporated the learning and forgetting capabilities into the synthesis method used in [lillo94]. Also, Chan and Zak [chan97], inspired by Lillo et al. [lillo94] and Perfetti [perfetti95], proposed a "designer" neural network for the synthesis of GBSB-based associative memories.

In this paper, we intend to develope a synthesis procedure for associative memory based on the GBSB model. Focusing on the reliable search for optimally performing GBSB neural associative memory, we first exploit some qualitative guidelines to synthesize the GBSB model. Next, on the basis of the weight matrix expressed on Lillo et al. [lillo94], which was system

Manuscript received september 15, 2001 ; revised Nobember 10. 2001.

atically constructed to satisfy several desired properties above explained, we parameterize the solution space utilizing a small number of parameters. In other words, yielding an associative memory with the desired properties of [lillo94], we intend to ease the complexity of the increasing number of parameters to be searched when we deal with large-scale practical problems such as character recognitions. Then, we recast the synthesis of GBSB neural associative memories as two constrained optimization problems, which are equivalent to finding a solution to the original synthesis problem. Finally, as an optimal searcher for the parameters, we employ an evolution program (EP), which has been reported robust for complex optimization problems in which an analytical solution is not directly available.

The paper is organized as follows: In section II, we briefly describe the fundamentals on the GBSB model, stability definitions, and present some qualitative guidelines to synthesize the GBSB model. In section III, we formulate the synthesis of GBSB-based associative memories as two constrained optimization problems (Parameterization I and II) via parameterization of solution space and interpretation of the parameterization. In section IV, we show how to implement the specific evolution programs related to the proposed systems. In section V, with the concrete simulation results from a design experiment, we compare the performance of the GBSB design using the proposed method with the associative memories design using other methods. Finally, in section VI, concluding remarks are given.

2. Background results

Throughout this paper, the following definitions and notation are used: R^n denotes the linear space of real n vectors. I_n denotes the $n\times n$ identity matrix. H_n denotes the closed hypercube $[-1,+1]^n$. Herein a bipolar vector means that every element is either -1 or $+1$, and B_n denotes the set of all bipolar vectors in H_n. $HD(v,v^*)$ denotes the Hamming distance between two vectors $v\in B_n$ and $v^*\in B_n$. For a matrix $V\in R^{n\times m}$, $V^T\in R^{m\times n}$ denotes the transpose of matrix V.

The dynamics of the GBSB model is described by the following state equation:

$$\begin{aligned} v(k+1) &= g[\,v(k)+\alpha W v(k)+\alpha b] \\ &= g[\,(I_n+\alpha W)v(k)+\alpha b], \end{aligned}$$

where $v(k)\in R^n$ is the state vector at time k, $\alpha>0$ is step size, $W\in R^{n\times n}$ is a weight matrix, $b\in R^{n\times 1}$ is a bias vector, and $g:R^n\rightarrow R^n$ is a piece-wise linear saturating function whose i-th component is defined as follows:

$$g_i([v_1\cdots v_i\cdots v_n]^T) = \begin{cases} +1 & \text{if } v_i\geq +1 \\ v_i & \text{if } -1<v_i<+1 \\ -1 & \text{if } v_i\leq -1 \end{cases}$$

A GBSB model is a generalized version of the BSB proposed by Anderson et al. [anderson77], and it differs from the original network for the presence of the bias vector b.

In the discussion on the stability of the GBSB model, we use the following definitions also described in [lillo94] and [haykin94]:

Definition 1. A point $v_e\in R^n$ is an equilibrium point of the GBSB system if $v(0)=v_e$ implies $v(k)=v_e$, $\forall k>0$.

Definition 2. An equilibrium point v_e of the GBSB system is stable if for any $\varepsilon>0$, there exists $\delta>0$ such that

$$\|v(0)-v_e\|<\delta \text{ implies } \|v(k)-v_e\|<\varepsilon,\ \forall k>0.$$

Definition 3. An equilibrium point v_e of the GBSB system is asymptotically stable if it is stable and there exists $\delta>0$ such that

$$\|v(0)-v_e\|<\delta \text{ implies } v(k)\rightarrow v_e \text{ as } k\rightarrow\infty.$$

Definition 4. The GBSB system is globally stable if every trajectory of the system converges to some equilibrium point.

The criteria on the stability of the GBSB model are now well established in [michel91][golden93][lillo94] as follows:

Criteion 1. A vertex v of the hypercube H_n is an equilibrium point of the GBSB system if and only if

$$\left(\sum_{j=1}^{n} w_{ij}v_j+b_i\right)v_i\geq 0,\ \forall i\in\{1,\cdots,n\}.$$

Criterion 2. A vertex v of the hypercube H_n is an asymptotically stable equilibrium point of the GBSB system if

$$\left(\sum_{j=1}^{n} w_{ij}v_j+b_i\right)v_i>0,\ \forall i\in\{1,\cdots,n\}.$$

Criterion 3. The GBSB system is globally stable if the weight matrix W is symmetric and

$$\lambda_{\min}(I+\alpha W)>-1.$$

where $\lambda_{\min}$ means minimum eigenvalue of matrix W.

In general, the design based only on the stability criteria does not result in satisfactory associative memories. Additional guidelines should be provided to address other performance criteria such as the extent of domains of attraction for each stored pattern. Perfetti [perfetti95] proposed guidelines for the BSB system based on the conjecture that the absence of equilibrium points near stored patterns would increase their domains of attraction, and the experimental results therein showed that such strategy was very effective on reducing the number of spurious states as well as on

increasing attraction domains for stored patterns. With the same strategy, the GBSB counterpart of Perfetti's theorem [perfetti95] can be obtained [park99].

Theorem 1. Suppose that $v \in B_n$ is an asymptotically stable equilibrium point of the GBSB system. If $w_{ii}=0$ for $i=1,\cdots,n$, then none of the vertices v^* such that $HD(v,v^*)=1$ is an equilibrium point.

Theorem 2. Suppose that $v \in B_n$ is an asymptotically stable equilibrium point of the GBSB system and that h is an integer in $\{1,\cdots,n\}$. If

$$\left(\sum_{j=1}^{n} w_{ij}v_j + b_i\right)v_i > 2h \max_j |w_{ij}|, \quad \forall i \in \{1,\cdots,n\}, \qquad (1)$$

then none of the vertices v^* satisfying $0 < HD(v,v^*) \le h$ is an equilibrium point.

Remark 1. The zero-diagonal condition of Theorem 1 also guarantees that only binary steady states are to be observed [perfetti95]. In addition, as concluded in [xu96], the zeroing-out the diagonal (i.e., no self-feedback connections) is useful as a general strategy to improve the performance of existing symmetric Hopfield-type neural networks.

Remark 2. Theorem 2 implies that the maximization of left side of inequality formula generally leads to better performance as to the extent of domains of attraction for each prototype pattern and the number of spurious states. Furthermore, when left side of each inequality in Theorem 2 has maximal value and $\max_j |w_{ij}|$ in its right side has minimal value, the value of h that directly controls the attraction domain has maximal value.

Remark 3. The sufficient condition of the Hopfield counterpart of Theorem 2 was also proposed in Theorem 1 of Tao [tao94] and Theorem 2 of Tao et al. [tao95].

3. Solution Space Parameterizations and Formulations

In this section, we explain how to parameterize the solution space of the GBSB model and recast the synthesis problem to constrained optimization problems.

The weight matrix in both [lillo94] and [zak96], which is for linearly independent prototype patterns, provides a good starting point for parameterizing the solution space of the GBSB model, because it was systematically constructed to satisfy several desired properties. The summary of the synthesis algorithm is as follows: Suppose m prototype patterns are linearly independent. Let $V=[v^1 \cdots v^m] \in \{-1,+1\}^{n \times m}$ be the matrix of given prototypes and V^+ denote pseudo-inverse of V (For details, see [golub96]).

- ***ALGORITHM (taken from [zak96]):***

(i) Form the matrix $B=[b \cdots b] \in R^{n \times m}$, where

$$b=\sum_{p=1}^{m} \varepsilon_p v^{(p)}, \quad \varepsilon_p > 0, \quad \forall p \in \{1,\cdots,m=rank(V)\}.$$

(ii) Form the matrix $D \in R^{n \times n}$ such that

$$d_{ii} > \sum_{j=1, j\neq i}^{n} |d_{ij}|, \quad \forall i \in \{1,\cdots,n\}, \text{ and}$$

$$d_{ii} < \sum_{j=1, j\neq i}^{n} |d_{ij}| + |b_i|, \quad \forall i \in \{1,\cdots,n\}.$$

(iii) Form the matrix $\Lambda \in R^{n \times n}$ such that

$$\lambda_{ii} < -\sum_{j=1, j\neq i}^{n} |\lambda_{ij}| - |b_i|, \quad \forall i \in \{1,\cdots,n\}.$$

(iv) Compute

$$W=(DV-B)V^+ + \Lambda(I_n - VV^+). \qquad (2)$$

(v) Apply criterion 1 defined above to all vertices of H_n to identify spurious equilibria.

They provide a heuristic explanation as to yield an interconnection matrix with desired properties, which includes storing all of the desired patterns as asymptotically stable equilibria with very few spurious states, but not automatically storing the negatives of the desired patterns as asymptotically stable equilibria (see [lillo94] and [Zak96] for complete description). Note that the generated weight matrix is asymmetric and this algorithm does not guarantee the global stability of the system.

Unfortunately, this algorithm requires $2n^2+n$ parameters without providing any methods to get optimal values of the parameters. Therefore some concrete guidelines are needed to improve above algorithm when it is applied to large-scale practical problems while preserving the desired properties. From this consideration, we propose the following guidelines by which we can represent the solution space as the limited number of parameters and obtain optimal parameter values.

3.1 First Parameterization and Formulation (Parameterization I)

- ***Parameterization of Solution Space***

As the first step, we pick the bias vector b by the linear combination of prototype patterns as Chan and Żak [chan97] did in order to direct the trajectories towards the desired patterns as follows.

$$b=\sum_{p=1}^{m} \varepsilon_p v^{(p)}, \quad \varepsilon_p = 1, \quad \forall p \in \{1,\cdots,m\}.$$

Next, we substitute the parameter matrices D and Λ with

$$D=\begin{pmatrix}\tau_{11} & & \\ & \ddots & \\ & & \tau_{1n}\end{pmatrix}=[\tau_{11}e_1 \cdots \tau_{1n}e_n],$$

$$\Lambda=-\begin{pmatrix}\tau_{21} & & \\ & \ddots & \\ & & \tau_{2n}\end{pmatrix}=-[\tau_{21}e_1 \cdots \tau_{2n}e_n],$$

respectively. All non diagonal entries are 0 in both D and Λ. Finally, by applying b, D, and Λ to formula (2), we get the weight matrix (Parameterization I):

$$\begin{aligned}W=&\ ([\tau_{11}e_i \cdots \tau_{1n}e_n]V-B)V^+\\&-[\tau_{21}e_1 \cdots \tau_{2n}e_n](I_n-VV^+),\\\text{s.t.}\ \ &0<\tau_{1i}<|b_i|<\tau_{2i},\ \ \forall i\in\{1,\cdots,n\}.\end{aligned}$$

Through this parameterization, we reduce $2n^2+n$ parameters to $2n$ parameters, without losing the desired properties of ALGORITHM.

- ***Formulation of Parameterization***

As Lillo et al. [lillo94] did, we observe the recall phase with the linearly independent prototype patterns as follows:

$$\begin{aligned}WV=&\ (([\tau_{11}e_1 \cdots \tau_{1n}e_n]V-B)V^+\\&-[\tau_{21}e_1 \cdots \tau_{2n}e_n](I_n-VV^+))V\\=&\ ([\tau_{11}e_1 \cdots \tau_{1n}e_n]V-B)V^+V\\&-[\tau_{11}e_1 \cdots \tau_{1n}e_n](I_n-VV^+)V\\=&\ [\tau_{11}e_1 \cdots \tau_{1n}e_n]V-BV^+V\\=&\ [\tau_{11}e_1 \cdots \tau_{1n}e_n]V\\&-[BV^+v^{(1)},\ BV^+v^{(2)},\ \cdots,\ BV^+v^{(m)}].\end{aligned}$$

Note that $\begin{aligned}&([\tau_{11}e_1 \cdots \tau_{1n}e_n]V-B)V^+V\\&=[\tau_{11}e_1 \cdots \tau_{1n}e_n]V-BV^+V\end{aligned}$ and $(I_n-VV^+)V=0$ by Moore-Penrose conditions [golub96] and a left inverse V^+ satisfies $V^+V=I_r$, $r=\text{rank}(V)$. Therefore, for p-th prototype pattern, we have

$$\begin{aligned}Wv^{(p)}=&\ [\tau_{11}e_1 \cdots \tau_{1n}e_n]v^{(p)}-b\\=&\begin{bmatrix}\tau_{11}v_1^{(p)}-b_1\\ \tau_{12}v_2^{(p)}-b_2\\ \vdots\\ \tau_{1n}v_n^{(p)}-b_n\end{bmatrix}.\end{aligned}$$

For i-th element of the p-th prototype pattern, we have

$$(Wv^{(p)})_i=\left(\sum_{j=1}^{n}w_{ij}v_j^{(p)}\right)_i=\tau_{1i}v_i^{(p)}-b_i,$$

Therefore formula (2) in Theorem 2 is rewritten as

$$\begin{aligned}&\left(\sum_{j=1}^{n}w_{ij}v_j^{(p)}+b_i\right)v_i^{(p)}\\=&\ (\tau_{1i}v_i^{(p)}-b_i+b_i)v_i^{(p)}\\=&\ \tau_{1i}\ (>0).\end{aligned}$$

Note that τ_{1i} has an important role as a design parameter closely related to the extent of attraction domain. With this result, we establish that maximizing left side of the condition of Theorem 2 means maximizing the parameter τ_{1i}:

$$\begin{aligned}&\text{Maximize}\left(\sum_{j=1}^{n}w_{ij}v_j^{(p)}+b_i\right)v_i^{(p)}\ (>0),\\&\forall i\in\{1,\cdots,n\},\ \forall p\in\{i,\cdots,m\}\\\approx&\ \text{Maximize}\ \ \tau_{1i}\ (>0)\quad \forall i\in\{1,\cdots,n\}\end{aligned}$$

Finally, the results allow us to find the GBSB performing optimally by solving the first constrained optimization problem (Formulation I):

Find $\tau_{1i},\cdots,\tau_{1n}$ and $\tau_{21},\cdots,\tau_{2n}$ which maximize h

$$\text{s.t.}\ \left(\sum_{j=1}^{n}w_{ij}v_j^{(p)}+b_i\right)v_i^{(p)}>2h\max_j|w_{ij}|,\quad \forall i\in\{1,\cdots,n\},\ \forall p\in\{1,\cdots,m\},\tag{3}$$

$$W=([\tau_{11}e_i \cdots \tau_{1n}e_n]V-B)V^+-[\tau_{21}e_1 \cdots \tau_{2n}e_n](I_n-VV^+),\tag{4}$$

$$0<\tau_{1i}<|b_i|<\tau_{2i},\ \ \forall i\in\{1,\cdots,n\},\tag{5}$$

$$w_{ii}=0,\ \forall i\in\{1,\cdots,n\}.\tag{6}$$

Each constraint of this optimization problem plays a respective role in the following aspects: The inequalities (3) are the sufficient condition for the given prototype patterns to be stored as asymptotically stable equilibria. Note that this condition is similar to that by Chan and Żak [chan97]. Roughly speaking, the larger h, the wider the attraction domain of each prototype patterns. Thus, the maximal h must be sought under maximal left side (i.e., τ_{1i} for every i) and minimal right side (i.e., $\max_j|w_{ij}|$ for every i) of inequality. Both (4) and (5) are come from parameterization of the solution space. As mentioned in [lillo94], these guidelines yield the desired properties including the asymmetry of weight matrix, no automatic storage of negative prototype patterns as asymptotically stable equilibria, and provisions to minimize the number of spurious states. The zero diagonal condition (6) guarantees no other equilibria in close proximity (i.e. $HD=1$) of the prototype patterns (Theorem 1).

The induced optimization problem satisfies the desired properties with only $2n$ parameters. The number of parameters is still sufficient to represent the solution space and now appropriate to be searched. Additionally, we step forward to propose another parameterization in which the conspicuous difference is in the shape and the number of parameter, which is dramatically reduced to only two parameters.

3.2 Second Parameterization and Interpretation (Parameterization II)

- ***Parameterization of Solution Space***

With the same strategy of parameterization I, we pick the bias vector b by the linear combination of prototype patterns. Next, we substitute the parameter matrices D and Λ with

$$D=\tau_1 I_n,$$

$$\Lambda=-\tau_2 I_n,$$

respectively. Finally, by applying b, D, and Λ to formula (2), we get the weight matrix (Parameterization II),

$$W=(\tau_1 V-B)V^+ - \tau_2(I_n - VV^+),$$
$$\text{s.t. } 0<\tau_1<\min_i|b_i|,\ \max_i|b_i|<\tau_2.$$

- ***Formulation of Parameterization II***

Without loss of generality, by applying the same procedure of parameterization I, we have i-th element of the p-th prototype pattern as,

$$(Wv^{(p)})_i=\tau_1 v_i^{(p)}-b_i.$$

Therefore, the left side of Theorem 2 is to be recalled as

$$\begin{aligned}&\left(\sum_{j=1}^{n} w_{ij}v_j^{(p)}+b_i\right)v_i^{(p)}\\ =\ &(\tau_1 v_i^{(p)}-b_i+b_i)v_i^{(p)}\\ =\ &\tau_1\ (>0).\end{aligned}$$

As parameterization I, we establish

$$\begin{aligned}\text{Maximize}&\left(\sum_{j=1}^{n} w_{ij}v_j^{(p)}+b_i\right)v_i^{(p)}\ (>0),\\ &\forall i\in\{1,\cdots,n\},\ \forall p\in\{i,\cdots,m\}\\ \approx\ &\text{Maximize}\ \ \tau_1(>0)\end{aligned}$$

Finally, the background results allow us to find the GBSB performing optimally by solving the second constrained optimization problem (*Formulation II*):

Find τ_1 and τ_2 which maximize h

$$\begin{aligned}\text{s.t. }&\left(\sum_{j=1}^{n} w_{ij}v_j^{(p)}+b_i\right)v_i^{(p)}>2h\max_j|w_{ij}|,\\ &\forall i\in\{1,\cdots,n\},\ \forall p\in\{1,\cdots,m\},\\ &W=(\tau_1 V-B)V^+-\tau_2(I_n-VV^+),\\ &0<\tau_1<\min_i|b_i|,\ \max_i|b_i|<\tau_2,\\ &w_{ii}=0,\ \forall i\in\{1,\cdots,n\}.\end{aligned}$$

Until now, by parameterizing the solution space of a known algorithm for the synthesis of GBSB-based associative memories, we have represented the solution space which was originally represented with $2n^2+n$ parameters as $2n$ and two parameters (Parameterization I and II). In addition, we have recast the synthesis problem into two constrained optimization problems (Formulation I and II).

4. Our Evolution Program

In this section, we briefly review Genetic Algorithms(GAs) and Evolution Programs(EPs) and show how to employ EPs as a optimizer for our optimization problems.

GAs are randomized multi-directional search algorithms based on the mechanics of natural selection and natural genetics [goldberg89]. They maintain a population of potential solutions and encourage information formation. Individuals are selected naturally; two or more individuals are stochastically crossed over and some are mutated to be effectively implemented parallel multipoint search, which distinguishes them from such classical search methods as hill climbing, random search, and simulated annealing et al. [mitchell96]. Holland [holland75] gave a theoretical framework of GAs. Later, De jong [dejong75] demonstrated GA's applicability to solve optimization problems, and numerous researchers have worked on modifications and improvements in theoretical and empirical aspects. With their computational simplicity, no fundamental limit about the search space has been reported especially suitable for complex scientific and engineering problems finding analytical solutions and global minima is difficult with classical analytical methods. These advantages make GAs become growing and promising fields. Classical GAs, which operate on fixed-length binary strings as data structures and only two genetic operators including binary crossover and mutation, require modifying an original problem in chromosome structure and taking care of decoders and repair algorithms. Whereas evolution programs (EPs), perceived as a generalization of GAs, incorporate problem-specific knowledge by using "natural" representation of potential solution for a given problem and problem-sensitive "genetic" operators [michalewicz95].

The flexibility of EPs due to a real value representation with a combination of applicable "genetic" operators might play an effective role in deriving the solution of our problems. We adapt following components of EPs in order to solve the two optimizations which we recast in the previous section.

4.1 Representation and initial population

For parameterization I, we can directly have each parameter(i.e., an allele of the chromosome), τ_{1i} and τ_{2i}, to satisfy the given constraint, $0<\tau_{1i}<|b_i|<\tau_{2i}$. However, to make implementation easy, we use the following real vector representation of chromosome,

$$T=<T_{1i},\cdots,T_{1n},T_{21},\cdots,T_{2n}>,$$

where each random real number $T_{1i}\in(0,1)$ and $T_{2i}\in(0,1)$ for all $i\in\{1,\cdots,n\}$. With this representation, the initial population of chromosome is generated to a given size. Its original value is obtained by $\tau_{1i}=|b_i|T_{1i}$, $\tau_{2i}=(limit-|b_i|)T_{2i}+|b_i|$ with a given margin *limit* satisfying $\tau_{2i}<limit$ for all i. (Note that we use $limit=10$.)

We also apply the same strategy of parameterization I to parameterization II. The chromosome for parameterization II is

$$T=<T_1,T_2>,$$

where $T_1\in(0,1)$ and $T_2\in(0,1)$. As similar to

parameterization I, its original value is by $\tau_1 = \min_i |b_i| T_1$, $\tau_2 = (limit - \max_i |b_i|) T_2 + \min_i |b_i|$.

4.2 Performance criterion function

Although EPs provide many opportunities for obtaining a global optimal solution., their performance is deterministic depending on the fitness function given by a system designer. With the core idea in subsection 3.1 that the maximal h must be sought under maximal left side (i.e., τ_{1i} for every i) and minimal right side (i.e., $\max_j |w_{ij}|$ for every i) of corollary 2, we can define the performance criterion function for the formulation I as follows:

$$\min J(W) = \sum_{i=1}^{n} (h - l_i) \cdot G(l_i - h),$$

where $l_i \equiv \frac{\tau_{1i}}{2\max_j |w_{ij}|}$, $G(x) = 0$ for $x \geq 0$, and $G(x) = 1$ for $x < 0$. Note that since we employ EPs' parallel multipoint search mechanism, unlike many optimization algorithm, we need not any gradient information of this criterion function. Furthermore, the trivial global minimum (i.e., $J(W) = 0$) can be obtained when $l_i \geq h$ for all i.

For parameterization II, we can define similar criterion function. Note here that only one difference between parameterization I and II is $l_i \equiv \frac{\tau_1}{2\max_j |w_{ij}|}$.

4.3 Genetic Operators and parameter setting

We combine the following genetic operators so as to acquire exploration of the solution space in early step and exploitation of the solution space in late step. To minimize positional bias and endpoint effect, we use two-point crossover operator [mitchell96]. In addition, to improve the fine local tuning capabilities of a GA, we use non-uniform mutation operator defined as follows.

if $S_v^* = \langle v_1, \cdots, v_m \rangle$ is a chromosome and the element v_k was selected for this mutation (domain of v_k is $[l_k, u_k]$), the result is a vector $S_v^{t+1} = \langle v_1, \cdots, v_k', \cdots, v_m \rangle$, with $k \in \{i, \cdots, n\}$, and

$$v_k' = \begin{cases} v_k + \Delta(t, u_k - v_k) & \textit{if a random digit is 0,} \\ v_k - \Delta(t, v_k - l_k) & \textit{if a random digit is 1.} \end{cases}$$

where the function $\Delta(t, y)$ returns a value in the range $[0, y]$ such that the probability of $\Delta(t, y)$ being close to 0 as t increases. This property causes this operator to search the space uniformly initially (when t is small), and very locally at later stages. we have used the following function:

$$\Delta(t, y) = y \cdot \left(1 - r^{(1 - \frac{t}{T})^b}\right)$$

where r is a random number from $[0, \cdots, 1]$, T is a maximal generation number, and b is a system parameter determining the degree of non-uniformity.

This operator provides a mechanism to search the space uniformly initially and very locally at later stages. Note that we use $b = 3$, which is a system parameter determining the degree of non-uniformity. In particular, we employ roulette-wheel selection scheme and elitism to generate offsprings from the previous generation.

Even though the EP's performance and efficiency depend on the selection of the operators and the value of the parameters, the best combination of the operators and parameter settings is still one of open questions in genetic algorithms society. Moreover this issue is out of the scope of this research. So without systematic tests for optimal parameter setting, we follow the parameter settings as De jong [dejong75] and Schaffer et al. [schaffer89] did. For example, Schaffer et al. [schaffer89] found reasonable values, $0.75 \leq P_c \leq 0.95$ for the crossover probability and $0.005 \leq P_m \leq 0.01$ for the mutation probability. In summary, we adapt the following parameter settings: The population size is 30. The maximum generation is 1000. The crossover probability, P_c, is 0.85. The mutation probability, P_m, is 0.01. And lastly, non-uniform degree, b, is 3.

5. Experiments and Results

To show the correctness and performance of the proposed method, we consider a design example. The dimension of the GBSB model to be considered is $n = 10$, and we wish to store the following five prototype patterns which are also cited in [perfetti95] and [chan97]:

$$v^{(1)} = [-1\ +1\ -1\ +1\ +1\ +1\ -1\ +1\ +1\ +1]^T$$
$$v^{(2)} = [+1\ +1\ -1\ -1\ +1\ -1\ +1\ -1\ +1\ +1]^T$$
$$v^{(3)} = [-1\ +1\ +1\ +1\ -1\ -1\ +1\ -1\ +1\ -1]^T$$
$$v^{(4)} = [+1\ +1\ -1\ +1\ -1\ +1\ -1\ +1\ +1\ +1]^T$$
$$v^{(5)} = [+1\ -1\ -1\ -1\ +1\ +1\ +1\ -1\ -1\ -1]^T$$

We directly obtained bias vector b by the linear combination of the prototype patterns as

$$b = [\ +1\ +3\ -3\ +1\ +1\ +1\ +1\ -1\ +3\ +1\ ]^T.$$

First, solving the corresponding optimization problems (i.e., Formulation I and II) along with $\alpha = 0.3$, $h = 1.0$, and $limit = 10$, we obtain the following results:

For Formulation I, we get the parameters, τ_{1i}'s and τ_{2i}'s,

$\tau_{1i} = [$ 0.998829 2.999468 2.998412 0.998925 0.999534
0.999945 0.999710 0.999510 2.999258 0.999193],

$\tau_{2i} = [$ 1.000081 3.003116 3.001702 1.000033 1.001805
1.001154 1.000241 1.001225 3.052658 1.004017].

Note that these parameters satisfy $\tau_{1i} < |b_i| < \tau_{2i}$, $\forall i \in \{1, \cdots, n\}$.

For Formulation II, we get the two parameters, τ_1 and τ_2,

$$\tau_1 = 0.999935,$$
$$\tau_2 = 3.000355.$$

Note again that these parameters satisfy $\tau_1 < \min_i |b_i|$, $\max_i |b_i| < \tau_2$, because $\min_i |b_i| = 1$ and $\max_i |b_i| = 3$. Table I and II show the weight matrix W obtained by Formulation I and II, respectively.

Next, to evaluate the performance of the induced weight of GBSB, we performed simulations for all possible initial binary states, and summarized the information on the domain of attraction for each prototype pattern in Table III and IV for Formulation I and Formulation II, respectively. The entries of the table should be interpreted as follows: "(the entry corresponding to $v^{(p)}$ and $HD = d$) $= s$" indicates that, out of all possible initial binary states with Hamming distance d, s of them converge to the prototype pattern $v^{(p)}$. Obviously, having large entries in the table indicates a desirable feature with respect to the domains of attraction for prototype patterns. For the comparison purpose, we performed the same simulations with $\alpha = 0.3$ for the GBSB of [chan97], and the results are shown in Table V. Note that the entries in the first column of Table III-V are all one, which shows that each of the given prototype patterns is stored as a stable equilibrium point in all three cases. Also, note that entries in Table III and IV are comparable to those in Table V.

To compare the recall capability of these associative memories, we investigated how many initial condition vectors converged to the nearest prototype pattern in the sense of Hamming distance. As shown in Table VI, our system has no spurious states, which are defined as asymptotically stable equilibrium points not corresponding to stored patterns. This result has shown that each trajectory starting from an initial binary state converges to a prototype pattern (Note that the global stability was not explicitly considered in Formulation I and II). In this simulation, about 84 percent of initial binary patterns converged to the closest prototype pattern in our system and about 80 percent did in the [chan97]' system. Although the difference in performance is not outstanding, the result of the proposed method is better. It should be noted that our systems requires only $2n$ and/or two parameters to synthesize associative memories, while Chan and Zak's system requires the whole weight matrix as the solution space (i.e., n^2 parameters).

In addition, the GBSBs proposed in this paper are comparable to other models in the following aspects:

(1) Since the resulted weight is automatically to be asymmetric, we can implement more easily with this parameterization than other structures such as Perfetti [perfetti95], which has symmetric constraints of weight matrices.

(2) Comparatively eigenstructure method [yen95] does heuristic search with only two parameters, so it has small search space and minimize optimization effect. Whereas since our method takes an advantage of EPs' reliable multipoint search mechanism with the appropriate number of parameters satisfying desired properties, the performance improvement is obtained.

(3) Lillo et al. [lillo94] and Perfetti [perfetti95] searched the whole solution space to get optimal associative memories. So the search space is too large to search as the dimension of the system increases. This makes difficulty in implementing large associative memories in practical aspect. Whereas our systems can overcome this problem with $2n$ and/or two parameters.

6. Concluding remarks

In this paper, we have proposed a novel synthesis method for the optimally performing neural associative memories based on the GBSB model. Particularly, the proposed methods have creatively incorporated such synthesis strategies as qualitative analysis of the GBSB model so as to store a set of prototype patterns as stable equilibria, reasonable parameterization of the solution space. Then, we recast of the synthesis problem as constrained optimization problems to apply optimization methods for finding parameter values. By reasonable employment of EPs, we robustly search the design parameter set without any gradient information about the solution space. In addition, we have illustrated a guideline to find the performance index directly related with the extent of the attraction domains, to get the reduced computational complexity for practical application. Through these, we have presented a valuable alternative to the previous heuristic or analytic synthesis methods.

The GBSBs generated by our proposed method have many desirable features: Each prototype pattern can be surely stored as an asymptotically stable equilibrium point; a performance index closely related to the size of domains of attraction for prototype patterns is optimized. Thus large basin of attraction is expected for each prototype pattern; near the stored prototype patterns and, there are no spurious states. A design example was presented to illustrate the proposed method, and the resulting GBSB validated all of the above advantages by outperforming the associative memories designed by other techniques.

References

[1] Anderson, J. A., Silverstein, J. W., Ritz, S. A. and Jones, R. S., 1977, "Distinctive features, categorical perception, and probability learning: some applications of a neural model," Psychological Review, vol. 84, pp. 413-452.

[2] Cohen, M. A. and Grossberg, S., 1983, "Absolute stability of global pattern formulation and parallel memory storage by competitive neural networks", *IEEE Transactions on Systems*, Man and Cybernetics 13, pp. 815-826.

[3] Chan, H. Y. and Zak, S. H., 1997, "On neural networks that design neural associative memories," IEEE Transactions on Neural Networks, vol. 8, pp. 360-372.

[4] De Jong, K. A., 1975, "An analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph.D. dissertation, Department of Computer and Communications Sciences, University of Michigan, Ann Arbor.

[5] Goldberg, D. E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley.

[6] Golden, R. M., 1993, "Stability and optimization analysis of the generalized brain-state-in-a-box neural network model," Journal of Mathematical Psychology, vol. 37, pp. 282-298.

[7] Golub G. H. and Van Loan, C. F., 1996, Matrix Computations, 3rd ed., Baltimore, MD: Johns Hopkins Univ. Press.

[8] Haykin, S., 1994, Neural networks: A Comprehensive Foundation, NJ: Macmillan College Publishing Company, Inc..

[9] Hopfield, J. J., 1982., "Neural networks and physical systems with emergent collective computational abilities," Proceedings of National Academy of Sciences, vol. 79, pp. 2554-2558.

[10] Hui, S. and Zak, S. H., 1992, "Dynamical analysis of the brain-state-in-a-box (BSB) neural models," IEEE Transactions on Neural Networks, vol. 3, pp. 86-100.

[11] Lillo, W. E., Miller, D. C., Hui, S. and Zak, S. H., 1994, "Synthesis of brain-state-in-a-box (BSB) based associative memories," IEEE Transactions on Neural Networks, vol. 5, pp. 730-737.

[12] Marcus, C. M. and Westervelt, R. M., 1989, "Dynamics of iterated-map neural networks," Phaysical Review A 40, pp. 501-504.

[13] Michalewicz, Z., 1994, Genetic Algorithms + Data Structures = Evolution Programs, 2nd edition, Springer-Verlag, New York.

[14] Michel A. N. and Farrell, J. A., 1990, "Associative memories via artificial neural networks," *IEEE Control Systems Magazine*, vol. 10, pp. 6-17.

[15] Michel, A. N., Si, J .and Yen, G., 1991, "Analysis and synthesis of a class of discrete-time neural networks described on hypercubes," *IEEE Transactions on Neural Networks*, vol. 2, pp. 32-46.

[16] Mitchell, M., 1996, An Introduction to Genetic Algorithms, MIT Press.

[17] Park, J. , Cho, H. and Park, D., 1999, "Design of GBSB Neural Associative Memories Using Semidefinite Programming," IEEE Transactions on Neural Networks, vol. 10, no. 4, pp. 946-950.

[18] Perfetti, R., 1995, "A synthesis procedure for brain-state-in-a-box neural networks," *IEEE Transactions on Neural Networks*, vol. 6, pp. 1071-1080.

[19] Schaffer, J. D., Caruana, R. A., Eshelman, L. J. and Da, R., 1989, "A study of control parameters affecting online performance of genetic algorithms for function optimization," in Proc. 3rd Int. Conf. Genetic Algorithms, pp. 51-60.

[20] Wang, T, Zhuang, X., Xing, X. and Xiao, X., 1993, "A Neural-Weighted Learning Algorithm and Its Hardware Implementation in Associative Memories," *IEEE Transactions on Computers*, vol. 42, no. 5, pp. 636-640.

[21] Wang, T., 1994, "Improving Recall in Associative Memories by Dynamic Threshold," Neural Networks, vol. 9, pp. 1379-1385.

[22] Xu, Z. B., Hu, G. Q. and Kwong, C. P., 1996, "Asymmetric Hopfield-type networks: Theory and Applications," Neural Networks, vol. 9, no. 3, pp. 483-501.

[23] Yen, G. G., 1995, "Unlearning algorithm in associative memory," in Proceedings of 1995 World Congress on Neural Networks, pp. 1466-1470, Washington, D.C..

[24] Zak, S. H., Lillo, W. E. and Hui, S., 1996, "Learning and forgetting in generalized brain-state-in-a-box (BSB) neural associative memories," Neural Networks, vol. 9, pp. 845-854.

Appendix

TABLE I

The weight matrix W obtained by Formulation I

$$W=\begin{bmatrix}
0.000 & -0.441 & 0.026 & -0.754 & -0.635 & -0.339 & -0.229 & 0.229 & -0.441 & 0.542 \\
-0.432 & 0.000 & 0.737 & 0.128 & -0.432 & -1.831 & -0.635 & 0.635 & 1.221 & 1.729 \\
-1.144 & 0.407 & 0.000 & 1.042 & -1.144 & -0.610 & 0.788 & -0.788 & 0.407 & -1.424 \\
-0.500 & 0.000 & 0.500 & 0.000 & -0.500 & 0.000 & -0.500 & 0.500 & 0.000 & 0.000 \\
-0.636 & -0.441 & 0.025 & -0.755 & 0.000 & -0.339 & -0.229 & 0.229 & -0.441 & 0.543 \\
0.025 & -0.542 & -0.161 & 0.110 & 0.025 & 0.000 & -0.551 & 0.551 & -0.542 & -0.102 \\
0.042 & -0.237 & 0.398 & -0.483 & 0.042 & -0.644 & 0.000 & -0.415 & -0.237 & -0.169 \\
-0.042 & 0.237 & -0.398 & 0.483 & -0.042 & 0.644 & -0.416 & 0.000 & 0.237 & 0.170 \\
-0.434 & 1.241 & 0.731 & 0.137 & -0.434 & -1.835 & -0.632 & 0.632 & 0.000 & 1.736 \\
0.161 & -0.101 & 0.313 & -0.636 & 0.161 & -0.848 & -0.823 & 0.823 & -0.101 & 0.000
\end{bmatrix}$$

TABLE II

The weight matrix W obtained by Formulation II

$$W=\begin{bmatrix}
0.000 & -0.508 & -0.432 & -1.178 & -1.195 & -0.237 & -0.110 & 0.110 & -0.508 & 0.780 \\
-0.364 & 0.000 & 0.975 & -0.246 & -0.364 & -1.661 & -0.771 & 0.771 & 0.441 & 1.458 \\
-0.687 & 0.644 & 0.000 & 1.025 & -0.687 & 0.034 & 0.873 & -0.873 & 0.644 & -1.254 \\
-0.924 & 0.373 & 0.517 & 0.000 & -0.924 & 0.441 & -0.653 & 0.653 & 0.373 & -0.305 \\
-1.195 & -0.508 & -0.432 & -1.178 & 0.000 & -0.237 & -0.110 & 0.110 & -0.508 & 0.780 \\
0.127 & -0.712 & -0.805 & 0.551 & 0.127 & 0.000 & -0.754 & 0.754 & -0.712 & -0.509 \\
0.161 & -0.102 & 0.314 & -0.636 & 0.161 & -0.847 & 0.000 & -1.178 & -0.102 & -0.644 \\
-0.161 & 0.102 & -0.314 & 0.636 & -0.161 & 0.847 & -1.178 & 0.000 & 0.102 & 0.644 \\
-0.364 & 0.441 & 0.975 & -0.246 & -0.364 & -1.661 & -0.771 & 0.771 & 0.000 & 1.458 \\
0.398 & 0.170 & 0.144 & -0.941 & 0.398 & -1.254 & -1.297 & 1.297 & 0.170 & 0.000
\end{bmatrix}$$

TABLE III
Domains of attraction for Formulation I (GBSB I)

vector \ HD	0	1	2	3	4
$v^{(1)}$	1	10	40	79	83
$v^{(2)}$	1	10	36	78	76
$v^{(3)}$	1	10	37	71	55
$v^{(4)}$	1	8	26	36	30
$v^{(5)}$	1	10	42	79	63

TABLE IV
Domains of attraction for Formulation II(GBSB II)

vector \ HD	0	1	2	3	4
$v^{(1)}$	1	9	28	49	38
$v^{(2)}$	1	10	39	73	78
$v^{(3)}$	1	10	43	66	40
$v^{(4)}$	1	9	42	81	72
$v^{(5)}$	1	10	40	73	61

TABLE V
Domains of attraction for the GBSB model by Chan and Żak [chan97]

vector \ HD	0	1	2	3	4
$v^{(1)}$	1	9	30	58	51
$v^{(2)}$	1	10	38	82	86
$v^{(3)}$	1	10	43	67	43
$v^{(4)}$	1	8	32	67	60
$v^{(5)}$	1	10	41	55	37

TABLE VI
Comparisons of the GBSB models
(A comparison of convergence from initial binary patterns)

Model	# of Spurious Patterns	# of the correct States
GBSB I	0	862
GBSB II	0	859
Chan and Żak [chan97]	0	820

Hyuk Cho
received his BS degree in computer engineering from Chonbuk National University, Korea, in 1994, and his Master degree in computer science from Korea University, Korea, in 1998. He is currently in graduate program for PhD degree in the University of Texas at Austin, Texas. His research interests include text data mining systems.

Joo-young Park
received his BS degree in electrical engineering from Seoul National University, Korea, in 1983, and his PhD degree in electrical and computer engineering from the University of Texas at Austin, Texas, in 1992. He joined Korea University in 1993, where he is currently an Associate Professor in the Department of Control and Instrumentation Engineering. His research interests include neural networks and nonlinear systems.

Jong-sub Moon
received his BS degree in computer sce from Seoul National University, Korea, in 1981, and his PhD degree in computer science from the Illinois Institute of Technology, Illinois, in 1991. He joined Korea University in 1993, where he is currently an Associate Professor in the Department of Electronics and Information Engineering. His research interests include neural networks, computer vision and information security.

Dai-hee Park
received his BS degree in mathematics from Korea University, Korea, in 1982, and his PhD degree in computer science from the Florida State University, Florida, in 1992. He joined Korea University in 1993, where he is currently an Associate Professor in the Department of Computer Science. His research interests include artificial intelligence and intelligence database.