International Journal of Fuzzy Logic and Intelligent Systems, vol. 1, no. 1, June 2001 pp. 35-43

Design of GBSB Neural Network Using Solution Space Parameterization and Optimization Approach

Hy-uk Cho*, Young-hee Im, Joo-young Park***, Jong-sup Moon**** and Dai-hee Park****

*Department of Computer Sciences, The University of Texas at Austin
**Department of Computer Science, Korea University
***Department of Control and Instrumentation Engineering, Korea University
****Department of Electronic and Information Engineering, Korea University

Abstract

In this paper, we propose a design method for GBSB (generalized brain-state-in-a-box) based associative memories. Based on the theoretical investigation about the properties of GBSB, we parameterize the solution space utilizing the limited number of parameters sufficient to represent the solution space and appropriate to be searched. Next we formulate the problem of finding a GBSB that can store the given pattern as stable states in the form of constrained optimization problems. Finally, we transform the constrained optimization problem into a SDP(semidefinite program), which can be solved by recently developed interior point methods. The applicability of the proposed method is illustrated via design examples.

Key Words : Brain-state-in-a-box, Associative memories, Solution space parametrization, Semidefinite program

I. Introduction

Reinvigoration of neural associative memories have been credited to Hopfield [1], who showed how fully interconnected feedback neural networks, trained by the Hebbian learning rule, can associatively recall stored binary patterns. Since then, several neural network models have been successfully introduced to synthesize suitable associative memories [2]. There have also been many studies on how well they perform as associative memories. In general, the desirable characteristics emphasized in the performance evaluation of given neural associative memories include the following [2-4]: an asymptotic stability of each prototype pattern; a minimal number of spurious states; a non-symmetric interconnection structure; the ability to control the extent of the basin of attraction; an incremental learning and forgetting capability; a high storage and retrieval efficiency; a global stability.

Among the various types of promising neural models that show good performance, are the so-called BSB (brain-state-in-a-box). This model was first proposed by Anderson et al. in 1977 [5], and has been regarded as particularly suitable for implementing associative memories. Its theoretical aspects, especially stability issues, are now well documented: Cohen and Grossberg [6] proved a theorem on the global stability of the continuous-time continuous-state BSB dynamical systems with real symmetric weight matrices. Golden showed that all trajectories of the discrete-time continuous-state BSB dynamical systems with real symmetric weight matrices approach the set of equilibrium points under certain conditions [7]. Marcus and Westervelt [8] also reported a related result for a large class of discrete-time continuous-state BSB type systems. Perfetti [9], inspired by Michel et al. [10], 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 non-linear constraint. Also, he proposed an ad hoc iterative algorithm to solve the constrained optimization. In this paper, we intend to develop a synthesis procedure for associative memories based on an advanced form of the BSB model, which is often referred to as GBSB (generalized BSB). The GBSB model was proposed and studied by Hui and Zak [11], and is now considered to be more appropriate for realizing associative memories than the BSB model in several respects [3,12]. The GBSB model has been studied extensively as effective tools for realizing associative memories: Lillo et al. [3] analyzed the dynamics of the GBSB model, and presented a novel synthesis procedure for GBSB-based associative memories. Their procedure utilizes a decomposition of interconnection matrix. This results in asymmetric interconnection structure, asymptotic stability of the desired memory patterns, and small number of spurious states. Zak et al. [13] incorporated the learning and forgetting capabilities into the synthesis method in Lillo's paper [3]. Also, Chan and Zak [12], inspired by Lillo et al. [3] and Perfetti [9], proposed "designer" neural network for the synthesis of GBSB-based associative memories.

Focusing on the reliable search for optimally performing GBSB neural associative memories, we first

Maniscript received March 15, 2001; revised May 1, 2001.
This work was supported by Korea Research Foundation Grant (KRF-99-041-E00266 E2304)

exploit some qualitative guidelines to synthesize the GBSB model. Next, on the basis of the weight matrix expression of Lillo et al. [3], which was systematically constructed to satisfy several desired properties written above, we parameterize the solution space utilizing the limited number of parameters. Consequently, yielding an associative memory with desired properties of [3], we intend to ease the increasing complexity along with increasing the number of parameters in dealing with large-scale practical problems. Next, 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 convert the optimization problems into SDPs (semidefinite programs), which consist of a linear objective and constraints in the form of LMIs (linear matrix inequalities). Since efficient interior point algorithms are now available to solve SDPs (i.e., to find the global optimum of a given SDP efficiently within a given tolerance or find a certificate of infeasibility) [14-16], recasting the synthesis problem to a SDP is equivalent to finding a solution to the original problem. In this paper, we use MATLAB LMI Control Toolbox [15] as an optimum searcher to solve the synthesis problem formulated as an SDP. The best part of these strategies is to recast the synthesis problem to SDPs, which is equivalent to finding a solution to the original synthesis problem, because the global optimum can be efficiently found by sophisticated convex optimization algorithms such as interior-point methods.

This paper is organized as follows: In section II, we briefly summarize the fundamentals on GBSB model, stability definitions, already-known results and present some newly-developed 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 parameterizing solution space and interpreting the parameterizations; we show how to recast the synthesis problems to SDPs. In section IV, with the concrete simulation results from the design experiment, we compare the performance of the GBSB designed by the proposed method with the associative memories designed by other methods. Finally, in section V, the concluding remarks are given.

II. Background Results

Throughout this paper the following definitions and notation are used: R^n denotes the normed linear space of real n vectors with the Euclidean norm $||\cdot||$. For a symmetric matrix $W\in R^{n\times n}$, $||W||$ denotes the induced matrix norm defined by $\max_{x\neq 0}||Wx||/||x||$. 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 these bipolar vectors in H_n. $HD(v,v^*)$ denotes the usual 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 usual transpose of V.

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

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

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

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

By modifying the activation function to take on values within the closed n-dimensional hypercube H_n, the GBSB model solves the problem that the overall response of positive feedback systems may grow without bound. The GBSB model is a generalized version of the BSB network proposed by Anderson et al. [5], 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 [3,17]:

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 [3,10,18] 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

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

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 [9] 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 [9] can be obtained [19].

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

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

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

Corollary 1. 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

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

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

Proof: Since $\max_j \| w_{ij} \| \le \max_{ij} \| w_{ij} \|$ is immediate, by the definition of max element norm defined as $\|W\|_{\max} \equiv \max_{ij} |w_{ij}|$, which is not matrix norm but one of generalized matrix norms (For details, see [20]), we can rewrite the right side of Theorem 2 as $2h\|W\|$. Subsequently, by applying the matrix norm relationships, $\|W\|_{\max} \le \|W\|_2$, we establish the corollary. Note that the inequality of corollary 1 is a sufficient condition for the validity of the condition of Theorem 2.

Remark 1. The zero-diagonal condition of Theorem 1 also guarantees that only binary steady states are to be observed [9]. In addition, as concluded in [21], the zeroing-out 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 (and Corollary 1) implies that the maximization of the left hand side of the if condition 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 the left hand side of each inequality in Theorem 2 (and Corollary 1) has maximal value and $\max_j |w_{ij}|$ (and $\|W\|$) in its right hand 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 [22] and Theorem 2 of Tao et al. [23].

III. Solution Space Parameterizations, Formulations, and Transformations

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

The expression of the weight matrix in [3] and [13], which are for the case of the 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 [20].).

- ALGORITHM (taken from [13]):

(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

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

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

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

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

(iv) Compute

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

(v) Apply criterion 1 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 (For complete description see [3] and [13]). Note that the resulted weight matrix is asymmetric and this algorithm does not guarantee the global stability of the system.

Unfortunately, this algorithm requires us to search $2n^2+n$ parameters without providing any methods to get optimal values of the parameters. Therefore some concrete guidelines are required to improve the applicability 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 SDP-based Synthesis

3.1.1 Parameterization of Solution Space (Parameterization I)

As the first step, we pick the bias vector b by the linear combination of prototype patterns as Chan and Żak [12] did 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. Finally, by applying b, D, and Λ to ALGORITHM, we get the weight matrix (Parameterization I):

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

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

3.1.2 Formulation of Parameterization (Formulation I)

As Lillo et al. [3] did, we observe the recall phase with the linearly independent prototype patterns as follows:

$(I_n-VV^{+})V=0$ by Moore-Penrose conditions [20] and a left inverse V^{+} satisfies $V^{+}V=I_r$, $r=\text{rank}(V)$. Therefore, for p-th prototype pattern, we have

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

Note that $([\tau_{11}e_1\ \cdots\ \tau_{1n}e_n]V-B)V^{+}V=[\tau_{11}e_1\ \cdots\ \tau_{1n}e_n]V-BV^{+}V$ and

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

For i-th element of the p-th prototype pattern, which is recalled as

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

Therefore each left hand side of the inequalities of Corollary 1 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 the left hand side of the condition of Theorem 2 means maximizing the parameter τ_{1i}:

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

$$\approx\ \text{Maximize}\ \tau_{1i}\ (>0)$$

Finally, the background 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

$$\text{s.t.}\quad \tau_{1i}>2h\|W\|,\quad \forall i\in\{1,\cdots,n\}, \qquad \text{(I-1)}$$

$$\begin{aligned}W=\ &([\tau_{11}e_1\ \cdots\ \tau_{1n}e_n]V-B)V^{+}\\ &-[\tau_{21}e_1\ \cdots\ \tau_{2n}e_n](I_n-VV^{+}),\end{aligned} \qquad \text{(I-2)}$$

$$0<\tau_{1i}<|b_i|<\tau_{2i},\quad \forall i\in\{1,\cdots,n\}, \qquad \text{(I-3)}$$

$$w_{ii}=0,\ \forall i\in\{1,\cdots,n\}. \qquad \text{(I-4)}$$

Each constraint of this optimization problem plays a respective role in the following aspects: The inequality (I-1) 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 [12]. Roughly speaking, the larger h, the wider the attraction domain of each prototype patterns. Thus, the maximal h must be sought under maximal left hand side (i.e., τ_{1i} for every i) and minimal right hand side (i.e., $||W||$). Both (I-2) and (I-3) come from parameterization of the solution space. As mentioned in [3], these guidelines yield 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 (I-4) guarantees no other equilibria in close proximity (i.e. $HD=1$) of the prototype patterns (Theorem 1).

This resulted optimization problem satisfies the desired properties with only $2n$ parameters. The number of parameters is still sufficient to represent the solution space and 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.

3.1.3 Transformation of formulation (SDP I)

In this subsection, we establish an SDP-based synthesis procedure for the GBSB neural associative memories by transforming the nonlinear constraint of formulation I into LMI's.

An LMI is any constraint of the form

$$A(z)\equiv A_0+z_1A_1+\cdots+z_NA_N>0 \qquad (*)$$

where $z\equiv[z_1\cdots z_N]^T$ is the variable, and $A_0,\cdots,A_N$ are given symmetric matrices. Since $A(x)>0$ and $A(y)>0$ imply that $A((x+y)/2))>0$, LMI (*) is a convex constraint on the variable z. Note that multiple LMI's $A^{(1)}(z)>0,\cdots,A^{(p)}(z)>0$ can be expressed as the single LMI $\mathrm{diag}(A^{(1)}(z),\cdots,A^{(p)}(z))>0$. Thus, there is no distinction between a set of LMI's and a single LMI. It is well-known that an optimization problem with a linear objective and LMI constraints, which is called a semidefinite program, can be efficiently solved by interior point methods [14-16], and a toolbox of MATLAB for convex problems involving LMI's is now available [15]. Each of the solutions of SDPs considered in this paper was obtained by this toolbox.

The first optimization problem (Formulation I) has both linear and nonlinear constraints. However, the nonlinear constraints can be easily converted to LMI's. For i-th element of the condition (I-1), since $(\tau_{1i})^2>(2h)^2||W||^2$ is equivalent to

$$x^T(\tau_{1i}^2 I_n)x>x^T(2hW)^T(2hW)x,\ \ \forall x\neq 0,$$

this constraint can be reduced to

$$\tau_{1i}I_n-(2hW)^T(\tau_{1i}I_n)^{-1}(2hW)>0.$$

Through Schur complements [14], we get the following LMI, which is equivalent to the above inequality,

$$\begin{bmatrix}\tau_{1i}I_n & 2hW^T\\ 2hW & \tau_{1i}I_n\end{bmatrix}>0.$$

Letting $c\triangleq 1/(2h)$, we rewrite it as

$$\begin{bmatrix}0 & W^T\\ W & 0\end{bmatrix}<c\tau_{1i}\begin{bmatrix}I_n & 0\\ 0 & I_n\end{bmatrix}.$$

Thus, for all elements, we can have

$$\begin{bmatrix}0 & W^T\\ W & 0\end{bmatrix}<c[\tau_{11}e_1\cdots\tau_{1n}e_n]\begin{bmatrix}I_n & 0\\ 0 & I_n\end{bmatrix}.$$

Therefore, the first optimization problem (Formulation I) can be transformed in to the following semidefinite program (SDP I):

With given $c(>0)$, find W such that $\sum_{i=1}^{n}\tau_{1i}$ is maximum,

$$\text{s.t.}\ \begin{bmatrix}0 & W^T\\ W & 0\end{bmatrix}<c[\tau_{11}e_1\cdots\tau_{1n}e_n]\begin{bmatrix}I_n & 0\\ 0 & I_n\end{bmatrix},$$

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

$$0<\tau_{1i}<|b_i|<\tau_{2i},$$

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

3.2 Second SDP-based Synthesis

3.2.1 Parameterization of Solution Space (Parameterization II)

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 ALGORITHM, we get the weight matrix (Parameterization II),

$$W=(\tau_1 V-B)V^+-\tau_2(I_n-VV^+),$$

$$\text{s.t.}\ \ 0<\tau_1<\min_i|b_i|,\ \ \max_i|b_i|<\tau_2.$$

3.2.2 Formulation of Parameterization (Formulation II)

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

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

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

$$\Big(\sum_{j=1}^{n} w_{ij}v_j^{(p)} + b_i\Big)v_i^{(p)} = (\tau_1 v_i^{(p)} - b_i + b_i)v_i^{(p)} = \tau_1 \ (>0).$$

As parameterization I, we establish

$$\text{Maximize}\Big(\sum_{j=1}^{n} w_{ij}v_j^{(p)} + b_i\Big)v_i^{(p)} \ (>0), \quad \forall i \in \{1,\cdots,n\}, \forall p \in \{i,\cdots,m\}$$

$$\approx \text{Maximize } \tau_1(>0)$$

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

$$\begin{aligned} \text{s.t. } & \tau_1 > 2h\|W\|, \\ & W = (\tau_1 V - B)V^+ - \tau_2(I_n - VV^+), \\ & 0 < \tau_1 < \min_i |b_i|, \quad \max_i |b_i| < \tau_2, \\ & w_{ii} = 0, \ \forall i \in \{1,\cdots,n\}. \end{aligned}$$

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).

3.2.3 Transformation of Parameterization II (SDP II)

Without loss of generality, apply the same SDP trasformation procedure of GBSB I. we get the second SDP problem as follows (SDP II):

With given $c(>0)$, find W such that τ_1 is maximum,

$$\begin{aligned} \text{s.t. } & \begin{bmatrix} 0 & W^T \\ W & 0 \end{bmatrix} < c\tau_1 \begin{bmatrix} I_n & 0 \\ 0 & I_n \end{bmatrix}, \\ & W = (\tau_1 V - B)V^+ - \tau_2(I_n - VV^+), \\ & 0 < \tau_1 < \min_i |b_i|, \quad \max_i |b_i| < \tau_2, \\ & w_{ii} = 0, \ \forall i \in \{1,\cdots,n\}. \end{aligned}$$

IV. Experiments and Results

To show the accuracy 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 have stored the following five prototype patterns:

$$v^{(1)} = [-1\ +1\ -1\ +1\ +1\ +1\ -1\ +1\ +1\ +1]^T$$

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

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

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

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

Note that these prototype patterns were considered in [9] and [12] as well. 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 the constant $c=2.86$ and $c=4.72$, respectively, we obtain the following results:

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

$$\tau_{1i} = [\ 0.999999\ 2.999999\ 2.999999\ 0.999999\ 0.999999\ 0.999999\ 0.999999\ 0.999999\ 2.999999\ 0.999999],$$

$$\tau_{2i} = [\ 1.070571\ 3.236817\ 3.918518\ 1.892539\ 1.070571\ 1.835188\ 1.830398\ 1.830398\ 3.236817\ 1.013358].$$

Note that these parameters satisfy

$$\tau_{1i} < |b_i| < \tau_{2i}, \quad \forall i \in \{1,\cdots,n\}.$$

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

$$\tau_1 = 0.999999,$$

$$\tau_2 = 3.292719.$$

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 resulting 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 SDP I and SDP 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 [12], 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 quality 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 systems have 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 the simulations for each SDP of this paper, about 87 and 84 percent of initial binary patterns converged to the closest prototype pattern and about 80 percent in the case of the GBSB of [12]. 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 on the other hand, Chan and Zak's system requires the whole weight matrix as the solution space (i.e., n^2 parameters).

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

(1) Since the resulted weight is automatically asymmetric, we can implement more easily with this parameterization than other structures such as Perfetti [9], which has symmetric constraints of weight matrices.

(2) Eigenstructure method [4] does heuristic search with only two parameters, which result in small search space and lack in fine tuning mechanism. Although SDP II is also required to search two parameters, better performance can be expected through employing systematic synthesis strategies shown in Section III. Moreover, the fine tuning effect by increased parameters results in the performance enhancement, thus the performance of SDP I is somewhat better than that of SDP II.

(3) Lillo et al. [3] computed the weight matrix by choosing not less than $2n^2$ parameters and Perfetti [9] 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 it difficult in implementing large associative memories in practical aspect, whereas our systems can overcome this problem with $2n$ and/or two parameters.

V. 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 to reduce the increasing number of parameters to be searched, formulation of the synthesis problem as constrained optimization problems to apply optimization methods for finding parameter values, and recast of these optimizations into SDPs to find the global optimum efficiently with available interior point algorithms. In addition, we have illustrated a guideline to find the performance index directly related to the extent of the attraction domains, to get the reduced computational complexity apt for practical application. Through these we have presented a valuable alternative to the previous heuristic or analytic synthesis methods.

The GBSBs designed by the 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 attraction basin is expected for each prototype pattern; near the stored prototype patterns, 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] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of National Academy of Sciences*, vol. 79, pp. 2554-2558, 1982.

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

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

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

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

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

[7] R. M. Golden, "The 'brain-state-in-box' neural model is a gradient descent algorithm," *Journal of Mathematical Psychology*, vol. 30, pp. 73-80, 1986.

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

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

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

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

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

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

[14] S. Boyd, L. ElGhaoui, E. Feron, and V. Balakrishnan, Linear Matrix Inequalities in Systems and Control Theory. Philadelphia, PA: SIAM, 1994.

[15] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, MATLAB LMI Control Toolbox, Natick, MA: MathWorks Inc., 1995.

[16] B. Jansen, Interior Point Techniques in Optimization: Complementarity, Sensitivity and Algorithms, Dordrecht, The Nethelands: Kluwer, 1997.

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

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

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

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

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

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

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

Appendix

Table 1. The weight matrix W obtained by SDP I

$$W = \begin{bmatrix} 0.000 & -0.443 & 0.009 & -0.769 & -0.655 & -0.335 & -0.225 & 0.225 & -0.443 & 0.551 \\ -0.440 & 0.000 & 0.709 & 0.171 & -0.440 & -1.851 & -0.620 & 0.620 & 1.313 & 1.761 \\ -1.354 & 0.298 & 0.000 & 1.050 & -1.354 & -0.906 & 0.749 & -0.749 & 0.298 & -1.502 \\ -0.689 & 0.166 & 0.508 & 0.000 & -0.689 & 0.197 & -0.568 & 0.568 & 0.166 & -0.136 \\ -0.655 & -0.443 & 0.009 & -0.769 & 0.000 & -0.335 & -0.225 & 0.225 & -0.443 & 0.551 \\ 0.068 & -0.613 & -0.430 & 0.294 & 0.068 & 0.000 & -0.636 & 0.636 & -0.613 & -0.272 \\ 0.092 & -0.181 & 0.363 & -0.546 & 0.092 & -0.729 & 0.000 & -0.732 & -0.181 & -0.367 \\ -0.092 & 0.181 & -0.363 & 0.546 & -0.092 & 0.729 & -0.732 & 0.000 & 0.181 & 0.367 \\ -0.440 & 1.313 & 0.709 & 0.171 & -0.440 & -1.851 & -0.620 & 0.620 & 0.000 & 1.761 \\ 0.163 & -0.100 & 0.312 & -0.638 & 0.163 & -0.850 & -0.825 & 0.825 & -0.100 & 0.000 \end{bmatrix}$$

Table 2. The weight matrix W obtained by SDP II

$$W = \begin{bmatrix} 0.000 & -0.518 & -0.499 & -1.240 & -1.277 & -0.222 & -0.093 & 0.093 & -0.518 & 0.814 \\ -0.374 & 0.000 & 0.940 & -0.191 & -0.374 & -1.686 & -0.751 & 0.751 & 0.555 & 1.497 \\ -0.753 & 0.609 & 0.000 & 1.028 & -0.753 & -0.060 & 0.860 & -0.860 & 0.609 & -1.279 \\ -0.986 & 0.427 & 0.519 & 0.000 & -0.986 & 0.505 & -0.675 & 0.675 & 0.427 & -0.350 \\ -1.277 & -0.518 & -0.499 & -1.240 & 0.000 & -0.222 & -0.093 & 0.093 & -0.518 & 0.814 \\ 0.142 & -0.737 & -0.899 & 0.615 & 0.142 & 0.000 & -0.784 & 0.784 & -0.737 & -0.568 \\ 0.178 & -0.082 & 0.301 & -0.658 & 0.178 & -0.877 & 0.000 & -1.290 & -0.082 & -0.714 \\ -0.178 & 0.082 & -0.301 & 0.658 & -0.178 & 0.877 & -1.290 & 0.000 & 0.082 & 0.714 \\ -0.374 & 0.555 & 0.940 & -0.191 & -0.374 & -1.686 & -0.751 & 0.751 & 0.000 & 1.497 \\ 0.433 & 0.209 & 0.119 & -0.985 & 0.433 & -1.314 & -1.366 & 1.366 & 0.209 & 0.000 \end{bmatrix}$$

Table 3. Domains of attraction for SDP I

vector \ HD	0	1	2	3	4
$v^{(1)}$	1	8	29	49	39
$v^{(2)}$	1	10	40	79	69
$v^{(3)}$	1	10	43	75	44
$v^{(4)}$	1	10	41	80	72
$v^{(5)}$	1	10	41	75	57

Table 4. IVDomains of attraction for SDP II

vector \ HD	0	1	2	3	4
$v^{(1)}$	1	10	43	80	76
$v^{(2)}$	1	10	39	73	78
$v^{(3)}$	1	10	43	66	40
$v^{(4)}$	1	8	27	50	34
$v^{(5)}$	1	10	40	73	61

Table 5. Domains of attraction for the GBSB model by Chan and Żak [12]

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 6. Comparisons of the GBSB models (A comparison of convergence from initial binary patterns)

Model	# of spurious patterns	# of the correct states
SDP I	0	887
SDP II	0	859
Chan and Žak [12]	0	820

Hy-uk Cho
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.

Young-hee Im
received her BS degree in computer science from Korca University, Korea, in 1994, and her Master degree in computer science from Korea University, Korea, in 1996. She is currently in graduate program for PhD degree in Korea University, Korea. Her research interests include artificial intelligence, information retrieval, text mining.

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 science 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.