Analysis and synthesis of Cohen-Grossberg networks with asymmetric connections

Pengsheng Zheng, Jianxiong Zhang & Wansheng Tang

Institute of Systems Engineering, Tianjin University, Tianjin, 300072, People's Republic of China

Available online: 09 Sep 2011
Analysis and synthesis of Cohen–Grossberg networks with asymmetric connections

Pengsheng Zheng, Jianxiong Zhang* and Wansheng Tang

Institute of Systems Engineering, Tianjin University, Tianjin 300072, People’s Republic of China

(Received 23 December 2010; final version received 3 August 2011)

In this paper, the dynamic behaviours of the asymmetric Cohen–Grossberg neural networks are studied, and some sufficient conditions for the local and global stability of the networks are proposed. Based on the stability results and recently developed system designing method, the networks are constructed for storing and retrieving binary and non-binary patterns, and the network performances are analysed by numerical simulations. It is shown that the designed networks can act as information retrieval systems.

Keywords: Cohen–Grossberg network; asymmetric connection; global stability; local stability; information retrieval

1. Introduction

During the past decades, a lot of recurrent neural networks (RNN) were proposed, some of them served as important tools for modelling the memory retrieval abilities of the brain (Amari 1977; Hopfield 1982; Cohen and Grossberg 1983; Kosko 1988). However, most of these RNNs were studied and designed by imposing strong symmetry hypothesis on the synaptic connections, which appeared to be biologically implausible. In addition, the practical applications of the RNN rely crucially on the analysis of the network dynamic behaviours. For these reasons, the dynamic behaviours analysis and system designing problems of the asymmetric connected RNNs have received considerable attention. Xu, Hu, and Kwong (1996) obtained some sufficient conditions for the discrete-time asymmetric Hopfield networks converging to a stable state. Guo and Huang (2006) studied the stability of the asymmetric Cohen–Grossberg neural networks (ACGNN) with and without delays. Some sufficient conditions for the local and global stability of the asymmetric Hopfield networks and Cohen–Grossberg networks were proposed in Chen and Amari (2001) and Lu and Chen (2003), respectively. In this paper, the dynamic behaviours of the ACGNNs are investigated, and some new sufficient conditions for the local and global stability of the network are proposed.

Traditional Hebb rule and pseudo-inverse rule are incapable of constructing asymmetric neural associative memories which strictly follow the local stability condition of the network. For dealing
with the design of asymmetric networks, a synthesis procedure for designing non-symmetric cellular neural networks was developed in Liu and Michel (1993). Lee and Chuang (2005) proposed a method for the design of asymmetric discrete-time Hopfield networks by using the concept of higher-order Hamming stability. An efficient Schur decomposition method for designing asymmetric Hopfield networks was proposed in Zheng, Tang, and Zhang (2010). In this paper, the ACGNNs are constructed following the Schur decomposition method (Zheng et al. 2010), and a given set of patterns can be easily assigned as locally stable equilibria of the network.

Image retrieval is an important application of the neural associative memories. Costantini, Casali, and Perfetti (2006) proposed a procedure for designing neural associative memories storing grey-scale images using multi-layer neural networks. Vazquez and Sossa (2008) developed a colour image retrieval system based on the hetero associative memory. Oh and Zak (2005) proposed a method for the design of asymmetric discrete-time Hopfield networks by using the generalised Brain-State-in-a-Box. Colour image retrieval in a class of sparsely connected auto-associative morphological memories was studied in Valle (2009). In this paper, the ACGNNs are constructed for storing and retrieving 256 grey-scale images. It is shown that the designed networks can act as image retrieval systems.

The remainder of this paper is organised as follows. In Section 2, the dynamic behaviours of the ACGNNs are investigated by resorting to the energy function method, and some sufficient conditions for the global and local stability of the network are proposed. In Section 3, numerical simulations are developed to illustrate the effectiveness of the proposed results. In addition, the ACGNNs are constructed for the Chinese character recognition and grey-scale image retrieval, and the network performances are studied by numerical simulations.

2. Network model and dynamic analysis

2.1. Model description

Let us consider a Cohen–Crossberg neural network of the following form:

$$
\dot{X} = -A(X)[B(X) - WG(X) + \Theta],
$$

(1)

where $X = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ is the neuron state vector, $\Theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \in \mathbb{R}^n$ is a real constant vector, $W \in \mathbb{R}^{n \times n}$ is the synaptic weights matrix which is asymmetrically defined, $A(X) = \text{diag}(a_1 (x_1), a_2 (x_2), \ldots, a_n (x_n))$, $B(X) = [b_1 (x_1), b_2 (x_2), \ldots, b_n (x_n)]^T$, $G(X) = [g_1 (x_1), g_2 (x_2), \ldots, g_n (x_n)]^T$, $a_i(\cdot)$ is a continuous function with $0 < a_i(x) < +\infty$ for all $x \in \mathbb{R}$, $b_i(0) = g_i(0) = 0$, $b_i(\cdot)$ and $g_i(\cdot)$ are differentiable defined by satisfying the following conditions:

$$
0 < l_i \leq \frac{b_i(x) - b_i(y)}{x - y}, \quad \forall x, y \in \mathbb{R}, \quad i = 1, 2, \ldots, n.
$$

(2)

$$
0 < \frac{g_i(x) - g_i(y)}{x - y} \leq k_i, \quad \forall x, y \in \mathbb{R}, \quad i = 1, 2, \ldots, n.
$$

(3)

where $l_i, k_i \in (0, +\infty)$.

In this paper, network (1) is restricted to have at least one equilibrium point. Here, the existence of the equilibrium point can be guaranteed by using the system designing algorithm introduced in Zheng et al. (2010). Throughout this paper, $X^*$ denotes an equilibrium point of network (1), $f'(x)$ denotes the derivative of $f(x)$ with respect to $x$, $U = [u_1, u_2, \ldots, u_n]^T = X - X^*$,

$$
U^* = X^* - X^* = 0,
$$
In this section, some sufficient conditions for the global and local asymptotical stability of network (1) are proposed by resorting to the energy function method.

**Theorem 2.1** Let $X^*$ be an equilibrium of network (1). If $H < 0$ and one of the following conditions holds:

$$\int_0^\infty \frac{b_i(s)}{a_i(s)} \, ds = \infty, \quad i = 1, 2, \ldots, n, \quad (5)$$

$$\int_0^\infty \frac{g_i(s)}{a_i(s)} \, ds = \infty, \quad i = 1, 2, \ldots, n, \quad (6)$$

then network (1) is globally asymptotically stable, and $X^*$ is the unique equilibrium point.

**Proof** Define a continuously differentiable energy function of network (1) as

$$E(U) = \sum_{i=1}^n \int_0^{u_i} \frac{\tilde{b}_i(s)}{a_i(s + x_i^*)} \, ds + \alpha \sum_{i=1}^n \int_0^{u_i} \frac{\tilde{g}_i(s)}{a_i(s + x_i^*)} \, ds, \quad (7)$$

where $U = [u_1, u_2, \ldots, u_n]^T = X - X^*$, and $\alpha$ is a positive scalar.

It follows from (2) and (3) that, $\tilde{b}_i(\cdot)$ and $\tilde{g}_i(\cdot)$ are monotonically increasing functions, and \(\tilde{b}_i(u), \tilde{g}_i(u) = 0\) if and only if $u = 0$. Note the fact that $a_i(\cdot)$ are positive continuous functions. Hence,

$$\int_0^{u_i} \frac{\tilde{b}_i(s)}{a_i(s + x_i^*)} \, ds \geq 0, \quad i = 1, 2, \ldots, n, \quad (8)$$

$$\int_0^{u_i} \frac{\tilde{g}_i(s)}{a_i(s + x_i^*)} \, ds \geq 0, \quad i = 1, 2, \ldots, n. \quad (9)$$

Thus, $E(U) \geq 0$, and $E(U) = 0$ if and only if $U = U^* = 0$. It follows from (5) or (6) that $E(U) \to +\infty$ as $\|U\| \to \infty$, $E(U)$ is positive definite (except $U = 0$) and radially unbounded. Moreover, if $U = U^* = 0$ then $X = X^*$. This implies that the global minimum point of $E(U)$ corresponds to the equilibrium point $X^*$ of network (1).

It follows from (2) that

$$\frac{\tilde{b}_i(u_i)}{u_i} = \frac{b_i(u_i + x_i^*) - b_i(x_i^*)}{u_i} \geq l_i.$$ 

If $u_i > 0$, then $\tilde{b}_i(u_i) \geq l_i \neq u_i$, it follows that

$$-\tilde{g}_i(u_i)\tilde{b}_i(u_i) \leq -\tilde{g}_i(u_i)l_i u_i.$$
Thus, we have
\[ -\tilde{g}_i(u_i) \tilde{b}_i(u_i) \leq -\tilde{g}_i(u_i) l_i u_i. \]

If \( u_i = 0 \), then
\[ -\tilde{g}_i(u_i) \tilde{b}_i(u_i) = -\tilde{g}_i(u_i) l_i u_i = 0. \]

Then, for any given \( u_i \),
\[ -\tilde{g}_i(u_i) \tilde{b}_i(u_i) \leq -\tilde{g}_i(u_i) l_i u_i. \]

Similarly, by condition (3), for any given \( u_i \),
\[ -\tilde{g}_i(u_i) l_i u_i \leq -\tilde{g}_i(u_i) \frac{l_i}{k_i} \tilde{g}(u_i). \]

Thus, we have
\[ -\tilde{g}_i(u_i) \tilde{b}_i(u_i) \leq -\tilde{g}_i(u_i) \frac{l_i}{k_i} \tilde{g}(u_i), \quad i = 1, 2, \ldots, n, \]
then
\[ -\tilde{G}^T(U) \tilde{B}(U) \leq -\tilde{G}^T(U) \Gamma \tilde{G}(U). \] (10)

Furthermore, for any \( X \in \mathbb{R}^n \), it holds
\[ X^T(W - \Gamma)X + [X^T(W - \Gamma)X]^T = 2X^T(W - \Gamma)X, \]
\[ X^T(W - \Gamma)X + X^T(W - \Gamma)^TX = 2X^THX, \]
then
\[ X^THX = X^T(W - \Gamma)X. \] (11)

By Equation (7), the time derivative of \( E(U)(dE(U)/dt) \) along the trajectory of network (1), denoted by \( \dot{E}(U) \), is given by
\[
\dot{E}(U) = -\tilde{B}^T(U)[\tilde{B}(U) - W\tilde{G}(U)] - \alpha \tilde{G}^T(U)[\tilde{B}(U) - W\tilde{G}(U)]
= -\tilde{B}^T(U)\tilde{B}(U) + \tilde{B}^T(U)W\tilde{G}(U) - \alpha \tilde{G}^T(U)\tilde{B}(U) + \alpha \tilde{G}^T(U)W\tilde{G}(U)
= -\frac{1}{4} \tilde{G}^T(U)W^TW\tilde{G}(U) - \alpha \tilde{G}^T(U)\tilde{B}(U) + \alpha \tilde{G}^T(U)W\tilde{G}(U)
= -[\tilde{B}^T(U) - \frac{1}{2}W\tilde{G}(U)][\tilde{B}(U) - \frac{1}{2}W\tilde{G}(U)] + \frac{1}{4} \tilde{G}^T(U)W^TW\tilde{G}(U)
- \alpha \tilde{G}^T(U)\tilde{B}(U) + \alpha \tilde{G}^T(U)W\tilde{G}(U).
\]

It follows from (10) and (11) that
\[
\dot{E}(U) \leq \frac{1}{4} \tilde{G}^T(U)W^TW\tilde{G}(U) - \alpha \tilde{G}^T(U)\tilde{B}(U) + \alpha \tilde{G}^T(U)W\tilde{G}(U)
\leq \frac{1}{4} \tilde{G}^T(U)W^TW\tilde{G}(U) + \alpha \tilde{G}^T(U)(W - \Gamma)\tilde{G}(U)
= \frac{1}{4} \tilde{G}^T(U)W^TW\tilde{G}(U) + \alpha \tilde{G}^T(U)H\tilde{G}(U). \] (12)

Hence, whenever \( \alpha \) is sufficiently large, if \( H < 0 \), then \( \dot{E}(U) \leq 0 \), and \( \dot{E}(U) = 0 \) if and only if \( \tilde{G}(U) = 0 \), i.e. \( U = U^* = 0 \).
Suppose that network (1) has another equilibrium point $X^+ (X^+ \neq X^*)$ under the condition $H < 0$, and $U^+ = X^+ - X^*$. Then it holds

$$-A(U^+ + X^*)(\tilde{B}(U^+) - W\tilde{G}(U^*)) = -A(X^+)(B(X^+) - WG(X^+) + \Theta) = 0.$$ 

It follows that $\dot{E}(U) = 0$ if $U = 0$ or $U = U^+$, which contradicts with the fact that $\dot{E}(U) = 0$ if and only if $U = U^* = 0$. Hence, if $H < 0$, then network (1) has a unique equilibrium point. Given an arbitrary initial state, as the network dynamic evolves with time, $E(U)$ decreases monotonically until it reaches its global minimum point $U^*$ which also corresponds to the unique equilibrium point $X^*$ of network (1). By the Lyapunov stability criterion, network (1) is globally asymptotically stable. This completes the proof. ■

**Remark 2.2** If $a_i(x) = 1$, $b_i(x) = x$, then network (1) reduces to a Hopfield network, and Theorem 2.1 reduces to the Theorem 2 in Zheng et al. (2010).

The neural associative memory is an important application of the RNN. Theorem 2.1 provides some sufficient conditions for the uniqueness of the equilibrium point. However, neural associative memories usually need more than one stable equilibrium. To endow network (1) with retrieval properties, a given set of patterns need to be assigned as locally asymptotically stable equilibria.

Let $Q = [W - \Lambda(X^*) + (W - \Lambda(X^*))^T]/2$, where $\Lambda(X) = \text{diag}(b'_1(x_1)/g'_1(x_1), b'_2(x_2)/g'_2(x_2), \ldots, b'_n(x_n)/g'_n(x_n))$. Here, $b'_i(x_i)$ and $g'_i(x_i)$ denote the derivative of $b_i(x_i)$ and $g_i(x_i)$ with respect to $x_i$, respectively. The following theorem presents a sufficient condition for the local asymptotically stability of the equilibria.

**Theorem 2.3** If $Q < 0$, then $X^*$ is a locally asymptotically stable equilibrium point of network (1).

**Proof** Construct an energy function as

$$E_1(U) = \sum_{i=1}^{n} \int_{0}^{u_i} \frac{\tilde{g}'_i(s)}{a_i(s + x^+_i)} ds. \quad (13)$$

Following the proof of Theorem 2.1, it can be verified that $E_1(U) \geq 0$, and $E_1(U) = 0$ if and only if $U = U^* = 0$. For $i = 1, 2, \ldots, n$, it holds

$$\tilde{g}'(u_i^*) = \frac{\tilde{g}'(0)}{a_i} = g'_i(x^+_i),$$

$$\tilde{b}'(u_i^*) = \frac{\tilde{b}'(0)}{b_i} = b'_i(x^+_i).$$

Then, the linearisations of $\tilde{G}(U)$ and $\tilde{B}(U)$ at $U^*$ yield

$$\tilde{g}_i(u_i) = u_i g'_i(x^+_i), \quad (14)$$

$$\tilde{b}_i(u_i) = u_i b'_i(x^+_i). \quad (15)$$

Define $\Omega_\epsilon = \{ U : ||U - U^*|| < \epsilon \}$ where $\epsilon$ is sufficiently small. Then, in $\Omega_\epsilon$, it follows from (14) and (15) that

$$-\tilde{G}^T(U)\tilde{B}(U) = -\tilde{G}^T(U)\Lambda(X^*)\tilde{G}(U). \quad (16)$$
It follows from (11), (13) and (16) that, the time derivative of \( E_1(U) \) in \( \Omega_\varepsilon \), denoted by \( \dot{E}_1(U) \), can be represented by

\[
\dot{E}_1(U) = -\bar{G}^T(U)[\bar{B}(U) - W\bar{G}(U)] \\
= -\bar{G}^T(U)\bar{B}(U) + \bar{G}^T(U)W\bar{G}(U) \\
= -\bar{G}^T(U)\Lambda(X^*)\bar{G}(U) + \bar{G}^T(U)W\bar{G}(U) \\
= \bar{G}^T(U)[W - \Lambda(X^*)]\bar{G}(U) \\
= \bar{G}^T(U)\bar{Q}\bar{G}(U). \tag{17}
\]

Note that Equation (17) only holds in \( \Omega_\varepsilon \). If \( Q < 0 \), then, in a certain neighbourhood of \( U^* \), \( \dot{E}_1(U) \leq 0, \dot{E}_1(U) = 0 \) if and only if \( U = U^* = 0 \). Given an initial state which is sufficiently close to \( U^* \), as the network dynamic evolves with time, \( E_1(U) \) keeps decreasing until it converges to \( U^* \) which also corresponds to the equilibrium point \( X^* \) of network (1). According to the Lyapunov stability criterion, \( X^* \) is locally asymptotically stable.

**Remark 2.4** If \( W \) is a symmetric matrix, Theorems 2.1 and 2.3 also hold, with \( H \) and \( Q \) reducing to \( W - \Gamma \) and \( W - \Lambda(X^*) \), respectively.

3. **Numerical simulations**

In this section, based on the stability results and the recently developed system designing method, two numerical examples are given to study the memory retrieval abilities of the designed ACGNNs.

3.1. **Example 1**

In this example, network (1) is constructed for Chinese characters recognition by applying the system designing procedure proposed in Zheng et al. (2010) (for details see Appendix).

Figure 1(a) shows the Chinese characters to be remembered, the characters are interpreted as matrixes with size \( 12 \times 11 \), each one corresponding to a 132 dimensional vector with binary units: \(-1\) (black) and \(1\) (white). Here and later, the transformation of matrix to vector and its inverse

![Figure 1](attachment:Figure_1.png)

Figure 1. Initial input patterns and final corresponding outputs. (a) Patterns need to be remembered. (b) Blurred patterns. (c) Retrieved patterns.
are defined as

$$M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n_1} \end{bmatrix}$$

where \( M \in \mathbb{R}^{n_1 \times m_1} \), \( M_i \) denotes the \( i \)th row of \( M \), and \( n_1, m_1 \) denote the row and column dimension of \( M \), respectively.

Consider a network with \( a_i(x) = \sin(x) + 1.1 \), \( b_i(x) = \arctan(x) \) and \( g_i(x) = \tanh(x) \) \((x \in \mathbb{R}, \; i = 1, 2, \ldots, n)\). The system designing procedure is performed with \( n = 132, \; m = 7, \; c = 1.03, \; \beta_{ij} = -0.5, \; \gamma_{ij} = 0.5 \) and \( \delta_i = -10 \). \( \Delta_{ij} \) \((i > j, \; i \geq 7, \; j \geq 7)\) are set to be random values within \((-0.5, 0.5)\). Then a proper network can be constructed. It can be verified that \( Q < 0 \) for all the remembered patterns, which implies that the given patterns have been assigned the locally asymptotically stable equilibria. In order to design a proper network, it is necessary to keep \( Q < 0 \) for all the remembered patterns. However, one may fail to design such a network. In this case, it is better to re-design the network by decreasing the values of \( \delta_i \).

Let \( X_1, X_2, \ldots, X_7 \) denote the patterns shown in Figure 1(a) from left to right, and \( \hat{H}_d(X^i) = \{X | D(X, X^i) = d\} \) \((j = 1, 2, \ldots, 7)\) be the Hamming shell of \( X^i \), where \( D(X, X^i) \) denotes the Hamming distance between \( X \) and \( X^i \). 30d vectors from every Hamming shell are collected as the initial states of the network and then are tested as to how many of them can be correctly retrieved. The percentage of vectors on the Hamming shell \( \hat{H}_d(X^i) \), which can be recalled to \( X^i \), is denoted by \( P(d) \). Usually, a larger value of \( P(d) \) implies a larger averaged basin width. Figure 2 plots the value of \( P(d) \) with \( d \) varying from 1 to 60 with step 5. It is shown that \( P(d) = 1 \) for \( d \leq 20 \), which implies that, in this case, all the collected vectors on the Hamming shells \( \hat{H}_d(X^i) \) belong to the attractive domain of \( X^i \). As shown in Figure 2, every remembered pattern is an attractor of the network, and the network can finally converge to a stored pattern if the initial state (blurred image) is sufficiently close to it.

Figure 1(b) shows seven blurred characters which are presented to the constructed network as initial states. Figure 1(c) illustrates the retrieved patterns of the corrupted ones in the same

![Figure 2. The value of $P(d)$ as $d$ varies from 0 to 60 with step 5. Each curve corresponds to a stored pattern as shown in the legend.](image-url)
Figure 3. Two hundred and fifty-six grey-scale images in Example 2. (a) Grey-scale images need to be remembered. (b) Blurred images with Gaussian white noise of mean 0.04 and variance 0.01. (c) Corresponding retrieved images of the blurred ones in the same column.

column. As shown in Figure 1(c), the network retrieves a previously stored pattern which most closely resembles the corrupted one, which demonstrates a perfect retrieval reliability.

3.2. Example 2

Consider the 256 grey-scale face images shown in Figure 3. Each image is $40 \times 35$ pixels. Conventionally, the images can be interpreted as data matrices of class unit 8, i.e. the pixel value ranges from 0 to 255. Here, for the convenience of the system designing, the data matrix is transformed by

$$\phi_n = 2 \times \frac{\phi_o}{255} - 1,$$

where $\phi_o$ denotes the original pixel value, $\phi_n \in [-1, 1]$ is the new pixel value. Then the data matrices can be transformed into $40 \times 35$ dimensional vectors by (18).

Consider a network (1) defined with $a_i(x) = 1$, $g_i(x) = \tanh(x)$ and

$$b_i(x) = \begin{cases} \arctan(x) - 0.024 & \text{if } x > 1, x < -1 \\ \tanh(x) & \text{if } -1 \leq x \leq 1 \end{cases}$$

for $i = 1, 2, \ldots, 1400$. The system designing procedure is performed with $n = 1400$, $m = 9$, $c = 1$, $\beta_{ij} = -0.5$, $\gamma_{ij} = 0$, $\delta_i = -6$, and $\Delta_{ij}$ ($i > j$, $i \geq 9$, $j \geq 9$) are set to be random values within $(-0.5, 0)$. Then $W$ and $I$ can be computed.

To examine the performance of the constructed network, nine blurred images, with Gaussian white noise of mean 0.04 and variance 0.01, are presented to the designed network as initial states. As the network dynamic evolves with time, the network finally converges to some stable patterns. To measure the quality of pattern recognition, we adopt the performance function NMSE in Valle (2009) as

$$\text{NMSE} = \frac{\sum_{i=1}^{n}(x_i^j - y_i^j)^2}{\sum_{i=1}^{n}(x_i^j)^2},$$

where $Y^j = [y_1^j, y_2^j, \ldots, y_n^j]^T$ is the final output and $X^j$ denotes the corresponding desired output. Figure 3(b) and (c) shows the blurred and corresponding retrieved images, respectively.
As illustrated in Figure 3, the retrieved images are almost identical with the initial ones. It can be calculated that the average NMSE is 0.002. The constructed network can act as an efficient associative memory for storing 256 grey-scale images.

4. Conclusion

The dynamic behaviours of the ACGNNs are studied, and some sufficient conditions for the local and global stability of the networks are proposed. In addition, the networks are designed for storing and retrieving binary patterns and 256 grey-scale images, and the network performances are analysed by numerical simulations. It is shown that the designed networks can act as efficient information retrieval systems.

Acknowledgements

The authors are very grateful to anonymous reviewers for their helpful comments and suggestions. This work was partly supported by the National Natural Science Foundation of China (No. 61004015), the Research Fund for the Doctoral Programme of Higher Education of China (No. 200900321200340), and the Program for Changjiang Scholars and Innovative Research Team in University of China.

Note


References

Appendix

The system designing algorithm introduced in Zheng et al. (2010).
Let \( X^j \) \((j = 1, 2, \ldots, m)\) be the patterns which need to be stored. If \( B(X^j) = cG(X^j) \) with \( c \in \mathbb{R} \), then a system designing procedure can be given as follows.

**Step 1:** Compute \( \bar{Y} = [Y^1 - Y^m, \ldots, Y^{(m-1)} - Y^m] \) where \( Y^j = G(X^j), j = 1, 2, \ldots, m \).
**Step 2:** Perform a singular value decomposition of \( \bar{Y} \), \( \bar{Y} = \bar{Y}_1 \bar{L} \bar{Y}_2 \), where \( \bar{Y}_1 \in \mathbb{R}^{n \times n} \) and \( \bar{Y}_2 \in \mathbb{R}^{(m-1) \times (m-1)} \) are orthogonal matrices, \( \bar{L} = \begin{bmatrix} \bar{L}_1 & 0 \\ 0 & \bar{L}_2 \end{bmatrix} \in \mathbb{R}^{n \times (m-1)} \) where \( \bar{L}_1 \in \mathbb{R}^{(m-1) \times (m-1)} \) is a diagonal matrix whose elements are the singular values of \( \bar{Y} \).
**Step 3:** Compute \( W = \bar{Y}_1 \Delta \bar{Y}_1^T \), and \( \Theta = -cY^m + WY^m \), where \( \Delta = [\Delta_{ij}] \in \mathbb{R}^{n \times n} \) is a lower triangular matrix with

\[
\Delta_{ij} = \begin{cases} 
  c & \text{if } i = j, \; i \leq m - 1,
  
  \text{rand}(\beta_{ij}, \gamma_{ij}) & \text{if } i > j, \; i \geq m, \; j \geq m,
  
  \delta_i & \text{if } i = j, \; i \geq m,
  
  0 & \text{otherwise},
\end{cases}
\]

where \( \delta_i < 0, \; \beta_{ij}, \gamma_{ij} \in \mathbb{R} \), and \( \text{rand}(\beta_{ij}, \gamma_{ij}) \) denotes random values within interval \((\beta_{ij}, \gamma_{ij})\).