Shadow-Cuts Minimization/Maximization and Complex Hopfield Neural Networks

Zekeriya Uykan, *Senior Member, IEEE*

***Abstract*— In this article, we continue our very recent work by extending it to the complex case. Having been inspired by the real Hopfield neural network (HNN) results, our investigations here yield various novel results, some of which are as follows. First, extending the "biased pseudo-cut" concept to the complex HNN (CHNN) case, we introduce a "shadow-cut" that is defined as the sum of intercluster phased edges. Second, while the discrete-time real HNN strictly minimizes the "biased pseudo-cut" in each neuron state change, the CHNN "tends" to minimize the shadow-cut (as the CHNN energy function is minimized). Third, these definitions pose a novel L-phased graph clustering (partitioning) problem in which the sum of the shadow-cuts is minimized (or maximized) for the Hermitian complex and the directed graphs whose edges are (possibly arbitrary positive/negative) complex numbers. Finally, combining the CHNN and the pioneering algorithm GADIA of Babadi and Tarokh and their modified versions, we propose simple indirect algorithms to solve the defined shadow-cuts minimization/maximization problem. The proposed algorithms naturally include the CHNN as well as the GADIA as its special cases. The computer simulations confirm the findings.**

***Index Terms*— Associative memory, clustering, complex Hopfield neural networks (CHNNs), GADIA, graphs with (positive and negative) complex edges, L-phased partitioning problem.**

I. Introduction and Motivation

COMPLEX-VALUED neural networks are becoming an emergent and rapidly developing field because they recently widened the scope of their applications to machine learning, imaging, remote sensing, optoelectronics, quantum neural systems, physiological neural systems, and artificial neural information processing [33], [34]. Complex-valued neural networks are as follows.

1) They are highly suitable not only for representing graphs with complex edges but also for processing the amplitude and phase. This is one of the core concepts in physical systems dealing with electromagnetic, light, and ultrasonic waves, as well as quantum waves [33].
2) They utilize complex arithmetic, showing specific dynamic characteristics in their learning, self-organizing, and processing [33], [34].

These facts bring critical advantages to representing and modeling complicated relations among the nodes in complex graphs for solving various machine learning problems. In addition, they bring critical advantages in analyzing and processing complex signals in space, time, frequency, and phase domains for practical applications in diverse fields of engineering [33].

The complex-valued Hopfield neural network (CHNN) [6]–[8] is one of the most successful models of complex-valued neural networks [35]. Various variants of CHNN have been proposed, such as a symmetric CHNN [22], a rotor Hopfield neural network (HNN) [36], and an O(2)-valued Hopfield network [37]. For details and applications of the CHNN, see [33]. A special case of the CHNN, the real HNN [9], [10], has been one of the most well-known and widely used neural networks in optimization since the 1980s. Its applications varied from associative memory system design problems (see [6], [11]–[13]) to radio resource optimization in wireless networks (see [15], [18]), from combinatorial optimization (see [16]–[19]) to image restoration, from robotics to system identification to analog-to-digital conversion (see [14]), and so on.

In both a real HNN and a CHNN, the weight matrix naturally represents a graph because of the full connectivity present among their neurons. For example, a real HNN represents an undirected simple graph whose edges are real numbers. A graph is simply a collection of vertices connected by edges. An edge represents the similarity, interaction, or relation between the corresponding two vertices in the graph. For binary graphs, each edge is either 0 or 1. For similarity graphs, the edges can be arbitrary real numbers. The similarity of vertices (nodes and data points) can be defined by different measures, such as the spatial distance and correlation. The edges may be directed or undirected. The graph is directed or undirected depending on whether the edges have directions or no direction associated with the vertices, respectively. The theory of such traditional graphs with nonnegative real edges has been well established since the 19th century. The associated applications from this theory have extended to mathematics, physics, chemistry, linguistics, biology, computer science, and social sciences. Furthermore, the well-established graph partitioning theory has recently broadened its areas of application to social network analysis, big data networks, and biological (protein interaction and gene coexpression) networks, among others. For the graph partitioning theory literature, see [40].

Most of the graph partitioning and graph clustering problems assume that graph edges are nonnegative (see [26]). For example, a wireless system can be represented by a graph with nonnegative edges (see [38], [39]). However, when graphs are allowed to have both negative and positive edges, the problem becomes much more complicated (see [27], [28]). For example, consider the standard minimum path problem

Manuscript received April 25, 2019; revised October 28, 2019 and January 27, 2020; accepted March 4, 2020. Date of publication April 17, 2020; date of current version March 1, 2021.

The author is with the College of Engineering and Technology, American University of the Middle East, Egaila, Kuwait (e-mail: zekeriya.uykan@aum.edu.kw).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2980237

for graphs with negative and positive edges: when this graph is converted to a nonnegative graph by simply adding a certain positive number to each edge, the originality of the optimization problem is generally violated and the optimum solution is eventually changed. For example, let the minimum path from nodes X to Z be $X \rightarrow Y \rightarrow Z$. Adding a positive value to each edge to satisfy the nonnegative condition yields a smaller path $X \rightarrow Z$ if the difference between the paths $X \rightarrow Y \rightarrow Z$ and $X \rightarrow Z$ is smaller than the added positive value. On the other hand, as the real HNN is applied to optimization problems, the corresponding weights (edges) may become (arbitrary) positive and negative numbers because its weights are dictated by the parameters of the problem and its constraints. In short, we need a well-established mathematical theory that deals with graphs with positive and negative edges as well as graphs with arbitrary complex edges.

However, the theory of weighted and/or directed graphs with negative or complex edges is almost unavailable. Many results obtained for traditional graphs do not apply to the directed graphs with complex edges (see [3]). Yet, such complex graphs are seen in various studies. For example, the graphs with complex edges are needed to code arbitrary quantum states in multipartite quantum systems [3]. However, Saif *et al.* reported that a mathematical theory of such graphs is lacking. In [4], the adjacency matrices of the graphs generated from proteomics maps (the actual mass and charge of each protein) are complex matrices, and Balasubramanian *et al.* [4] also noted that the algorithms and techniques for graphs with complex weights need to be generalized. A complex-valued adjacency matrix is used to represent the distance and projection angles in a communication graph of a multiagent control system in [5]. Recently, it has been reported that the complex network and graph theory approach is becoming the emergent field to detect various brain disorders and abnormalities [30], [31].

In a recent article [32], the authors stated that to the best of their knowledge, the construction of similarity graphs with both positive and negative edges from feature vectors for classification has not been studied in the graph-based classifier literature ([32, pp. 716–717]). One unfortunate consequence of negative edge weights is that the graph Laplacian matrix L can be indefinite. Cheung *et al.* [32] presented a perturbation matrix Δ so that $L + \Delta$ is positively semidefinite for their algorithm.

In this article, we study the directed graphs whose edges are possibly negative and/or positive complex numbers. Due to the complexity of the problem, here, we focus on the directed graphs whose edge matrix is a Hermitian complex matrix (as in the CHNN case). To the best of our knowledge, for example, even max-cut and min-cut of such graphs with arbitrary complex edges have not been examined in the literature yet. How can we respond to such Hermitian complex graphs?

Given an $N \times N$ Hermitian weight matrix $\mathbf{W}$, the CHNN maximizes the following Lyapunov (energy) function:

$$L(t) = \overline{\mathbf{x}}^T \mathbf{W} \mathbf{x} = \sum_{j=1}^{N} \sum_{k=1}^{N} w_{jk} \overline{x}_j(t) x_k(t) \qquad (1)$$

where $x_j, x_k \in \left\{e^{-i\phi_1}, e^{-i\phi_2}, \ldots, e^{-i\phi_L}\right\}$, and $\overline{x}_j$ is the complex conjugate of x_j and $w_{jk} = \overline{w}_{kj}$. In our very recent article [1], we shed light on the working principle of a real HNN showing that every neuron in the discrete real HNN with positive/negative weights strictly decreases a "Biased Pseudo-Cut" as the HNN minimizes its energy (Lyapunov) function to solve an optimization problem.

On the other hand, the GADIA proposed by Babadi and Tarokh [2] has been a pioneering algorithm in the interference avoidance literature on wireless communications. Representing the N transmit–receive pairs as a graph, the original basic GADIA addresses the channel allocation problem in the power domain, and thus, all channel power gains (i.e., edges) are modeled as positive real numbers. In this article, our graph represents the same N transmit-receive pairs but in the magnitude domain so that each channel magnitude gain (i.e., edge) is an arbitrary negative/positive complex number. This implies that the proposed simple algorithms extend the original basic GADIA of Babadi and Tarokh [2], which is only in the positive real domain, to the complex case.

A. Contributions of This Article

This article is a continuation of our very recent work in [1], and here, we extend the work in [1] to the complex case. Our contributions can be summarized as follows.

1) Extending the "biased pseudo-cut" concept in [1] to the CHNN case, we introduce a "shadow-cut" concept that is defined as the sum of intercluster phased edges.
2) We show that while the discrete-time real HNN strictly minimizes the "biased pseudo-cut" in each neuron state change, the CHNN "tends" to minimize the shadow-cut (as the CHNN energy function is minimized).
3) These definitions pose a novel L-phased graph clustering (partitioning) problem in which the sum of the shadow-cuts is minimized for the Hermitian complex and directed graphs whose edges are (possibly arbitrary positive/negative) complex numbers.
4) We propose simple indirect algorithms to solve the defined shadow-cuts minimization/maximization problem by combining the CHNN and the GADIA of Babadi and Tarokh [2] and their modified versions. Eventually, the proposed algorithms include the CHNN as well as the GADIA of Babadi and Tarokh [2] as their special cases.
5) The GADIA proposed by Babadi and Tarokh [2] has been a pioneering algorithm in the wireless communications interference avoidance literature. A modification in the presented simple algorithm extends the original GADIA [2] to the complex case.

This article is arranged as follows. In Section II, we pose a pseudo max-cut and pseudo min-cut problem and shadow-cuts minimization problem (i.e., L-phased clustering problem) for the Hermitian complex and directed graphs. We also summarize the related main finding of our very recent article [1]. The simple algorithms solving the posed shadow-cuts minimization problem are presented in Section III, followed by computer

simulation results in Section IV. The conclusions are given in Section V.

II. Shadow-Cuts Minimization/Maximization Problem: L-Phased-Clustering Problem of Hermitian Complex Graphs

First, we state some definitions from graph theory [20] as follows.

Definition (Undirected Graph): If the edges have no direction associated with the vertices, then the graph is called an undirected graph.

Definition (Directed Graph): If the edges have directions associated with the vertices, then the graph is called a directed graph. A directed graph, sometimes called a digraph, allows two edges to join the same vertex, but they must go in opposite directions. However, more than one edge cannot go in the same direction between the vertices. If w_{jk} represents the complex edge from vertex k to vertex j, then w_{kj} represents the complex edge from vertex j to vertex k.

Definition (Inbound/Outbound Edges): Given a directed graph, inbound edges of vertex j are those edges that end at vertex j, i.e., $\{w_{jk}\}_{j=1}^{N}$and outbound edges of vertex j are those edges that originate from vertex j, i.e., $\{w_{kj}\}_{j=1}^{N}$.

Definition (Simple Graph): An undirected graph whose edges are real numbers is called a simple graph.

Definition (Cut of a Simple Graph): A cut means a partition of the vertices of the graph into two sets.

Definition (Size of a Cut): The size of a cut is the sum of the edges cutting the graph into two parts.

Definition (Max-cut of a Graph): A maximum cut is a cut whose size is not smaller than the size of any other cut.

Definition (Complex Graph): A directed graph with arbitrary complex edges is defined as a complex graph in this article.

Definition (Hermitian Complex Graph): A complex graph whose edge matrix is Hermitian is called a Hermitian complex graph. A Hermitian matrix is a complex square matrix that is equal to its own conjugate transpose, i.e., $w_{jk} = \overline{w}_{kj}$, where $\overline{w}_{kj}$ is the conjugate of w_{kj}, for every j and k.

Definition [State or Phase of Each Vertex (x_j)]: Let the number of vertices (nodes, data points) of the graph be N and the number of classes or phases be L (where $N \gg L$). Let us evenly divide the complex unit circle into L regions that are represented by the complex set $P = \{e^{-i\phi_1}, e^{-i\phi_2}, \ldots, e^{-i\phi_L}\}$. Each and every vertex (nodes, neurons, and data points) is assigned to only one of these states (phases) at any given time. The state or phase of vertex j at time t is represented by $x_j(t) = e^{-i\phi_{s(j)}}$, where $j \in \{1, 2, \ldots, N\}$ and $s(j) \in \{1, 2, \ldots, L\}$. The time dependence of $s(j)$ is omitted in this article for simplifying the notation.

Definition: The sum of intracluster edges, denoted as $\Phi_{\text{sum}}^{\text{intra}}(t)$, is defined as

$$\Phi_{\text{sum}}^{\text{intra}}(t) = \sum_{s=1}^{L}\left(\sum_{j\in\ C_S(t)}\sum_{k\in\ C_S(t)} w_{jk}\overline{x}_j x_k\right) = \sum_{s=1}^{L}\left(\sum_{j\in\ C_S(t)}\sum_{k\in\ C_S(t)} w_{jk}\right). \quad (2)$$

In (2), $j, k \in C_s(t)$, i.e., both vertices (nodes) are in the same cluster C_s at time t, therefore, $s(j) = s(k)$, and thus, $\overline{x}_j x_k = e^{-i\phi_{s(j)}} e^{i\phi_{s(k)}} = 1$.

Definition: The sum of shadow-cuts is defined as the sum of intercluster *phased* edges. Given a Hermitian complex graph whose edges are $w_{jk} = r_{jk}e^{i\alpha_{jk}}$, let the cluster set (i.e., phase set) be $P = \{e^{-i\phi_1}, e^{-i\phi_2}, \ldots, e^{-i\phi_L}\}$. Then, let us assign N vertices to L clusters (phases), and let the sets $\{C_s\}_{s=1}^{L} = \{C_1, C_2, \ldots, C_L\}$ represent the cluster sets, i.e., the indices of vertices in each cluster. Therefore, the sum of shadow-cuts is given by

$$J_{\text{SC}}\left(\{C_s(t)\}_{s=1}^{L}\right) = \sum_{s=1}^{L}\left(\sum_{j\in\ C_S}\sum_{k\in\ \overline{C}_S} w_{jk}\overline{x}_j x_k\right) = \sum_{s=1}^{L}\left(\sum_{j\in\ C_S}\sum_{k\in\ \overline{C}_S} r_{jk}e^{i\left(\alpha_{jk}+\phi_{s(k)}-\phi_{s(j)}\right)}\right) \quad (3)$$

where set $\overline{C}_S$ represents the indices of all other vertices except cluster s and $s(j)$ and $s(k)$ represent the cluster (phase) index of vertex j and vertex k, respectively, and $s(j), s(k) \in \{1, 2, \ldots, L\}$. Each of the L clusters is represented by its unique phase. The phase set is defined as $P = \{e^{-i\phi_1}, e^{-i\phi_2}, \ldots, e^{-i\phi_L}\}$.

Definition: Biased pseudo-cut for graphs/matrices with negative and positive edges/weights is defined in [1]

$$J_{\text{PC}}(k) = \sum_{i\in\ C_1(k)}\sum_{j\in\ C_2(k)} w_{ij} + \sum_{j\in\ C_1(k)} b_j. \quad (4)$$

In this article, graph clustering means finding sets of "related" vertices in graphs [40]. It is the task of grouping the vertices of the graph into clusters in such a way that the vertices in the same cluster are more "similar" to each other than to those in other clusters. Different metrics for graph clustering are proposed for graph clustering. For details, see [40].

A. Working Principle of the Real Discrete HNN

In this section, we summarize the working principle of the discrete-time real HNN. Bruck [41] shed light on the behavior of any neuron in the discrete HNN when proving the convergence of the discrete HNN in his paper. A very recent paper [1] has further investigated the behavior of any neuron in not only the discrete-time HNN but also the continuous-time HNN when the corresponding energy function is minimized during an optimization process. For the sake of consistency and brevity, here, we use the notation in [1] when explaining the discrete HNN's working principle in [41]. Let us assume that a given constrained or unconstrained optimization function has already been written in the form of the Lyapunov function of the discrete real HNN. From [1, Propositions 1–3], any neuron in the real discrete-HNN with weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$ changes its state if and only if it negatively contributes to the defined biased pseudo-cut $J_{\text{PC}}(k)$ in (4); in other words, if and only if the neuron decreases biased pseudo-cut $J_{\text{PC}}(k) = \sum_{i\in\ C_1(k)}\sum_{j\in\ C_2(k)} w_{ij} + \sum_{j\in\ C_1(k)} b_j$ in (4), in each step when the HNN energy function is decreased

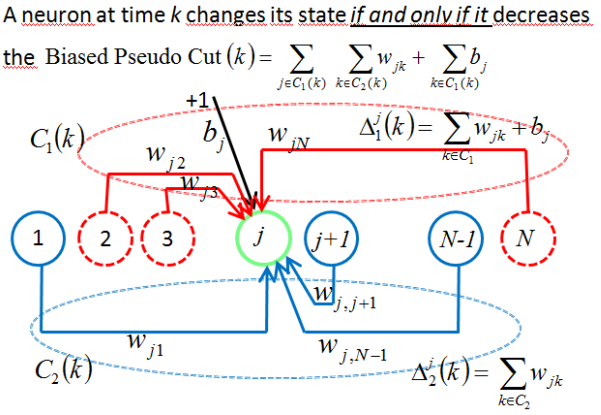

Fig. 1. Illustration of the discrete HNN working principle in [1], [41] for an arbitrary neuron j at an arbitrary time k.

during an optimization process. To be precise, let $\Delta_1^j(k) = \sum_{k\in C_1} w_{jk} + b_j$ and $\Delta_2^j(k) = \sum_{k\in C_2} w_{jk}$ represent the contribution of neuron j to its current class and its potential contribution to the other class (depending on the neuron state) at time k. Neuron j changes its state (phase) if and only if it decreases the biased pseudo-cut $J_{\rm PC}(k)$ in (4) in step k. On the other hand, for an HNN with $(-\mathbf{W})$ and $(-\mathbf{b})$, any neuron changes its state if and only if it further increases the biased pseudo-cut $J_{\rm PC}(k)$ in (4). The working principle of the discrete HNN is shown in Fig. 1 for an arbitrary neuron j at time k (Algorithm L_N, step 3 [42, p. 1581]).

The article [1] also shows that the discrete HNN with $(-\mathbf{W})$ and $(-\mathbf{b})$ is equivalent to the GADIA [2] with $(\mathbf{W})$ and $(\mathbf{b})$ ([1, Preposition 5]), which implies that the discrete HNN is actually a special case of the GADIA in optimization. These results imply that the "information" of a real HNN with $\mathbf{W}$ and an HNN with $-\mathbf{W}$ is always stored in the local minima and maxima of the biased pseudo-cut, respectively, $J_{\rm PC}(k) = \sum_{i\in C_1(k)} \sum_{j\in C_2(k)} w_{ij} + \sum_{j\in C_1(k)} b_j$ in (4) regardless of the type of optimization problem.

B. Pseudo Max-Cut and Pseudo Min-Cut Problem of a Directed Complex Graph (Case $L = 2$)

Definition: The pseudo min-cut problem of a Hermitian complex graph. Let $\mathbf{W} \in C^{N\times N}$ be a Hermitian matrix consisting of positive and negative complex edges of a directed complex graph. If $L = 2$ and $\{\phi_1 = 0, \phi_1 = \pi\}$, i.e., $x_j \in \{+1, -1\}$, then the pseudo min-cut problem of a Hermitian complex graph is defined as minimizing the shadow-cut $J_{\rm SC}(\cdot)$ (i.e., minimizing the sum of intercluster *phased* edges) in (3) for $L = 2$. Therefore, we are to determine the two cluster sets C_1 and C_2 that minimize $J_{\rm SC}(\cdot)$ in (3)

$$\text{pseudo min cut}(\mathbf{W}) = \min_{\{C_1,C_2\}} J_{\rm SC}(C_1, C_2) = \max_{\{C_1,C_2\}} \left(\sum_{j\in C_1} \sum_{k\in C_2} w_{jk} + \sum_{j\in C_2} \sum_{k\in C_1} w_{jk} \right). \tag{5}$$

If $L = 2$, then $\overline{x}_j x_k = -1$ when x_j and x_k are from different clusters. This provides the second row in (5).

Definition: The pseudo max-cut problem of a Hermitian complex graph is defined as follows. From (3) and (5)

$$\text{pseudo max cut}(\mathbf{W}) = \max_{\{C_1,C_2\}} J_{\rm SC}(C_1, C_2) = \min_{\{C_1,C_2\}} \left(\sum_{j\in C_1} \sum_{k\in C_2} w_{jk} + \sum_{j\in C_2} \sum_{k\in C_1} w_{jk} \right). \tag{6}$$

From (3) and (4), $J_{\rm SC}(C_1, C_2) = -J_{\rm PC}(C_1, C_2)$.

Proposition 1: Let $\mathbf{W} \in C$ be a Hermitian matrix that consists of positive and negative complex edges of a directed graph. If $L = 2$ and $\{\phi_1 = 0, \phi_1 = \pi\}$, i.e., $x_j \in \{+1, -1\}$, then 1) the CHNN with $\mathbf{W}$ maximizes the shadow cut $J_{\rm SC}(C_1, C_2)$ in (3) and 2) the CHNN with $(-\mathbf{W})$ minimizes the shadow cut $J_{\rm SC}(C_1, C_2)$ in (3) for the Hermitian directed complex graphs.

Proof:

1) *CHNN With $\mathbf{W}$:* To simplify the notation, we write $I_{\rm sum}^{\rm intra}(t) = I_{\rm sum}^{\rm intra}(C_1(t), C_2(t))$ and $J_{\rm SC}(t) = J_{\rm SC}(C_1(t), C_2(t))$. If $x_j, x_k \in \{-1, +1\}$, then $(x_j - x_k)^2 = 2(1 - x_j x_k)$ and

$$\sum_{j=1}^{N} \sum_{k=1}^{N} w_{jk} \big(x_j(t) - x_k(t)\big)^2 = 2 \sum_{j=1}^{N} \sum_{k=1}^{N} w_{jk} - 2L(t) \tag{7}$$

where $L(t)$ is the Lyapunov (energy) function of the CHNN. On the other hand, from (3)

$$\sum_{j=1}^{N} \sum_{k=1}^{N} w_{jk} \big(x_j(t) - x_k(t)\big)^2 = -4J_{\rm SC}(t). \tag{8}$$

From (7) and (8)

$$L(t) = V + 2J_{\rm SC}(t) \tag{9}$$

where $V = \sum_{j=1}^{N} \sum_{k=1}^{N} w_{jk}$ is a constant. From (9), we conclude that maximizing the energy function $L(t)$ in (1) by the CHNN is equivalent to maximizing $J_{\rm SC}(t)$ in (3). In other words, because the energy function $L(t)$ in (1) is further increased in each cluster (phase) update of the CHNN, the shadow-cut $J_{\rm SC}(t)$ in (3) is further increased (i.e., the pseudo-cut $J_{\rm PC}(t)$ in (4) is further decreased) in the same way. Therefore, the CHNN with $\mathbf{W}$ maximizes the shadow-cut $J_{\rm SC}(t)$ in (3) for the two-cluster case.

2) *CHNN With $-\mathbf{W}$:* Let $(\hat{\mathbf{W}} = -\mathbf{W})$, $\hat{J}_{\rm SC}(t) = -J_{\rm SC}(t)$, and $\hat{L}(t) = -L(t)$. Following (7) to (9), we have $\hat{L}(t) = -V + 2\hat{J}_{\rm SC}(t)$. So

$$-V = \hat{L}(t) + 2J_{\rm SC}(t). \tag{10}$$

From (6) and (10), because the energy function $L(t)$ in (1) is further decreased in each cluster (phase) update of the CHNN, the shadow-cut $J_{\rm SC}(t)$ in (3) is further decreased (i.e., the pseudo-cut $J_{\rm PC}(t)$ in (4) is further increased) in the same way. Therefore, the CHNN with $-\mathbf{W}$ minimizes the shadow-cut $J_{\rm SC}(t)$ in (3), which completes the proof. ■

C. Minimization of the Sum of Shadow-Cuts: L-Phased Clustering (Partitioning) Problem for Hermitian Complex Graphs (Case $L \geq 3$)

In Section II-B, the number of clusters was 2, i.e., $L = 2$. In what follows, we first pose a complex graph clustering (partitioning) problem for $L \geq 3$, and then, in Section III, we propose simple indirect algorithms to solve it.

Definition [Sum of Shadow-Cuts Maximization Problems (or L-Phased Clustering (Partitioning) Problem for Hermitian Complex Graphs)($L \geq 3$)]: Given a directed complex graph whose edge matrix is an $N \times N$ Hermitian complex matrix, the L-phased clustering (partitioning) problem is defined as determining the optimum cluster sets $\{C_s\}_{s=1}^{L} = \{C_1, C_2, \ldots, C_L\}$, which maximizes shadow-cut $J_{\mathrm{SC}}(t)$ in (3), i.e., maximizes the sum of intercluster phased edges

$$\max_{\{C_1,C_2,\ldots,C_L\}} J_{\mathrm{SC}}\left(\{C_s\}_{s=1}^{L}\right) = \max_{\{C_1,C_2,\ldots,C_L\}} \sum_{s=1}^{L} \left(\sum_{j \in C_s} \sum_{k \in \overline{C}_s} w_{jk} e^{i\left(\overbrace{\phi_{s(k)}}^{t-\text{phase}} - \overbrace{\phi_{s(j)}}^{r-\text{phase}}\right)} \right) \quad (11)$$

where $\phi_{s(k)}$ and $\phi_{s(j)}$ are the transmit phase (t-phase), which is of vertex k, and receive phase (r-phase), which is of vertex j, respectively, with respect to the directed edge w_{jk}. The last equation in (11) is due to the fact that the edge matrix is Hermitian, i.e., $w_{jk} = \overline{w}_{kj}$ for every j and k.

Our motivation for maximizing the sum of intercluster phased edges [i.e., the shadow-cut $J_{\mathrm{SC}}(t)$ in (3)] comes from the results in [1], from Proposition 1, which shows that the shadow-cut $J_{\mathrm{SC}}(t)$ is maximized when energy is maximized for the $L = 2$ case, and from the notion of the clustering. The common understanding of a clustering process is that the vertices (neurons and data points) from different clusters are expected to be as "far" or "different" as possible.

Referring to the shadow-cut maximization problem that we pose by (11), we note the following.

1) It is well known that the graph clustering problem for even simple undirected graphs whose edges are nonnegative real numbers is an NP-complete problem. The phased graph clustering problem in (11) is a much more challenging problem because a new optimization dimension, vertex-specific phase dimension, is added on top of the original undirected graph clustering problem and the edges are (positive and negative) complex numbers in (11), while the edges are only nonnegative real numbers in the original undirected simple graph clustering problem.
2) Any vertex j is allocated to only one cluster denoted as $s(j)$ at a given time. Each cluster is represented and distinguished by its unique phase ϕ_s in the phase domain, and therefore, all the vertices that are in the same cluster have the same unique phase.

TABLE I

COMPARISON OF THE SUM OF SHADOW CUTS $J_{\mathrm{SC}}(t)$ WITH THE STANDARD CUT, THE PSEUDO-CUT IN [1], AND THE GADIA

L	**Edges (Weights)** w_{jk}	**Sum of Shadow Cuts $J_{SC}(t)$ in (3)**
2	Real and nonnegative $\left(w_{jk} \in \Re^{+}\right)$	$J_{SC}(C_1, C_2) = -2\sum_{j \in C_1} \sum_{k \in C_2} w_{jk} = -cut(C_1, C_2)$ ($-$ standard definition of graph *cut*)
>2	Real and nonnegative $\left(w_{jk} \in \Re^{+}\right)$	If ***phase*** is neglected, then $J_{SC}(t)$ is equal to the GADIA case. $J_{SC}\left(\{C_s\}_{s=1}^{L}\right) = -J_{GADIA}\left(\{C_s\}_{s=1}^{L}\right)$ in [1] $= -\sum_{s=1}^{L}\left(\sum_{j \in C_s} \sum_{k \in \overline{C}_s} w_{jk}\right)$
2	Real, negative and positive $\left(w_{jk} \in \Re\right)$	$J_{SC}(C_1, C_2) = -J_{PC}(C_1, C_2)$ (shadow cut = -pseudo cut in [1])
≥ 2	Complex, negative and positive $\left(w_{jk} \in \mathbf{C}\right)$	sum of *inter*-cluster *phased* edges $J_{SC} = \sum_{s=1}^{L}\left(\sum_{j \in C_s} \sum_{k \in \overline{C}_s} w_{jk} e^{i\left(\overbrace{\phi_{s(k)}}^{t-phase} - \overbrace{\phi_{s(j)}}^{r-phase}\right)}\right)$

3) For a given real graph/HNN with nonnegative real edges/weights, if $L = 2$, then we have only two clusters (phases), say, $P = \{e^{i0}, e^{i\pi}\}$, equivalently $x_j \in \{-1, +1\}$, and then, the following holds.
   a) The shadow maximization/minimization problem in (11) is reduced to the standard min-cut/max-cut problem in the literature (see Table I).
   b) These observations, the results in [41], and Proposition 1 imply that what the discrete HNN in this case really does (as its energy function is minimized) is actually to realize the well-known "min-cut max-flow" within the HNN.
4) As mentioned in Section I, the theory of weighted and directed graphs with possibly negative and positive complex edges is almost unavailable. To the best of our knowledge, the shadow-cut concept as well as the shadow-cuts maximization/minimization problem for such Hermitian complex graphs is defined for the first time in this article. The pseudo-cut concept for (undirected simple) graphs with negative and positive real edges has very recently been proposed in [1] for the first time, to the best of our knowledge.
5) In correlation clustering [24], [25], [29], we also have positive and negative real edge weights. The positive edge and negative edge indicate that the corresponding vertices are similar or dissimilar, respectively, and the

goal is to either maximize agreements (the sum of positive edge weights within a cluster plus the absolute value of the sum of negative edge weights between clusters) or minimize disagreements (the absolute value of the sum of negative edge weights within a cluster plus the sum of positive edge weights across clusters). Denoting positive and negative edges as w_{jk}^{+} and w_{jk}^{-}, respectively, the correlation clustering problem is defined as

$$\max\left(\underbrace{\sum_{s=1}^{L}\left(\sum_{j\in\ C_S}\sum_{k\in\ C_S} w_{jk}^{+}\right)}_{\substack{\text{sum of positive edges}\\ \text{within the same cluster}}}+\underbrace{\sum_{s=1}^{L}\left(\sum_{j\in\ C_S}\sum_{k\in\ \overline{C}_S}\left|w_{jk}^{-}\right|\right)}_{\substack{\text{absolute sum of negative edges}\\ \text{between different clusters}}}\right) \tag{12}$$

or

$$\min\left(\underbrace{\sum_{s=1}^{L}\left(\sum_{j\in\ C_S}\sum_{k\in\ C_S}\left|w_{jk}^{-}\right|\right)}_{\substack{\text{absolute sum of negative edges}\\ \text{within the same cluster}}}+\underbrace{\sum_{s=1}^{L}\left(\sum_{j\in\ C_S}\sum_{k\in\ \overline{C}_S} w_{jk}^{+}\right)}_{\substack{\text{sum of positive edges}\\ \text{between different clusters}}}\right). \tag{13}$$

Comparing (3) with (12) and (13), we see that the main differences between the proposed shadow-cuts minimization and the standard correlation clustering are as follows: 1) the absolute values of the negative weights are taken into the maximization of the correlation clustering, while the phased weights (without any absolute value operation) are considered in the shadow-cuts minimization and 2) the weights are only real numbers in the correlation clustering, but the weights are complex in the proposed shadow-cuts minimization problem. As mentioned earlier, to the best of our knowledge, the shadow-cut concept as well as the shadow-cuts maximization/minimization problem for complex graphs is defined for the first time in this article. Therefore, we cannot take any correlation clustering algorithm as a reference algorithm in this article. The algorithms from the well-established graph partitioning literature cannot be used as reference algorithms either simply because they deal with graphs with real edges only and there is no "phase" concept in standard graph partitioning problems, i.e., the edges are not "phased." Having been inspired by the recent results in [1] and Proposition 1, the CHNN seems to be the only reference algorithm for our proposed shadow-cuts minimization/maximization algorithms so far.

III. Simple Indirect Algorithms for Solving the L-phased Clustering Problem for Hermitian Complex Graphs

For a given Hermitian directed complex graph, first let us define the sum of cluster-specific edges for vertex (node) n, denoted as Φ_n^s at time t

$$\Phi_n^s(t)=\underbrace{\sum_{k\in\ C_S} w_{nk}}_{\substack{\text{sum of "inbound" edges (from the}\\ \text{vertices in cluster } s \text{ to vertex } n)}}+\underbrace{\sum_{k\in\ C_S} w_{kn}}_{\substack{\text{sum of "outbound" edges (from}\\ \text{vertex } n \text{ to the vertices in cluster } s)}} \tag{14}$$

where $s=1,2,\ldots,L$ and $w_{nk}=r_{nk}e^{i\alpha_{nk}}$. Therefore, for vertex (node, neuron) n, we have $\{\Phi_n^1,\Phi_n^2,\ldots,\Phi_n^L\}$.

Definition: The sum of phased-inbound-edges for vertex n, denoted as I_n^{sum}, is defined as

$$I_n^{\text{sum}}(t)=\sum_{k=1}^{N} w_{nk}x_k(t)=\left|I_n^{\text{sum}}\right|e^{i\theta_{\text{sum}}}. \tag{15}$$

A. Indirect Greedy and Nongreedy Algorithms for $L\geq 3$

In Section II-A, the number of clusters was 2, i.e., $L=2$. In this section, we analyze the same problem for an arbitrary number of clusters, i.e., for $L\geq 3$. In what follows, we present a simple indirect greedy asynchronous algorithm (SIGA) and its nongreedy version, called the simple indirect nongreedy asynchronous algorithm (SINA), in order to solve the shadow-cuts maximization problem, i.e., L-phased complex graph partitioning problem that we pose by (11).

1) Simple Indirect Greedy Algorithm (SIGA): Let the state (i.e., cluster, phase, and color) of vertex n at time t be $s(n)^{prev}$. The SIGA is presented in Table II.

Proposition 2: Let $\mathbf{W}\in\ C^{N\times N}$ be an arbitrary Hermitian matrix consisting of the edges of a complex directed graph. The proposed algorithm SIGA in (16)–(18) maximizes the sum of shadow-cuts $J_{\text{SC}}(t)$, i.e., sum of intercluster phased edges, in (3).

Proof: The first part is called "Complex GADIA" because it is a straightforward extension of the basic GADIA in [2] to the complex case. In order to differentiate between the real matrix case and the complex matrix case, let us refer to the latter case as "Complex GADIA." Otherwise, the algorithm is the same as the basic GADIA in [2]: In other words, the following "Complex GADIA" extends the pioneering algorithm GADIA proposed by Babadi and Tarokh [2], which was only for positive edge matrices, to a complex case where the edges are positive and negative complex numbers:

To simplify the notation, we write $I_{\text{sum}}^{\text{intra}}(t)=I_{\text{sum}}^{\text{intra}}(\{C_s(t)\}_{s=1}^{L})$ and $J_{\text{SC}}(t)=J_{\text{SC}}(\{C_s(t)\}_{s=1}^{L})$. Taking the sum of cluster-specific edges for vertex (node) n, denoted as Φ_n^s in (14), into account, we put the sum of intracluster edges $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2) into two parts from the perspective of vertex n as follows: From (2) and (14)

$$\Phi_{\text{sum}}^{\text{intra}}(t)=\Phi_n^{s(n)}(t)+\Phi_{\text{others}}^{\text{intra}}(t) \tag{19}$$

where $\Phi_n^{s(n)}(t)=\sum_{k\in\ C_{s(n)}}(w_{nk}+w_{kn})$ from (14) and $\Phi_{\text{others}}^{\text{intra}}(t)$ show the sum of all other terms in $\Phi_{\text{sum}}^{\text{intra}}(t)$.

TABLE II
SIGA

Step (1) : Determine the following two winning clusters for vertex n at time t:

(i) winner 1 ("complex GADIA"): The sum of cluster-specific edges $\Phi_n^s(t)$ for vertex n is calculated by (14): $\{\Phi_n^1, \Phi_n^2, \cdots, \Phi_n^L\}$. Then

$$w_1 = index_s \min\{\Phi_n^s(t)\} \tag{16}$$

(ii) winner 2 (CHNN): The sum of *phased*-inbound-edges I_n^{sum} and its phase θ_{sum} is calculated from (15). Then

$$w_2 = index_s \min|\theta_{sum} - \phi_s| \tag{17}$$

Step (2): The cluster (i.e. state, phase, color) index of vertex n is updated at time *(t+1)* according to the following simple rule:

$$s(n) = \begin{cases} w_1, & \text{if } w_1 = w_2 \\ s(n)^{prev}, & \text{otherwise} \end{cases} \tag{18}$$

where $s(n)$ is the cluster index at time *(t+1)*. So, the state of vertex n is $x_n(t+1) = e^{i\phi_{s(n)}}$. Whenever the cluster (phase) of vertex n is changed from $s(n)^{prev}$ to a new cluster (phase) w_1, then the corresponding two cluster sets $C_{s(n)^{prev}}(t+1)$ and $C_{s(n)}(t+1)$ are updated accordingly (and all other sets remain the same).

Step (3): Continue step (1) and (2) *asynchronously* for each vertex one by one until there is no state (cluster) change any more, reaching a local maxima.

Therefore, $\Phi_n^{s(n)}(t)$ represents the "contribution" of vertex/node/neuron n to its cluster $s(n)$ and $\Phi_{\text{others}}^{\text{intra}}$ represents all the terms that do not include x_n according to the cluster sets $\{C_s(t)\}_{s=1}^L$. From (16), for the first winning cluster w_1 at time $(t+1)$

$$\Phi_n^{w_1}(t+1) = \min\ \{\Phi_n^1(t), \Phi_n^2(t), \ldots, \Phi_n^{\text{L}}(t)\}. \tag{20}$$

Thus, at time *(t+1)*

$$\Phi_{\text{sum}}^{\text{intra}}(t+1) = \Phi_n^{w_1}(t+1) + \Phi_{\text{others}}^{\text{intra}}(t+1). \tag{21}$$

Because $I_{\text{others}}^{\text{intra}}(t)$ does not include x_n, $\Phi_{\text{others}}^{\text{intra}}(t+1) = \Phi_{\text{others}}^{\text{intra}}(t)$. Therefore, from (19)–(21)

$$\Phi_{\text{sum}}^{\text{intra}}(t+1) \le \Phi_{\text{sum}}^{\text{intra}}(t). \tag{22}$$

On the other hand, it is well known that the CHNN maximizes the Lyapunov (energy) function $L(t)$ in (1). Using (1)–(3) and (15), we obtain

$$L(t) = \underbrace{2\left|I_n^{\text{sum}}\right| \cos(\theta_{\text{sum}} - \phi_{n(s)})}_{\text{related to } x_n} + \underbrace{L_n^{\text{others}}(t)}_{\text{without } x_n} \tag{23}$$

where $L_n^{\text{others}} = \sum_{\substack{j=1 \\ (j \neq n)}}^N \sum_{\substack{k=1 \\ (k \neq n)}}^N w_{jk}\overline{x}_j(t)x_k(t)$ includes all terms that do not contain x_n. Furthermore, note that I_n^{sum} does not include x_n either because edge matrix $\mathbf{W}$ is zero-diagonal. From (17)

$$\left|\theta_{\text{sum}} - \phi_{w_2}\right| \le \{|\theta_{\text{sum}} - \phi_1|, |\theta_{\text{sum}} - \phi_2|, \ldots, |\theta_{\text{sum}} - \phi_L|\}. \tag{24}$$

For the second winning cluster w_2 at time *(t+1)*

$$L(t+1) = 2\left|I_n^{\text{sum}}\right| \cos(\theta_{\text{sum}} - \phi_{w_2}) + L_n^{\text{others}}(t+1). \tag{25}$$

In (23) and (25), $L_n^{\text{others}}(t+1) = L_n^{\text{others}}(t)$ because they do not include any terms with x_n. From (23)–(25)

$$L(t+1) \ge L(t). \tag{26}$$

Using (1)–(3), at any time t

$$J_{\text{SC}}(t) = L(t) - \Phi_{\text{sum}}^{\text{intra}}(t). \tag{27}$$

From (22), (26), and (27), we conclude that

$$J_{\text{SC}}(t+1) \ge J_{\text{SC}}(t) \tag{28}$$

and whenever a vertex is assigned from its current cluster to a different cluster by (18), $J_{\text{SC}}(t)$ in (3) is strictly increased. Because the maximum of $J_{\text{SC}}(t)$ in (3) is finite, the SIGA in (16)–(18) converges to a local maximum after a finite number of steps, which completes the proof. ■

Corollary 1: Let $\mathbf{W}$ be an arbitrary Hermitian matrix consisting of the edges of a complex directed graph. The proposed algorithm in (16)–(18) for $(-\mathbf{W})$ minimizes the sum of shadow cuts $J_{\text{SC}}(t)$, i.e., the sum of intercluster phased edges, in (3). ■

Corollary 2: If all edges are positive real and symmetrical, i.e., matrix $\mathbf{W} \in \Re_+^{N\times N}$, and if the second winner w_2 is omitted above, i.e., if

$$s(n) = w_1 \tag{29}$$

in (18), then the proposed SIGA is reduced to the basic GADIA algorithm of Babadi and Tarokh [2]. The basic GADIA is for a simple undirected graph whose edge matrix $\mathbf{W}$ is a positive real symmetric only and thus minimizes the sum of intracluster edges $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2). ■

Corollary 3: Let $\mathbf{W} \in C^{N\times N}$ be an arbitrary Hermitian matrix consisting of the edges of a complex directed graph. If the first winner w_1 is omitted in step 2 above, i.e., if

$$s(n) = w_2 \tag{30}$$

in (18), then the proposed SIGA is reduced to the well-known CHNN, which maximizes the Lyapunov (energy) function $L(t)$ in (1). ■

Comparison of SIGA With the Basic GADIA in [2]: The GADIA proposed by Babadi and Tarokh [2] has been a pioneering algorithm in the interference avoidance literature on wireless communications. The original basic GADIA is designed in the power domain, and thus, all channel power gains (edges) are modeled as positive numbers. Channel power gains correspond to edges. Therefore, the basic GADIA solves the simple undirected graph clustering problem with positive edges only. However, the abovementioned SIGA addresses the complex and phased graph clustering problem. The edges of SIGA may represent the amplitude channel gains that are arbitrary complex numbers in the context of wireless systems.

2) Simple Indirect Nongreedy Algorithm (SINA): In this section, we shed light on the working principle of the SIGA in (16)–(18) and present its "nongreedy" version to solve the same L-phased complex graph partitioning problem in (11): The proposed simple algorithm in (16)–(18) is a greedy one in the sense that for vertex n, it determines the cluster index w_1 that decreases the sum of intracluster edges $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2) the most and the cluster index w_2 that increases the Lyapunov function $L(t)$ in (1) the most at time t. However, most of the time, there are multiple other clusters that also decrease the same $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2) (although they reduce it less than the cluster w_1 does). Similarly, most of the time, there are multiple other clusters that also increase the same Lyapunov function $L(t)$ in (1) (although these clusters increase it less than the cluster w_1 does). How can we find all the cluster indices decreasing $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2) and all the cluster indices increasing the Lyapunov function $L(t)$ in (1)? In what follows, we first examine this question and then present our nongreedy algorithm:

First, we suggest that we put the sum of phased-inbound-edges I_n^{sum} in (15) into two parts as follows:

$$I_n^{\text{sum}} = \left|I_n^{\text{sum}}\right| e^{i\theta_{\text{sum}}} = \underbrace{I_n^{w}}_{\text{intra-cluster}} + \underbrace{I_n^{o}}_{\text{inter-cluster}} \tag{31}$$

where

$$\underbrace{I_n^{w}}_{\text{intra-cluster}} = \underbrace{\sum_{k\in\ C_w} w_{nk} e^{i\phi_w}}_{\substack{\text{phased "inbound" weights only from} \\ \text{the vertices in cluster } w \text{ to vertex } n}} = \left|I_n^{w}\right| e^{i\theta_{nw}} \tag{32}$$

and

$$\underbrace{I_n^{o}}_{\text{inter-cluster}} = \underbrace{\sum_{k\in\ \overline{C}_w} w_{nk} e^{i\phi_{s(k)}}}_{\substack{\text{phased "inbound" weights from all the} \\ \text{vertices in clusters other than } w \text{ to vertex } n}} = \left|I_n^{o}\right| e^{i\theta_{no}}. \tag{33}$$

From (14) and (32) and the fact that $w_{jk} = \overline{w}_{kj}$, we obtain $\Phi_n^{w}(t) = 2|I_n^{w}|\cos(\theta_{nw} - \phi_w)$. In (31)–(33), the superscript "w" stands for the "winner or winning cluster" and the superscript "o" stands for "other clusters." Therefore, I_n^{w} belongs to the intracluster term $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2) and I_n^{o} belongs to the sum of the shadow-cut $J_{\text{SC}}(\{C_s\}_{s=1}^{L})$, i.e., sum of intercluster *phased*-edges, in (3). Note that in (32) and (33), I_n^{w} and I_n^{o} depend on set C_w, and therefore, both magnitudes $|I_n^{w}|$ and $|I_n^{o}|$ as well as both angles θ_{nw} and θ_{no} also depend on set C_S. Whenever set C_S is updated, all magnitudes and angles change accordingly. To shed light on the working principle of the proposed simple algorithm, we examine Fig. 2 illustrating I_n^{sum}, I_n^{w}, and I_n^{o} for vertex n in two random snapshots. Fig. 2 shows the vector projections of I_n^{sum}, I_n^{O}, and I_n^{w} onto the cluster vector x_w for vertex n. Let vertex n be assigned to cluster w. We conclude from Fig. 2 that the following holds.

1) The projection of I_n^{sum}, I_n^{w}, and I_n^{o} onto its cluster vector $x_{s(n)}$ belongs to the Lyapunov function $L(t)$ in (1), the sum of intracluster edges $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2), and the shadow cuts $J_{\text{SC}}(\{C_s\}_{s=1}^{L})$, i.e., sum of intercluster phased edges, in (3), respectively.
2) All the clusters that strictly increase/decrease $L(t)$ in (1), $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2), and $J(\{C_s\}_{s=1}^{L})$ in (3) can independently and computationally easily be determined from these three projections. The greater the projection of I_n^{sum} onto cluster vector x_{sum}, the greater $L(t)$ is. Similarly, the smaller the projection of I_n^{w} onto cluster vector x_w, the smaller $\Phi_{\text{sum}}^{\text{intra}}(t)$ is. For the details, see proof of Proposition 3.

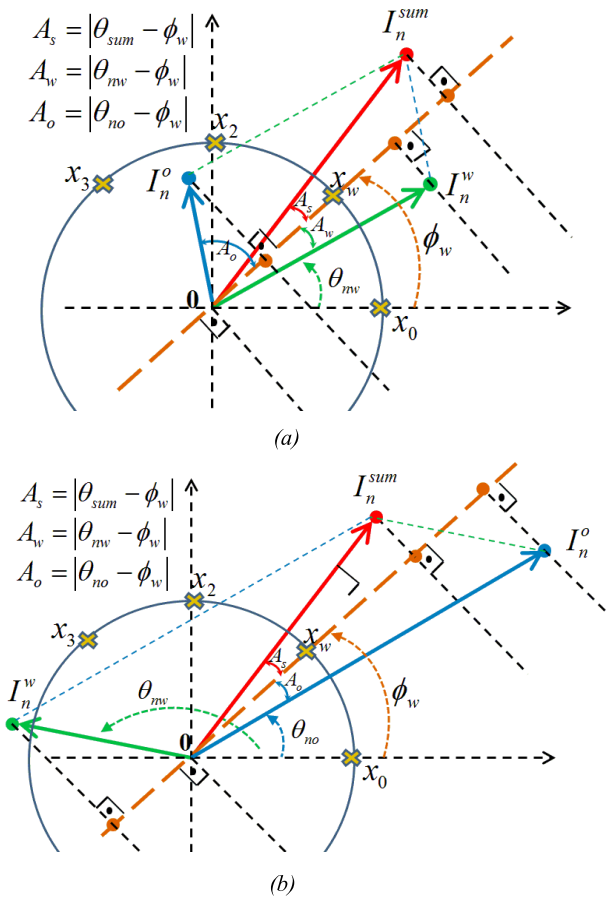

Fig. 2. Illustration of vector projections of I_n^{sum}, I_n^{0}, and I_n^{w} onto the cluster vector x_w for node n at two random snapshots in (a) and (b). (a) Random snapshot. (b) Another random snapshot.

The proposed SINA when $L \geq 3$ is summarized in Table III, where the cluster (i.e., state, phase) index of vertex n at time t is represented by $s(n)$.

Whenever the cluster of vertex n is changed from $s(n)^{\text{prev}}$ to a new cluster w, then the corresponding two cluster sets $C_{s(n)^{\text{prev}}}(t+1)$ and $C_{s(n)}(t+1)$ are updated accordingly.

In step (1), the set $S_{\mathbf{w}_1}$ in (35) consists of all the cluster indices decreasing $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2). The set $S_{\mathbf{w}_2}$ in (36) consists of all the cluster indices increasing the Lyapunov function $L(t)$ in (1).

Proposition 3: Let $\mathbf{W} \in C^{N\times N}$ be an arbitrary Hermitian matrix consisting of the edges of a complex directed graph. The proposed asynchronous and nongreedy algorithm, SINA, in (35)–(38) maximizes the sum of shadow-cuts $J_{\text{SC}}(t)$, i.e., sum of intercluster phased edges, in (3).

TABLE III
SINA

Step (1): Determine the following two winning sets:

(i) winner1-set ("complex N-GAIR [21]"): From (14), for vertex n, we have $\{\Phi_n^1, \Phi_n^2, \cdots, \Phi_n^L\}$ where

$$\Phi_n^s(t) = \sum_{k \in C_s(n)} \left(w_{jk} + w_{kj}\right) \tag{34}$$

The set of winner1 candidate clusters $S_{\mathbf{w}_1}$ is determined by

$$S_{\mathbf{w}_1} = indices_s \left\{\Phi_n^s(t) < \Phi_n^{s(n)}(t)\right\} \tag{35}$$

(ii) winner2-set ("non-greedy CHNN"): The sum of *phased*-inbound-edges I_n^{sum} and its phase θ_{sum} is calculated by (15). Then

$$S_{\mathbf{w}_2} = indices_s \left\{\cos\left(\theta_{sum} - \phi_s\right) > \cos\left(\theta_{sum} - \phi_{s(n)}\right)\right\} \tag{36}$$

or alternatively

$$S_{\mathbf{w}_2} = indices_s \left\{\mathrm{Re}\left(I_n^{sum} e^{-i\phi_s}\right) > \mathrm{Re}\left(I_n^{sum} e^{-i\phi_{s(n)}}\right)\right\} \tag{37}$$

Step (2): Find the set of candidate clusters as intersection of $S_{\mathbf{w}_1}$ and $S_{\mathbf{w}_2}$, i.e., $S_{\mathbf{w}} = S_{\mathbf{w}_1} \cap S_{\mathbf{w}_2}$. If $S_{\mathbf{w}}$ is an empty set, then vertex j remains in its cluster. Otherwise, let w represents the cluster index chosen randomly from set $S_{\mathbf{w}}$. The cluster (i.e. phase, color) index of vertex n is updated at time *(t+1)* according to the following simple rule

$$s(n) = \begin{cases} w, & \text{if } S_{\mathbf{w}} \text{ is NOT empty set} \\ s(n)^{prev}, & \text{otherwise} \end{cases} \tag{38}$$

where $s(n)^{prev}$ is the cluster index at time *t*. So, the state of vertex n is $x_n(t+1) = e^{i\phi_{s(n)}}$.

Step (3): Continue step (1) and (2) asynchronously for each vertex until there is no cluster (phase) change any more, reaching a local maxima.

Proof: Let us assume that the vertex n is moved from cluster $s(n)^{prev}$ to the cluster w at time $(t+1)$ by the proposed SINA in (35)–(38). Using (23) and (31)–(33) and $w_{jk} = \overline{w}_{kj}$ for cluster w at time $(t+1)$ gives

$$\left|I_n^{\mathrm{sum}}\right| \cos(\theta_{\mathrm{sum}} - \phi_w) = \left|I_n^w\right| \cos(\theta_{nw} - \phi_w) + \left|I_n^o\right| \cos(\theta_{no} - \phi_w). \tag{39}$$

Equation (39) can also be verified easily by the projection vectors in Fig. 2. Using (23), (27), and (39), we obtain the sum of shadow-cuts $J_{\mathrm{SC}}(t+1)$, i.e., the sum of intercluster phased edges, as follows:

$$J_{\mathrm{SC}}(t+1) = 2\left|I_n^{\mathrm{sum}}\right| \cos(\theta_{\mathrm{sum}} - \phi_w) + L_n^{\mathrm{others}}(t+1) - 2\left|I_n^w\right| \cos(\theta_{nw} - \phi_w) - \Phi_{\mathrm{others}}^{\mathrm{intra}}(t+1) \tag{40}$$

where $L_n^{\mathrm{others}}(t+1)$ and $\Phi_{\mathrm{others}}^{\mathrm{intra}}(t+1)$ do not include x_n. On the other hand, from (14) and (32), we obtain

$$\begin{aligned} \Phi_n^w(t+1) &= \sum_{k \in C_w(t+1)} \left(w_{jk} + w_{kj}\right) \\ &= e^{-i\phi_w} \sum_{k \in C_w(t+1)} \left(w_{jk} + w_{kj}\right) e^{i\phi_w} \\ &= 2\left|I_n^w\right| \cos\left(\theta_{n,w} - \phi_w\right). \end{aligned} \tag{41}$$

Similarly, for the same vertex n in cluster $s(n)$ at time t, we obtain the sum of shadow-cuts $J_{\mathrm{SC}}(t)$ in (3) using (27) and (39) as follows:

$$J_{\mathrm{SC}}(t) = 2\left|I_n^{\mathrm{sum}}\right| \cos\left(\theta_{\mathrm{sum}} - \phi_{s(n)}\right) + L_n^{\mathrm{others}}(t) - 2\left|I_n^{s(n)}\right| \cos\left(\theta_{n,s(n)} - \phi_w\right) - \Phi_{\mathrm{others}}^{\mathrm{intra}}(t) \tag{42}$$

where $L_n^{\mathrm{others}}(t)$ and $\Phi_{\mathrm{others}}^{\mathrm{intra}}(t)$ do not include x_n. Furthermore, from (14), (32), and (35)–(38), we obtain

$$2\left|I_n^w\right| \cos(\theta_{nw} - \phi_w) = \Phi_n^w(t+1). \tag{43}$$

From (40)–(43) and using the fact that $L_n^{\mathrm{others}}(t)$, $L_n^{\mathrm{others}}(t+1)$, $\Phi_{\mathrm{others}}^{\mathrm{intra}}(t)$, and $\Phi_{\mathrm{others}}^{\mathrm{intra}}(t+1)$ do not include x_n, we obtain

$$\begin{aligned} &J_{\mathrm{SC}}(t+1) - J_{\mathrm{SC}}(t) \\ &\quad = 2\left|I_n^{\mathrm{sum}}\right| \left\{\cos(\theta_{\mathrm{sum}} - \phi_w) - \cos\left(\theta_{\mathrm{sum}} - \phi_{s(n)}\right)\right\} \\ &\qquad - \Phi_n^w(t) + \Phi_n^{s(n)}(t). \end{aligned} \tag{44}$$

Because $\Phi_n^w(t) < \Phi_n^{s(n)}(t)$ from (35) and $\cos(\theta_{\mathrm{sum}} - \phi_w) > \cos(\theta_{\mathrm{sum}} - \phi_{s(n)})$ from (36), we obtain from (44) that

$$J_{\mathrm{SC}}(t+1) \geq J_{\mathrm{SC}}(t) \tag{45}$$

and therefore, whenever a vertex is assigned from its current cluster to a different cluster by the SINA in (35)–(38), the sum of shadow cuts $J_{\mathrm{SC}}(t)$ in (3) is strictly increased. Because the maximum of the shadow cuts $J_{\mathrm{SC}}(t)$ is finite, the SINA converges to a local maximum after a finite number of steps, which completes the proof. ■

Comparison of the SINA with the N-GAIR in [21]: The first step of the SINA includes the N-GAIR in [21]. However, the N-GAIR in [21], a variant of GADIA [2], solves the simple undirected graph clustering problem for positive edges only. However, the abovementioned SINA addresses the complex directed graph clustering problem.

3) SIGA2 and SINA2: In the abovementioned SIGA and SINA, $L(t)$ is increased and $\Phi_{\mathrm{sum}}^{\mathrm{intra}}(t)$ is decreased simultaneously, and that is why $J(t)$ is indirectly increased and both SIGA and SINA are called indirect algorithms. Note that $L(t) = \Phi_{\mathrm{sum}}^{\mathrm{intra}}(t) + J_{\mathrm{SC}}(t)$ from (27). However, both SIGA and SINA have strict restrictions: A cluster update takes place in SIGA if and only if the cluster that increases the $L(t)$ the most coincides with the cluster that decreases the $\Phi_{\mathrm{sum}}^{\mathrm{intra}}(t)$ the most. In addition, there is no correlation between $L(t)$ and $\Phi_{\mathrm{sum}}^{\mathrm{intra}}(t)$ due to the randomness of the edges. Similar restriction is available for the SINA as well. Thus, in what follows, in light of the mathematical analysis of the SINA mentioned earlier, we conclude that it is computationally easy to determine the change in $L(t)$ as well as in $\Phi_{\mathrm{sum}}^{\mathrm{intra}}(t)$ for each candidate cluster using only the local information that is readily available at each vertex (neuron, node) and relax the strict conditions and allow the cluster update to occur whenever the change in $L(t)$ is greater than the change in $\Phi_{\mathrm{sum}}^{\mathrm{intra}}(t)$.

These two conclusions are the main points of the following versions, called SIGA2 and SINA2.

SIGA2: The steps of SIGA2 are given in Table IV.

Corollary 4: Let $\mathbf{W} \in C^{N \times N}$ be an arbitrary Hermitian matrix consisting of the edges of a complex directed graph. The proposed simple indirect algorithm SIGA2 in (46)–(48)

TABLE IV
SIGA2

Step (1) : Determine index w_2 from (17).

Step (2) : Determine $\Delta^L_{w_2}$ and Δ^Φ_s as follows:

$$\Delta^L_{w_2} = 2\,\mathrm{Re}\left\{I_n^{sum}\left(x_{w_2} - x_n\right)\right\} \tag{46}$$

where $x_{w_2} = e^{i\phi_{w_2}}$, and

$$\Delta^\Phi_{w_2} = \Phi_n^{w_2}(t) - \Phi_n^{s(n)^{prev}}(t) \tag{47}$$

where $s(n)^{prev}$ is the cluster index of vertex *n* at time *(t)*.

Step (3) : Apply the following simple rule:

$$s(n) = \begin{cases} w_2, & \text{if } \Delta^L_{w_2} > \Delta^\Phi_{w_2} \\ s(n)^{prev}, & \text{otherwise} \end{cases} \tag{48}$$

Step (4): Continue steps (1) to (3) asynchronously for each vertex until there is no phase (state, cluster) update any more, reaching a local maxima.

TABLE V
SINA2

Step (1) : Determine the set $S_{\mathbf{w}_2}$ from (37).

Step (2) :

for $q \in S_{\mathbf{w}_2}$

Determine Δ^L_q and Δ^Φ_q, where $q \in S_{\mathbf{w}_2}$ as follows:

$$\Delta^L_q = 2\,\mathrm{Re}\left\{I_n^{sum}\left(x_q - x_n\right)\right\} \tag{49}$$

where $x_q = e^{i\phi_q}$, and

$$\Delta^\Phi_q = \Phi_n^q(t) - \Phi_n^{s(n)^{prev}}(t) \tag{50}$$

where $s(n)^{prev}$ is the cluster index of vertex *n* at time *(t)*.

$$s(n) = q, \quad \text{if } \Delta^L_q > \Delta^\Phi_q \tag{51}$$

break the for-loop if $q \neq s(n)^{prev}$

end

locally maximizes the sum of shadow-cuts $J_{\mathrm{SC}}(t)$, i.e., sum of intercluster phased edges, in (3).

Proof: Writing (46)–(48) in the steps in (19)–(25) results in $J_{\mathrm{SC}}(t+1) > J_{\mathrm{SC}}(t)$ whenever there is a cluster update by the SIGA2. Because $J_{\mathrm{SC}}(t)$ is upper bound, $J_{\mathrm{SC}}(t)$ converges to a local maximum after a finite number of steps, which completes the proof. ■

SINA2: The steps of SINA2 are given in Table V.

Corollary 5: Let $\mathbf{W} \in C^{N\times N}$ be an arbitrary Hermitian matrix consisting of the edges of a complex directed graph. The proposed simple indirect algorithm SINA2 in (49)–(51) locally maximizes the sum of shadow-cuts $J_{\mathrm{SC}}(t)$, i.e., sum of intercluster phased edges, in (3).

Proof: Writing (49)–(51) in the steps in (39)–(45) results in $J_{\mathrm{SC}}(t+1) > J_{\mathrm{SC}}(t)$ whenever there is a cluster update by the SINA2. Because $J_{\mathrm{SC}}(t)$ is upper bounded, $J_{\mathrm{SC}}(t)$ converges to a local maximum after a finite number of steps, which completes the proof. ■

IV. Simulation Results

In this section, we design three different examples in order to examine mainly the evolution and characteristics of the defined sum of shadow cuts [i.e., sum of *inter cluster phased edges* $J_{\mathrm{SC}}(t)$] (3) for the CHNN and the proposed algorithms.

A. Example 1: Random Complex Matrix Case

In order to shed light on the working principles of the proposed algorithms, first, we examine their behaviors in the cases where the edges are random complex numbers normally distributed. Therefore, the real and/or imaginary parts of about half of the edges are negative. The number of vertices is 25, and thus, the dimension of the Hermitian matrix is 5×5. The number of clusters is eight, i.e., $L = 8$. Therefore, the complex unit circle is evenly divided into eight parts, and

$$x_i \in P = \left\{e^{-i0}, e^{-i\frac{\pi}{4}}, e^{i\frac{\pi}{4}}, e^{-i\frac{\pi}{2}}, e^{i\frac{\pi}{2}}, e^{-i\frac{3\pi}{4}}, e^{i\frac{3\pi}{4}}, e^{i\pi}\right\} \tag{52}$$

where $i = 1, 2, \ldots, N$. First, let us investigate the traditional CHNN. As the CHNN evolves by time from its initial condition until it converges, we calculate the percentages of changes and the normalized sum of changes in the Lyapunov (energy) function $L(t)$ in (1), the sum of intracluster edges $\Phi^{\mathrm{intra}}_{\mathrm{sum}}(t)$ in (2), and the shadow-cuts $J_{\mathrm{SC}}(t)$, (i.e., sum of intercluster phased edges) in (3) and plot all of them in Fig. 3. We observe the following from Fig. 3(a) and (b): for most of the cluster updates, roughly, around 80% of the cluster updates, $J_{\mathrm{SC}}(t)$ in (3) also increases, and; $\Phi^{\mathrm{intra}}_{\mathrm{sum}}(t)$ in (2), on the other hand, at roughly half of the cluster updates, increases/decreases.

In other words, Fig. 3(a) and (b) shows that while $L(t)$ in (1) is maximized, the traditional CHNN has a tendency that most of the cluster updates (roughly 80% for this specific example) also increase $J_{\mathrm{SC}}(t)$ in (3) for a normally distributed random Hermitian matrix realization.

We run all the proposed algorithms and the traditional CHNN for the same Hermitian matrix and the same initial condition and plot the evolution of the sum of intercluster *phased* edges (i.e., the sum of shadow-cuts) $J_{\mathrm{SC}}(t)$ in (3) for each algorithm in Fig. 4(a). A zoom into the sum of the shadow cuts $J_{\mathrm{SC}}(t)$ after step 25 is given in Fig. 4(b). As seen from Fig. 4(b), the traditional CHNN does not mathematically guarantee that $J_{\mathrm{SC}}(t)$ increases or decreases for a cluster update though the complete plot Fig. 4(a) suggests that the traditional CHNN seems to have "a tendency" to increase $J_{\mathrm{SC}}(t)$ as well. Fig. 4 also confirms that all the proposed algorithms strictly increase $J_{\mathrm{SC}}(t)$ for each cluster update.

The number of cluster updates for all the proposed algorithms is given in Fig. 5. We see from Fig. 5 that the following holds.

1) The SIGA has only a few cluster updates as expected due to the strict cluster restrictions on both $L(t)$ and $\Phi^{\mathrm{intra}}_{\mathrm{sum}}(t)$.
2) *About the Proposed Algorithms:* The SINA has more cluster updates compared to the SIGA as expected; similarly, SINA2 has more cluster updates compared to the SIGA2 due to the SINA's and SINA2's non-greedy features. On the other hand, SIGA2 has (always)

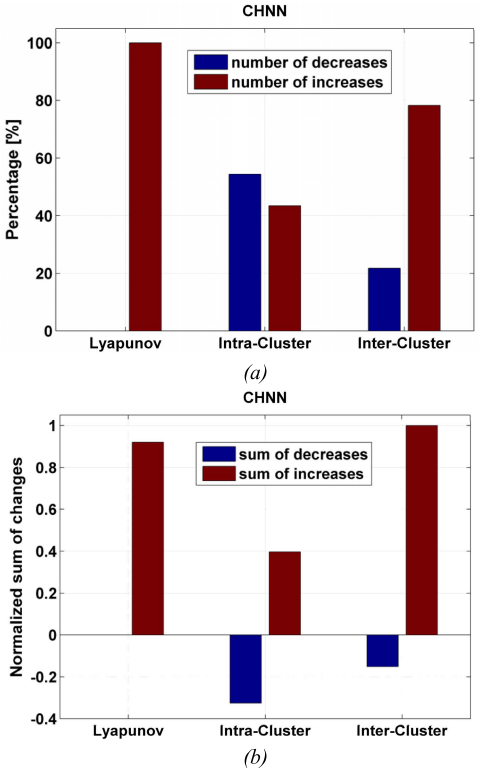

Fig. 3. (a) Percentages of changes and (b) normalized sum of changes in Lyapunov function $L(t)$ in (1), the sum of intracluster edges $\Phi_{\text{sum}}^{\text{intra}}$ in (2), and the sum of intercluster phased edges $J_{\text{SC}}(t)$ in (3) in Example 1.

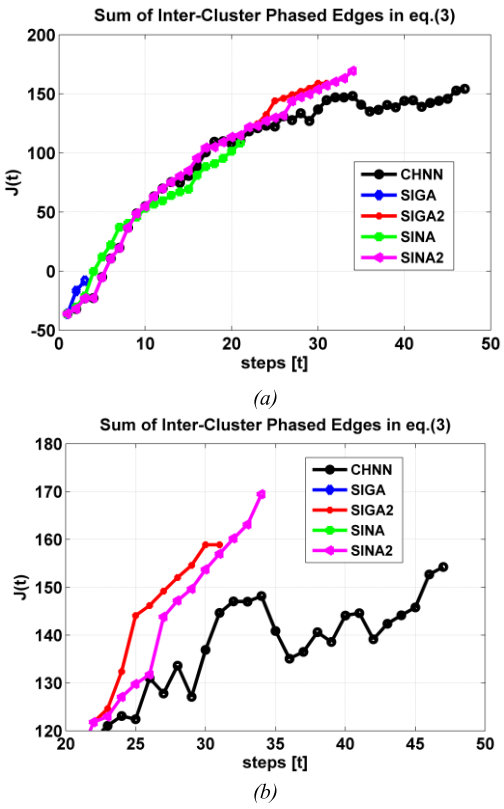

Fig. 4. (a) Evolution of the sum of intercluster phased edges $J_{\text{SC}}(t)$ in (3) for the algorithms in Example 1. (b) A closer look after step 25.

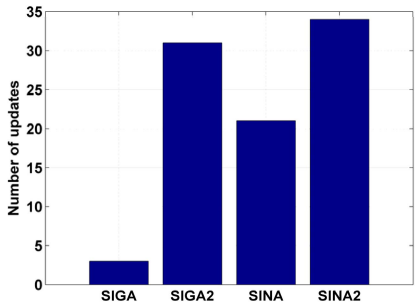

Fig. 5. Number of cluster updates in Example 1.

more cluster updates than SIGA; similarly, SINA2 has (always) more cluster updates than SINA because the cluster candidate sets of SIGA and SINA are (always) subsets of the candidate sets of SIGA2 and SINA2, respectively. This yields additional cluster (phase) candidates in the cases of SIGA2 and SINA2 as compared to the SIGA and SINA, respectively.

In order to gain insight into the evolution of $L(t)$, $\Phi_{\text{sum}}^{\text{intra}}(t)$ and $J_{\text{SC}}(t)$ for the same algorithm, we plot the results of SIGA2 in Fig. 6, where there are 30 cluster updates. Fig. 6 confirms that as the sum of shadow cuts $J_{\text{SC}}(t)$ is maximized, $L(t)$ also strictly increases in each cluster update by the SIGA2, as explained in Section III.

On the other hand, in order to gain insight into the evolution of the sum of the intracluster edges $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2) for all algorithms, we plot them in Fig. 7. While $\Phi_{\text{sum}}^{\text{intra}}(t)$ for the SIGA and the SINA strictly decreases (as expected), it fluctuates for the SIGA2, the SINA2, and the CHNN.

B. Example 2: Pattern Restoration (Deterministic Case)

This example is related to the complex-valued associative memory systems for pattern restoration. The desired patterns are shown in Fig. 8, where there are 25 pixels. The patterns can be represented by a graph where each pixel corresponds to a vertex. Each pixel (vertex) has a color (phase or cluster). The number of gray colors (clusters or phases) is 8, i.e., $L = 8$ as in Example 1, $x_i \in P$ in (52), where $i = 1, 2, \ldots, 25$. As an example, the pattern 2 is defined by the matrix in (53). Concatenating the rows of the matrix in (53) into one column vector gives the desired pattern vector 2, called $\mathbf{d}_2$

$$\begin{bmatrix} e^{i\pi} & e^{i\pi} & e^{i\frac{\pi}{2}} & e^{i\frac{\pi}{2}} & e^{i0} \\ e^{i0} & e^{i0} & e^{i0} & e^{i\frac{3\pi}{4}} & e^{i0} \\ e^{-i\frac{3\pi}{4}} & e^{-i\frac{3\pi}{4}} & e^{i\frac{\pi}{2}} & e^{i\frac{\pi}{2}} & e^{i0} \\ e^{i\frac{\pi}{4}} & e^{i0} & e^{i0} & e^{i0} & e^{i0} \\ e^{i\pi} & e^{i\pi} & e^{-i\frac{\pi}{4}} & e^{-i\frac{\pi}{4}} & e^{i0} \end{bmatrix}. \tag{53}$$

Various learning algorithms can be used to determine the complex edge matrix $\mathbf{W}$ like the Hebbian rule, the energy based methods in [12] or in [13], symmetric complex matrix in [22], or the projection rule in [23]. For the sake of brevity, here, we use the Hebbian rule [19]

$$\mathbf{W} = \sum_{k=1}^{3} \mathbf{d}_k \mathbf{d}_k^* \tag{54}$$

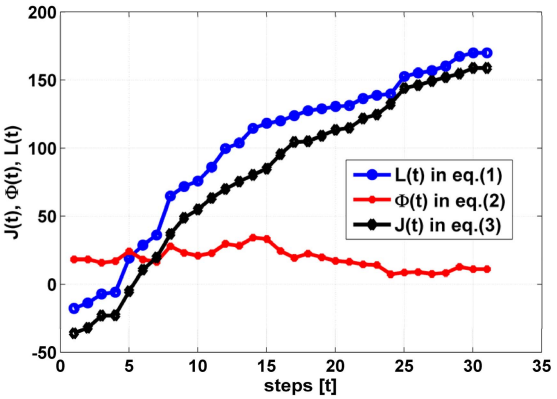

Fig. 6. Evolution of $L(t)$, $\Phi_{\text{sum}}^{\text{intra}}(t)$, and $J_{\text{SC}}(t)$ by the SIGA2 in Example 1.

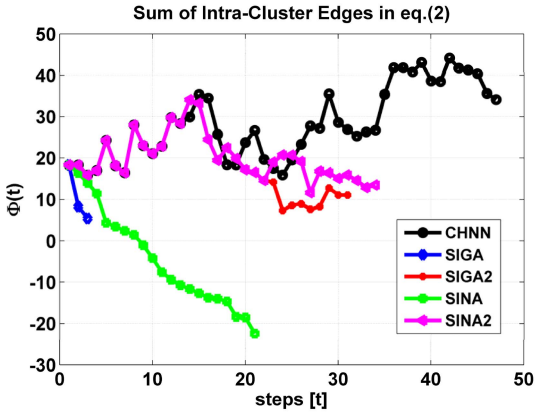

Fig. 7. Evolution of the sum of intracluster edges $\Phi_{\text{sum}}^{\text{intra}}(t)$ in (2) for all algorithms in Example 1.

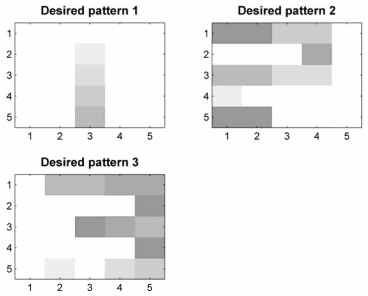

Fig. 8. Desired patterns for Example 2.

where $\mathbf{d}_k$ is the desired complex vectors representing the patterns in Fig. 8 and $\mathbf{d}_k^*$ is the transpose conjugate of $\mathbf{d}_k$. The complex edge matrix $\mathbf{W}$by the Hebbian learning in (54) is already a Hermitian.

We have carried out various simulation campaigns for a Hamming distance (HD) of 3. As a typical example, a distorted pattern with HD of 3 is shown in Fig. 9, for pattern 3. The evolution of $L(t)$, $\Phi_{\text{sum}}^{\text{intra}}(t)$, and $J_{\text{SC}}(t)$ for the SIGA2 is shown in Fig. 10, where there are three cluster updates at steps 1–3 (step 0 is the initial condition). The figure shows while $J_{\text{SC}}(t)$ is maximized, the energy function $L(t)$ also increases but the sum of intracluster weights $\Phi_{\text{sum}}^{\text{intra}}(t)$ fluctuates (i.e., increases and decreases) for this particular example

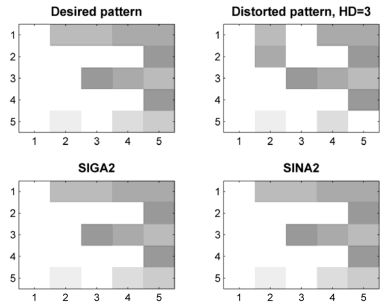

Fig. 9. Desired and a distorted pattern (with HD = 3) for pattern 3 and the results of SIGA2 and SINA2.

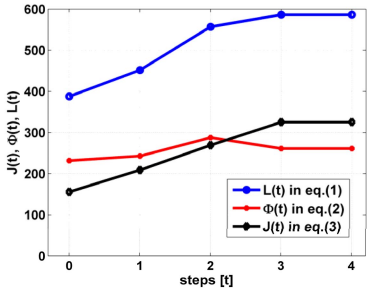

Fig. 10. Evolution of $L(t)$, $\Phi_{\text{sum}}^{\text{intra}}(t)$, and $J_{\text{SC}}(t)$ by the SIGA2 for pattern 3 in Example 2.

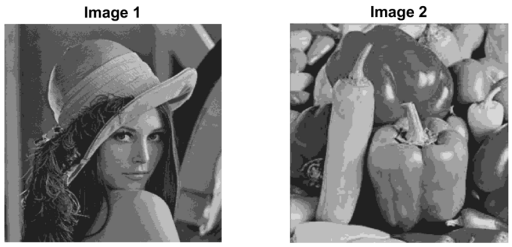

Fig. 11. 3-bit gray-scale versions of two well-known test images, namely, Lenna and peppers.

during the optimization process. The results of the proposed SIGA2 and SINA2 are shown in the second row of Fig. 9, which shows that both the SIGA2 and the SINA2 manage to restore pattern 3.

C. Example 3: Image Restoration

We examine a 3-bit image restoration problem where each pixel is represented by 3 bits, i.e., by eight different phases. Without the loss of generality, we use 3-bit gray-scale versions of two well-known test images, namely, Lenna and peppers test images as shown in Fig. 11.

Similar to what the authors did in, e.g., [12], after segmenting each image and distorting them by salt-and-pepper noise (where the noise intensity is 0.35), we test to restore the images by the CHNN, the SIGA2, and the SINA2. The same weight matrix obtained by the Hebbian learning rule is used for all CHNN, SIGA2, and SINA2. As an example, a distorted image (with salt-and-pepper noise) and the restored images by the CHNN, the SIGA2, and the SINA2 are shown

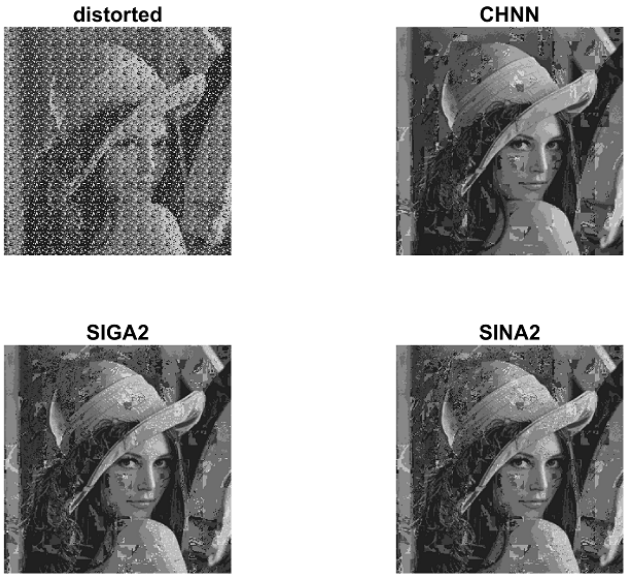

Fig. 12. Distorted 3-bit Lenna image with pepper noise (noise density 0.35) and their reconstructions by the CHNN, SIGA2, and SINA2.

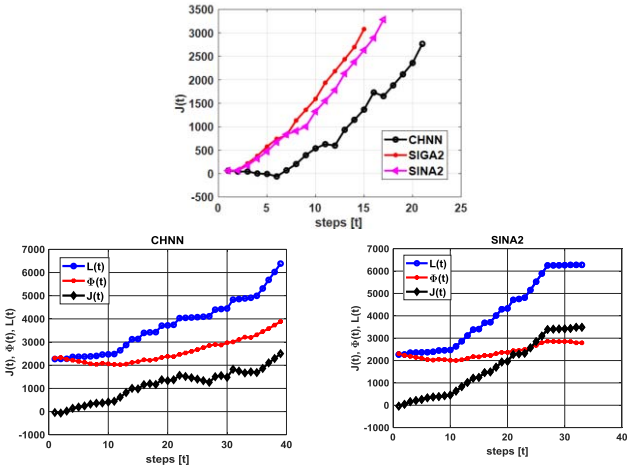

Fig. 13. Typical evolution of $L(t)$, $\Phi_{\text{sum}}^{\text{intra}}(t)$, and $J_{\text{SC}}(t)$ of the CHNN, SIGA2, and SINA2 during a segment processing of the test image.

in Fig. 12. As seen in Fig. 12, all three methods manage to restore the original image to an acceptable extent. Here, we note that the results presented in Fig. 12 are obtained straightforwardly by the Hebbian rule without any further tuning or optimization, whatsoever. For example, even 3-bit gray-scale level optimization is not considered here, etc.

To give an idea about the evolution of $L(t)$, $\Phi_{\text{sum}}^{\text{intra}}(t)$, and $J_{\text{SC}}(t)$ of the CHNN, the SIGA2, and the SINA2 during segment processing, we plot them in Fig. 13. Referring to our simulation results and conclusions in Examples 1 and 2, Fig. 13 confirms the findings presented in Sections II and III.

V. Conclusion and Future Work

In this article, we extend the work in [1] to the complex case. Being inspired by the real HNN results in [1], our investigations here yield various novel results.

1) Extending the "biased pseudo-cut" concept in [1] to the Complex HNN (CHNN) case, we introduce a "shadow-cut" concept that is defined as the sum of inter*cluster phased edges*.
2) While the discrete-time real HNN strictly minimizes the "biased pseudo-cut" in each neuron state change, the CHNN "tends" to minimize the "shadow-cut" (as the HNN energy function is minimized).
3) These definitions pose a novel L-phased graph clustering (partitioning) problem in which the sum of the shadow-cuts is minimized for the Hermitian complex and directed graphs whose edges are (possibly arbitrary positive/negative) complex numbers.
4) We propose simple indirect algorithms to solve the defined shadow-cut minimization problem via combining the CHNN and the GADIA of Babadi and Tarokh [2] and their modified versions. Therefore, the proposed algorithms include the CHNN as well as the GADIA of Babadi and Tarokh [2] as its special cases.
5) The theory of weighted and directed graphs with possibly negative and positive complex edges is almost unavailable, and to the best of our knowledge, the shadow-cut concept as well as the shadow-cut maximization/ minimization problem for Hermitian complex graphs is defined for the first time in this article.
6) The computer simulations confirm the findings.
7) The results in this article imply that the "information" of the proposed algorithms (the SINA, the SIGA, the SINA2, and the SIGA2) is always stored in the local minima (or local maxima) of the defined sum of the shadow-cuts in (3) and (11) regardless of the type of optimization problem.

We think that due to the multidisciplinary nature of the research topic (touching complex-valued neural networks, real HNN, Ising model, Complex HNN, graph theory, graph clustering with negative/positive edges, and machine learning and optimization), the presented results could be extended to quite a few other interesting dimensions. Our current and near future research subjects include developing novel effective methods for designing CHNN weight matrices as applied to various associative memory or optimization problems using the "deeper understanding" developed in this article. The limitation of this study is that the optimization problem should be formulated as the (standard CHNN) energy function in (1) where the complex matrix $\mathbf{W}$ is Hermitian.

References

[1] Z. Uykan, "On the working principle of the hopfield neural networks and its equivalence to the GADIA in optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 4, 2019, doi: 10.1109/TNNLS.2019.2940920.
[2] B. Babadi and V. Tarokh, "GADIA: A greedy asynchronous distributed interference avoidance algorithm," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6228–6252, Dec. 2010.
[3] A. S. M. Hassan and P. S. Joag, "A combinatorial approach to multipartite quantum systems: Basic formulation," *J. Phys. A, Math. Theor.*, vol. 40, no. 33, pp. 10251–10290, Aug. 2007.
[4] K. Balasubramanian, K. Khokhani, and S. C. Basak, "Complex graph matrix representations and characterizations of proteomic maps and chemically induced changes to proteomes," *J. Proteome Res.*, vol. 5, no. 5, pp. 1133–1142, May 2006.

[5] Y. Lou and Y. Hong, "Distributed surrounding design of target region with complex adjacency matrices," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 283–288, Jan. 2015.
[6] A. J. Noest, "Discrete-state phasor neural networks," *Phys. Rev. A, Gen. Phys.*, vol. 38, no. 4, pp. 2196–2199, Aug. 1988.
[7] A. Hirose, "Dynamics of fully complex-valued neural networks," *Electron. Lett.*, vol. 13, no. 16, pp. 1492–1494, 1992.
[8] S. Jankowski, A. Lozowski, and J. M. Zurada, "Complex-valued multistate neural associative memory," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1491–1496, 1996.
[9] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci., USA*, vol. 79, no, 8, pp. 2554–2558, 1982.
[10] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 55, pp. 141–146, Jul. 1985.
[11] V. Gimenez-Martinez, "A modified hopfield auto-associative memory with improved capacity," *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 867–878, Jul. 2000.
[12] M. K. Muezzinoglu, C. Guzelis, and J. M. Zurada, "A new design method for the complex-valued multistate hopfield associative memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 891–899, Jul. 2003.
[13] M. K. Muezzinoglu, C. Guzelis, and J. M. Zurada, "An energy function-based design method for discrete hopfield associative memory with attractive fixed points," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 370–378, Mar. 2005.
[14] V. Chande and P. G. Poonacha, "On neural networks for analog to digital conversion," *IEEE Trans. Neural Netw.*, vol. 6, no. 5, pp. 1269–1274, Sep. 1995.
[15] M. Sheikhan and E. Hemmati, "High reliable disjoint path set selection in mobile ad-hoc network using hopfield neural network," *IET Commun.*, vol. 5, no. 11, pp. 1566–1576, Jul. 2011.
[16] M. Atencia, G. Joya, and F. Sandoval, "Dynamical analysis of continuous higher-order hopfield networks for combinatorial optimization," *Neural Comput.*, vol. 17, no. 8, pp. 1802–1819, Aug. 2005.
[17] G. Joya, M. Atencia, and F. Sandoval, "Hopfield neural networks for optimization: Study of the different dynamics," *Neurocomputing*, vol. 43, nos. 1–4, pp. 219–237, 2002.
[18] Z. Uykan, "Fast-convergent double-sigmoid hopfield neural network as applied to optimization problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 990–996, Jun. 2013.
[19] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. New York, NY, USA: PWS, 1996.
[20] N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory*. London, U.K.: Oxford Univ. Press, 1986.
[21] Z. Uykan, "N-GAIR: Non-greedy asynchronous interference reduction algorithm in wireless networks," *Ad Hoc Sensor Wireless Netw.*, vol. 23, nos. 1–2, pp. 93–116, 2014.
[22] M. Kobayashi, "Symmetric complex-valued hopfield neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 1011–1015, Apr. 2017.
[23] T. Isokawa *et al.*, "Complex-valued associative memories with projection and iterative learning rules," *J. Artif. Intell. Soft Comput. Res.*, vol. 8, no. 3, pp. 237–249, Jul. 2018.
[24] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Mach. Learn.*, vol. 56, nos. 1–3, pp. 89–113, 2004.
[25] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica, "Correlation clustering in general weighted graphs," *Theor. Comput. Sci.*, vol. 361, nos. 2–3, pp. 172–187, Sep. 2006.
[26] P. Mitra and M. Samal, "Approximation algorithm for correlation clustering," in *Proc. 1st Int. Conf. Netw. Digital Technol.*, Jul. 2009, pp. 140–145.
[27] H. Y. Wei and Z. X. Su, "The maximum flow problem with negative capacities or flows," in *Proc. Joint Int. Conf. Service Science, Manage. Eng. (SSME)*, 2016, pp. 1–6.
[28] A. Clark, Q. Hou, L. Bushnell, and R. Poovendran, "A submodular optimization approach to leader-follower consensus in networks with negative edges," in *Proc. American Control Conference (ACC)*, May 2017, pp. 1346–1352.
[29] P. Li, G. J. Puleo, and O. Milenkovic, "Motif and hypergraph correlation clustering," *IEEE Trans. Inf. Theory*, early access, Sep. 10, 2019, doi: 10.1109/TIT.2019.2940246.
[30] S. Supriya, S. Siuly, H. Wang, J. Cao, and Y. Zhang, "Weighted visibility graph with complex network features in the detection of epilepsy," *IEEE Access*, vol. 4, pp. 6554–6566, 2016.
[31] C. J. Stam and E. C. W. van Straaten, "The organization of physiological brain networks," *Clin. Neurophysiol.*, vol. 123, no. 6, pp. 1067–1087, Jun. 2012.
[32] G. Cheung, W.-T. Su, Y. Mao, and C.-W. Lin, "Robust semisupervised graph classifier learning with negative edge weights," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 712–726, Dec. 2018.
[33] H. Akira, *Complex-Valued Neural Networks*. Berlin, Germany: Springer-Verlag, 2012.
[34] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.
[35] M. Kobayashi, "Fast recall for complex-valued hopfield neural networks with projection rules," *Comput. Intell. Neurosci.*, vol. 2017, May 2017, Art. no. 4894278.
[36] M. Kobayashi, "Stability of rotor hopfield neural networks with synchronous mode," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 744–748, Mar. 2018.
[37] M. Kobayashi, "O(2)-valued hopfield neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3833–3838, Dec. 2019.
[38] Z. Uykan and R. Jantti, "Transmission-order optimization for bidirectional device-to-device (D2D) communications underlaying cellular TDD Networks—A graph theoretic approach," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 1–14, Jan. 2016.
[39] Z. Uykan and R. Jantti, "Joint optimization of transmission-order selection and channel allocation for bidirectional wireless links—Part II: Algorithms," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3991–4002, Jul. 2014.
[40] S. E. Schaeffer, *Survey: Graph Clustering, Computer Science Review I*. Amsterdam, The Netherlands: Elsevier, 2007, pp. 27–64.
[41] J. Bruck, "On the convergence properties of the hopfield model," *Proc. IEEE*, vol. 78, no. 10, pp. 1579–1585, Oct. 1990.

Zekeriya Uykan (Senior Member, IEEE) received the B.S. and M.S. degrees (Hons.) from Istanbul Technical University (ITU), Istanbul, Turkey, in 1993 and 1996, respectively, and the Lic.Tech. (Hons.) and Dr.Tech. degrees from the Helsinki University of Technology (HUT, now Aalto University), Espoo, Finland, in 1998 and 2001, respectively, all in electrical engineering.

He was a Research and Teaching Assistant with ITU, from April 1994 to September 1996, and also a Research Scientist with HUT, from September 1996 to February 2001. He was with the NOKIA Research Center (now Nokia-Bell Labs), Radio Comm. Laboratory, Helsinki, Finland, from July 2000 to April 2007, and Nokia Siemens Networks (now Nokia Solutions and Networks), Espoo, from April 2007 to December 2010. He was a Visiting Researcher with Sabanci University, Istanbul, Turkey, from May 2000 to July 2000; a Visiting Scholar with Stanford University, Stanford, CA, USA, from January 2002 to December 2002; a Visiting Scientist with Harvard University, Cambridge, MA, USA, from September 2008 to June 2009; and a Visiting Associate Professor with the COMNET Department, School of Electrical Engineering, Aalto University, Espoo, from August 2012 to September 2012 and from February 2013 to August 2013. He was with Doğuş University, Istanbul, from September 2009 to September 2015, and has been with the American University of the Middle East, Egaila, Kuwait, since September 2014. His research interests include artificial neural networks, artificial intelligence, and radio resource optimization in emerging wireless communication systems.