Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 1426212, 10 pages
https://doi.org/10.1155/2021/1426212

Research Article

A Simple Visualization Method for Three-Dimensional (3D) Network

Sangkwon Kim,[1] Chaeyoung Lee,[1] Jintae Park,[1] Sungha Yoon,[2] Yongho Choi,[3] and Junseok Kim[1]

[1]Department of Mathematics, Korea University, Seoul 02841, Republic of Korea
[2]Division of Applied Mathematical Sciences, Korea University, Sejong 30019, Republic of Korea
[3]Department of Mathematics and Big Data, Daegu University, Gyeongsan-si, Gyeongsangbuk-do 38453, Republic of Korea

Correspondence should be addressed to Junseok Kim; cfdkim@korea.ac.kr

Received 25 April 2021; Revised 10 May 2021; Accepted 19 May 2021; Published 27 May 2021

Academic Editor: Juan L. G. Guirao

The network is a concept that can be seen a lot in many areas of research. It is used to describe and interpret datasets in various fields such as social network, biological network, and metabolic regulation network. As a result, network diagrams appeared in various forms, and methods for visualizing the network information are being developed. In this article, we present a simple method with a weight of information data to visualize the network diagram for the three-dimensional (3D) network. The generic method of network visualization is a circular representation with many intersections. When dealing with a lot of data, the three-dimensional network graphics, which can be rotated, are easier to analyze than the two-dimensional (2D) network. The proposed algorithm focuses on visualizing three factors: the position and size of the nodes and the thickness of the edge between linked nodes. In the proposed method, an objective function is defined, which consists of two parts to locate the nodes: (i) a constraint for given distance, which is the weight of the relationship among all the data, and (ii) the mutual repulsive force among the given nodes. We apply the gradient descent method to minimize the objective function. The size of the nodes and the thickness of the edges are defined by using the weight of each node and the weight between other nodes associated with it, respectively. To demonstrate the performance of the proposed algorithm, the relationships of the characters in the two novels are visualized using 3D network diagram.

1. Introduction

The network is a vital concept to understand modern society structure and components of the society. The concept of the network has been applied not only to people but also to social network [1–7], biological network [8–12], metabolic regulation network [13], online social network [14, 15], sports social network [16, 17], structure of neural network [18–20], etc. As a result, the data for storing network information has become complex and diverse. In addition, the need to make complex data more recognizable has been essential, and efforts to meet such demands continue to this day. Therefore, a medium called a network diagram has emerged, which has become essential to understand network information. Network diagrams have appeared in various forms in two-dimensional (2D) and three-dimensional (3D) space. In addition, various methods and programs for drawing such diagrams or visualizing network information have appeared and developed.

In 2D space, the classical algorithm to draw the network automatically is the force-directed algorithm using the straight edges as springs [21]. This algorithm draws aesthetic graphs using only the information of a given graph. Teja and Yemula [22] modified and improved the force-directed algorithm to solve the limitation of the algorithm. Another algorithm for drawing graphs is stress majorization [23], which is widely used. Recently, the authors in [24] proposed an improved stress majorization method. Moreover, this method has an advantage that there is no need to solve the optimization problem with constraints. However, it is not easy to grasp the network in 2D space, and there are needs for the network plot in 3D space.

There are various tools and techniques for network visualization in 3D space. The author in [25] used a Java-based tool called CAVALIER (Communication and Activity VisuALIsation for the EnterRprise) to analyze the social network. In addition, the spring-embedding and alternative layout algorithm was introduced for network visualization. Ho et al. [26] studied the interactions between proteins in 3D diagrams. These visualizations help users to understand the biological process and relationships easily. In [27], the authors proposed a novel visualization tool named Arena3D that could address large scale and complex networks. They demonstrated the utility of Arena3D using network data relevant to Huntington disease. Paananen and Wong [28] presented a new visualization method based on real-time 3D force-directed graphs that can be used to discover new knowledge in the data. They also created and used a software tool which is called FORG3D. By using the given genomic data, they proved the functionality of FORG3D to visualize and explore unified genome-scale data. In [29], the authors developed the evidence network visualization algorithm with the exploration of covariate distribution for network meta-analysis (NMA) in 3D. This algorithm extended the z-axis to the generic 2D network to display the covariate bars for each trial, placing the covariate bars for each information data at the edge between the nodes involved.

In this article, we present a simple method with weight of information data to visualize the network diagram for 3D, and the rotatable 3D network diagram can be obtained to analyze data more easily. While the proposed method is an extension of the method applied in 2D [30], it is further improved in terms of the simplicity of the algorithm. In general, the problem is solved in a regular structure [31–36]; however, we visualized the network using the importance of information and concept of distance in the unstructured place. Furthermore, the position of nodes is relocated by minimizing an objective function which consists of repulsive and attractive energies. The gradient descent method is applied to minimize the objective function. Therefore, an optimal network diagram to the given data is obtained. The proposed method has advantages over other methods for visualizing networks in terms of simplicity.

The composition of this paper consists of the following order. Section 2 describes the proposed algorithm in detail. In Section 3, to demonstrate the performance of the proposed algorithm, 3D network diagram visualizations of the relationship of characters in two novels are presented. Finally, conclusions are made in Section 4.

2. Numerical Algorithm

2.1. Simple Network Visualization Algorithm. The purpose of the proposed algorithm for network visualization in 3D is to find optimal node positions $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ which minimize the following objective function:

$$\mathscr{E}(\mathbf{X}) = \frac{1}{2}\sum_{i<j}^{N} \Psi\left(w_{ij}\right)\left(\left|\mathbf{X}_i - \mathbf{X}_j\right| - d_{ij}\right)^2 - \gamma \sum_{i<j}^{N} \left|\mathbf{X}_i - \mathbf{X}_j\right|, \tag{1}$$

where w_{ij} is the normalized weight of the relationship between nodes $\mathbf{X}_i$ and $\mathbf{X}_j$ and $\Psi(w)$ is defined as $\Psi(w) = 1$ if $w > 0$; otherwise, $\Psi(w) = 0$. In addition, γ is the strength parameter that keeps $\mathbf{X}_i$ and $\mathbf{X}_j$ away from each other. For the simplicity, $0 \le w_{ij} \le 1$ is assumed. The distance function d_{ij} for w_{ij} can be used as follows [30]:

$$d_{ij} = d\left(w_{ij}\right) = \frac{1}{w_{ij}^{q}}, \quad \text{for } w_{ij} > 0, \tag{2}$$

where q is a nonnegative constant, and then, $d_{ij} \ge 1$. Let $D_{\max}$ and $D_{\min}$ be the maximum and minimum distances, respectively. Then, by the definition of the normalized weight, $D_{\min} = 1$, and for a given $D_{\max}$ value, the nonnegative constant q is defined as follows:

$$q = -\frac{\ln\left(D_{\max}\right)}{\ln\left(\min W\right)}, \tag{3}$$

where $\min W$ is the minimum positive value of w_{ij}, i.e.,

$$\min W = \min_{\substack{1 \le i,j \le N \\ w_{ij} > 0}} w_{ij}. \tag{4}$$

For example, if $\min W = 0.5$ and $D_{\max} = 3$, then q is obtained approximately as follows:

$$q = -\frac{\ln(3)}{\ln(0.5)} \approx 1.5850. \tag{5}$$

Figure 1 shows the distance function d_{ij} using given normalized weight w_{ij} and maximum distance $D_{\max}$ values.

Let $\{\mathbf{X}_1^n, \mathbf{X}_2^n, \ldots, \mathbf{X}_N^n\}$ be the set of positions of given nodes at iteration n. Figure 2 shows the effect of the first term of the objective function $\mathscr{E}$. There are two possible cases of forces at nodes $\mathbf{X}_i^n$ and $\mathbf{X}_j^n$: repulsive force when $|\mathbf{X}_i^n - \mathbf{X}_j^n| < d_{ij}$ and attractive force when $|\mathbf{X}_i^n - \mathbf{X}_j^n| > d_{ij}$.

The second term of the objective function $\mathscr{E}$ produces the mutual repulsive forces between $\mathbf{X}_i^n$ and $\mathbf{X}_j^n$, see Figure 3. The effect of this force places nodes with low weight outward in the network. In other words, nodes with small weights are positioned outside the main network diagram.

Then, we relocate the position of the node points as

$$X_i^{n+1} = X_i^n - \Delta t \frac{\partial \mathrm{E}}{\partial X_i}\left(X^n\right), \quad \text{for } 1 \le i \le N, \tag{6}$$

where Δt is an artificial temporal step and $(\partial \mathscr{E} / \partial \mathbf{X}_i)$ is the differentiation of equation (1) with respect to $\mathbf{X}_i$, i.e.,

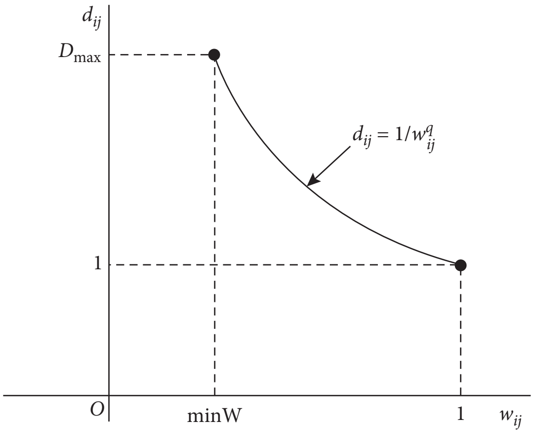
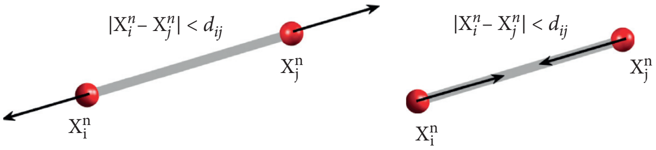
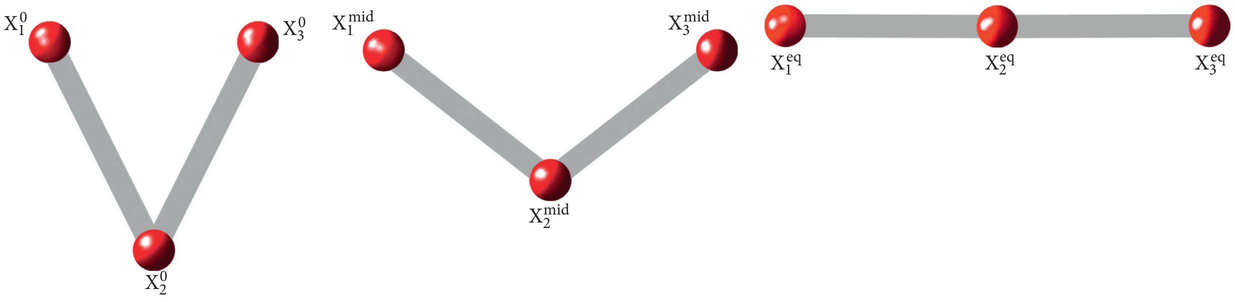


Figure 1: Illustration of distance function d_{ij} using the normalized weighting value w_{ij} of the relationship between nodes $\mathbf{X}_i$ and $\mathbf{X}_j$. Here, $q = -(\ln(D_{\max})/\ln(\min W))$.



(a) (b)

Figure 2: Two possible forces at nodes $\mathbf{X}_i^n$ and $\mathbf{X}_j^n$: (a) repulsive force when $|\mathbf{X}_i^n - \mathbf{X}_j^n| < d_{ij}$ and (b) attractive force $|\mathbf{X}_i^n - \mathbf{X}_j^n| < d_{ij}$ for the given d_{ij} value.



(a) (b) (c)

Figure 3: Mutual repulsive force among nodes: (a) initial state, (b) middle state, and (c) equilibrium state.

$$\frac{\partial \mathscr{E}}{\partial \mathbf{X}_i}(\mathbf{X}^n) = \sum_{j=1}^{N} \Psi(w_{ij})\left(\left|\mathbf{X}_i^n - \mathbf{X}_j^n\right| - d_{ij}\right)\frac{\mathbf{X}_i^n - \mathbf{X}_j^n}{\left|\mathbf{X}_i^n - \mathbf{X}_j^n\right|} - \gamma \sum_{j=1}^{N} \frac{\mathbf{X}_i^n - \mathbf{X}_j^n}{\left|\mathbf{X}_i^n - \mathbf{X}_j^n\right|}. \quad (7)$$

Once the positions of the nodes are renewed, then the network diagram is plotted automatically. The iterative process will be terminated if the process reaches an equilibrium state, i.e.,

$$\left\|\mathbf{X}^{n+1} - \mathbf{X}^n\right\|_2 < \text{tol}, \quad (8)$$

for $n \geq N_t$ and for some N_t. For example, let us consider four points $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_4$. Furthermore, assume that the weighting matrix $\mathbf{W}$ between $\mathbf{X}_i$ and $\mathbf{X}_j$ is given as

$$\mathbf{W} = \begin{pmatrix} 0 & 2 & 3 & 1 \\ 2 & 0 & 2 & 1 \\ 3 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \quad (9)$$

Let the matrix $\mathbf{W}$ be redefined by dividing all the elements of $\mathbf{W}$ by the largest value among the elements of the given $\mathbf{W}$. That is,

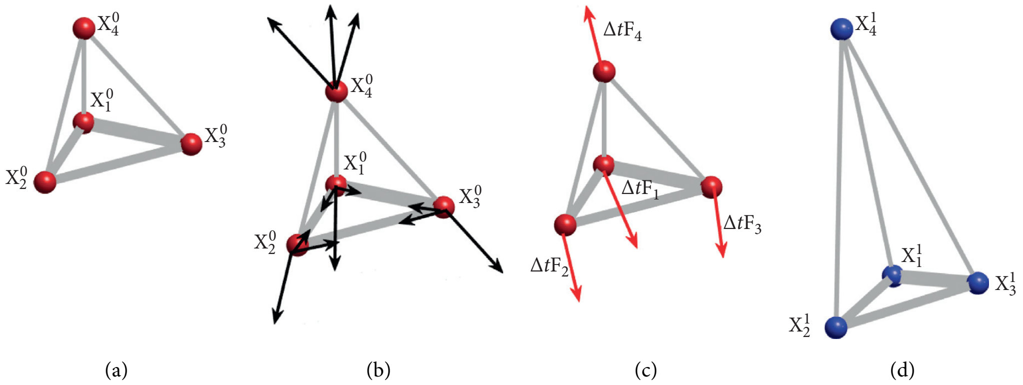


Figure 4: Schematic illustration of the proposed algorithm. (a) Initial state of four nodes. (b) Direction of forces acting on each node. (c) Net force direction of each point. (d) Position of four nodes after 1 iteration using the proposed method. Here, the normalized weight matrix **W** in equation (10) is used.

$$\mathbf{W} = \begin{pmatrix} 0 & \frac{2}{3} & 1 & \frac{1}{3} \\ \frac{2}{3} & 0 & \frac{2}{3} & \frac{1}{3} \\ 1 & \frac{2}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix}. \quad (10)$$

We take $D_{\max} = 4$ and get $q = 1.2619$ because $\min W = (1/3)$:

$$\begin{pmatrix} & d_{12} & d_{13} & d_{13} \\ d_{21} & & d_{23} & d_{13} \\ d_{31} & d_{32} & & d_{13} \\ d_{41} & d_{42} & d_{43} & \end{pmatrix} = \begin{pmatrix} & 1.67 & 1.0 & 4.0 \\ 1.67 & & 1.67 & 4.0 \\ 1.0 & 1.67 & & 4.0 \\ 4.0 & 4.0 & 4.0 & \end{pmatrix}. \quad (11)$$

Let $\mathbf{X}_1^0 = (0,0,0)$, $\mathbf{X}_2^0 = (1,0,0)$, $\mathbf{X}_3^0 = (0,1,0)$, and $\mathbf{X}_4^0 = (0,0,1)$ be the initial positions of four nodes, where the superscript 0 denotes the starting index. We use $\Delta t = 0.2$, $\gamma = 0.01$, and tol $= 0.001$. In Figure 4, the node markers and edges are depicted by sphere and gray lines, respectively. Here, the red and blue spheres mean initial and after 1 iteration positions, respectively. Figure 4(a) shows the initial state consisting of four points with red spheres and gray edges of linked nodes using scaled **W**. Note that the values of each element of **W** are represented by the thickness of connecting lines in the diagram. In Figure 4(b), black arrows indicate the force direction acting on each node. The red arrows in Figure 4(c) are net force vectors $\mathbf{F}_1^0$, $\mathbf{F}_2^0$, $\mathbf{F}_3^0$, and $\mathbf{F}_4^0$. Using net force vectors, we update the positions of nodes as follows:

$$\mathbf{X}_i^1 = \mathbf{X}_i^0 + \Delta t \mathbf{F}_i^0, \quad \text{for } i = 1,2,3, \text{ and } 4. \quad (12)$$

Figure 4(d) shows the network diagrams after 1 iteration. The nodes are initially located in a tetrahedron, and the network diagram is drawn according to the given weights.

So far we have only considered the positions of nodes, now we discuss the visualization of nodes and edges. In general, nodes and edges are visualized as unit sphere shape and straight lines with a fixed thickness. We visualize the size of the nodes S_i and the thickness of edges T_{ij} according to the weight **W** of each node as follows:

$$S_i = s\sqrt{\sum_{j=1}^{N} w_{ij}}, \qquad T_{ij} = t w_{ij}^2 + \Psi\left(w_{ij}\right), \quad (13)$$

where s and t are positive constants. Characters with relatively high activity are indicated by larger nodes; meanwhile, relatively intimate relationships are depicted by thicker edges.

3. Numerical Results

Numerical simulations are conducted to demonstrate the performance of the proposed algorithm and to visualize the relationship examples of characters in two novels in the 3D network diagram. The first example is "The Venice Merchant" and the second example is "Romeo and Juliet," which are William Shakespeare's representative plays. The elements of the weighting matrix are set based on the number of conversations between the characters in each novel.

In the first novel, let $N = 19$ of the characters to make the weighting matrix **W**, which is provided in Appendix. Each element w_{ij} of **W** means the cumulative number of dialogues between characters $\mathbf{X}_i$ and $\mathbf{X}_j$. The parameters are given as $\Delta t = 0.2$, $\gamma = 0.01$, $D_{\max} = 5$, and tol $= 0.001$. Then, $q \approx 0.4363$ is obtained. As shown in Figure 5, the motion of nodes and edges can be observed from the initial state where the nodes are randomly located until the optimal network diagram is found by applying the proposed algorithm repeatedly. The proposed algorithm for "The Venice Merchant" data reached the equilibrium state at 828 iterations.

Figure 6 shows a diagram of the network in the equilibrium of the iterative algorithm for "The Venice Merchant" dataset. The equilibrium state means that all nodes are properly located according to the weights of a given network.

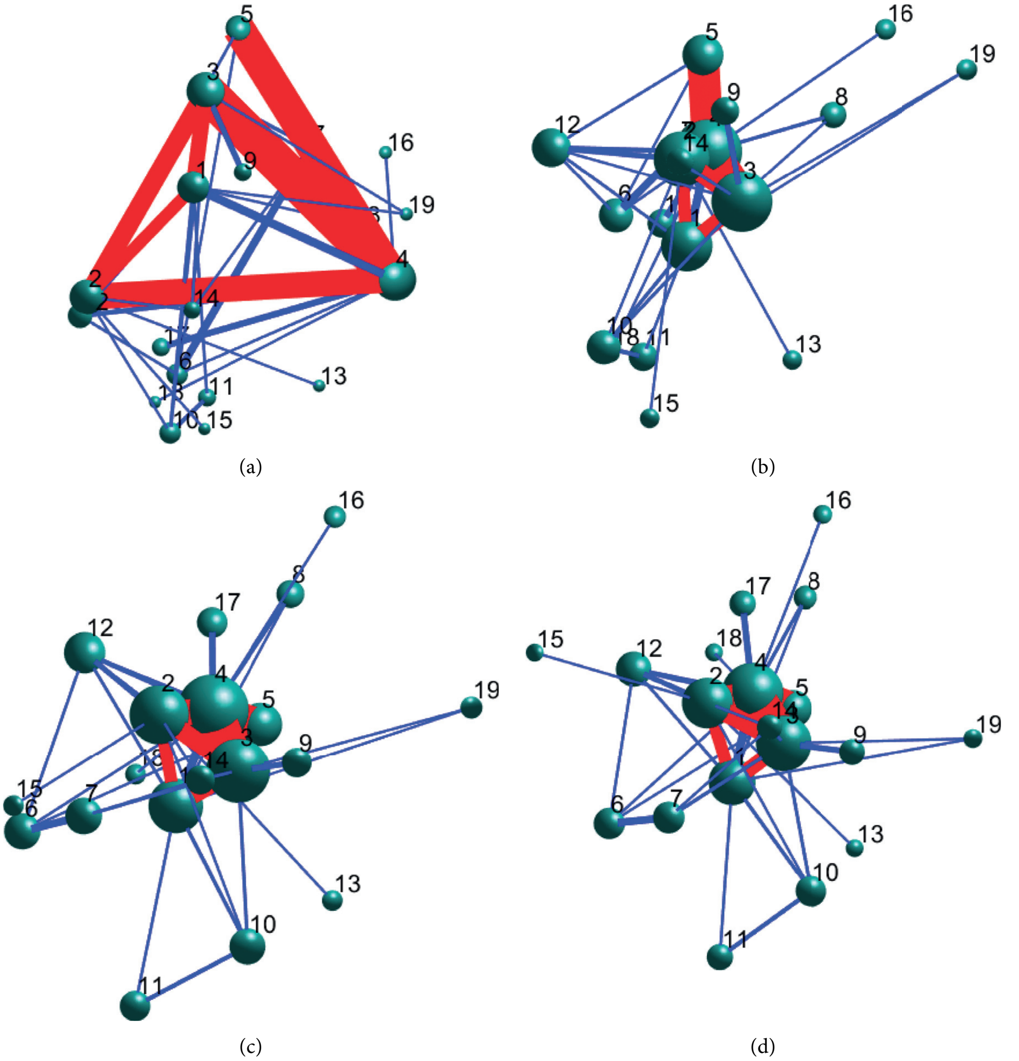


FIGURE 5: Snapshots of the process applying the proposed algorithm for “The Venice Merchant.” (a) Initial state of 19 nodes at $n = 0$. (b) Intermediate state of nodes at $n = 10$. (c) Intermediate state of nodes at $n = 400$. (d) Equilibrium state of nodes at $n = 828$.

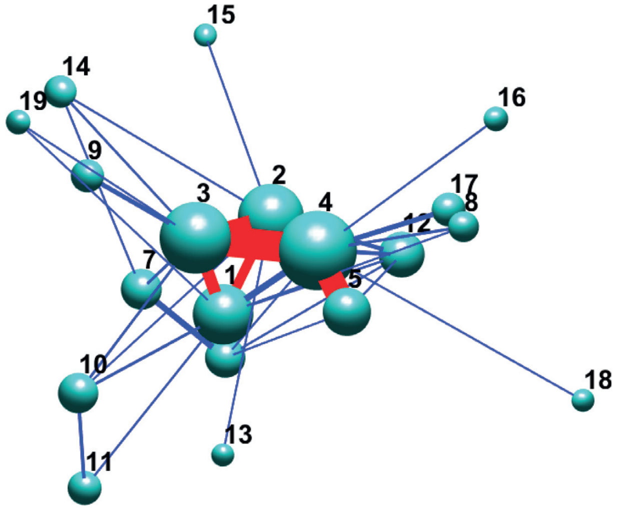


FIGURE 6: Magnified network plot of the equilibrium state from another directional view for “The Venice Merchant.” Large-sized nodes are linked together with thicker lines, and small-sized nodes are linked with thinner lines and located around the large-sized nodes.

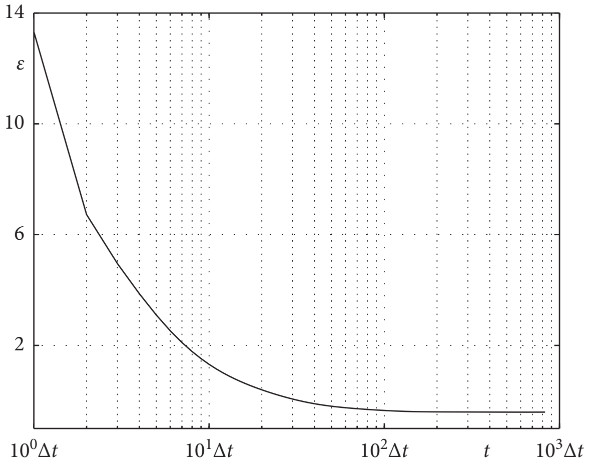
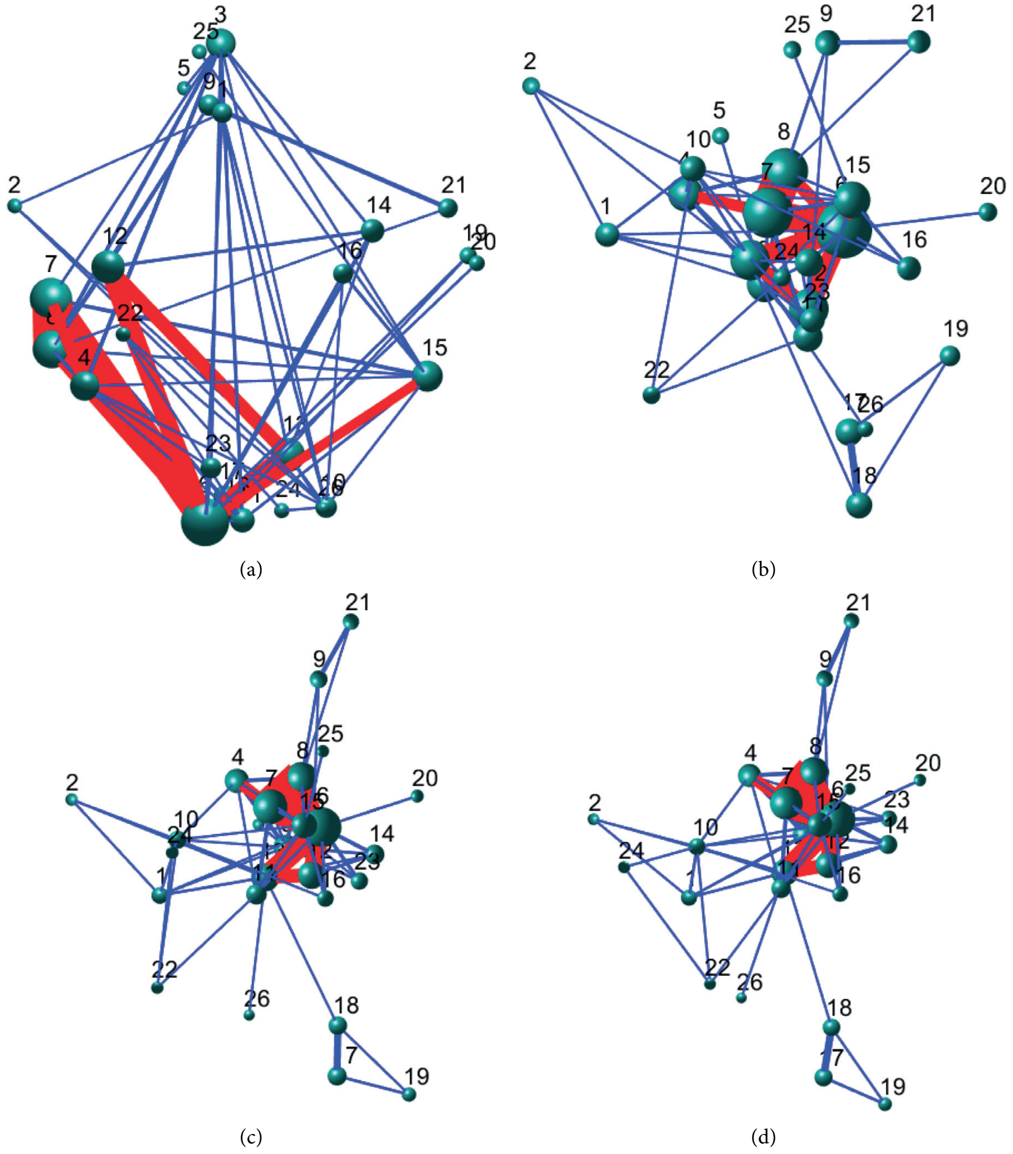


Figure 7: Temporal evolution of the objective function $\mathscr{E}$ with respect to the iteration of the algorithm for "The Venice Merchant" case.



(a)

(b)

(c)

(d)

Figure 8: Snapshots of the process applying the proposed algorithm for "Romeo and Juliet." (a) Initial state of 26 nodes at $n = 0$. (b) Intermediate state of nodes at $n = 25$. (c) Intermediate state of nodes at $n = 310$. (d) Equilibrium state of nodes at $n = 617$.

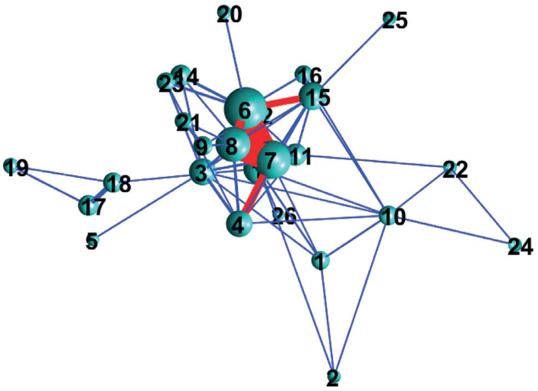


Figure 9: Magnified network plot of the equilibrium state from another directional view for “Romeo and Juliet.” Large-sized nodes are linked together with thicker lines, and small-sized nodes are linked with thinner lines and located around the large-sized nodes.

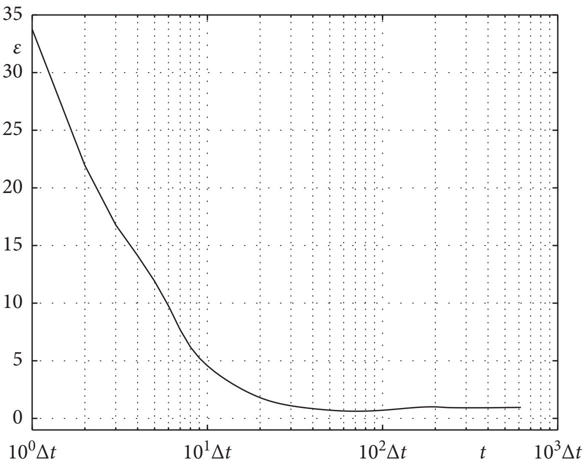


Figure 10: Temporal evolution of the objective function $\mathscr{E}$ with respect to the iteration of the algorithm for the “Romeo and Juliet” case.

Assortativity refers to the tendency of nodes in a network being connected with similar sized nodes. Therefore, highly assortativity networks are clearly shown to have a core periphery structure between nodes with large and small sizes. These characteristics mean that the basic structure of the interaction that makes up the story of the novel is centered on the main characters. Figure 6, therefore, shows that “The Venice Merchant” has a high assortativity, that is, a novel property that revolves around the main character. As the nodes move, we can confirm that the objective function $\mathscr{E}$ decreases, as shown in Figure 7.

As a second example, let $N = 26$ of the characters to set the weighting matrix $\mathbf{W}$ which is based on the number of conversations between 26 characters in “Romeo and Juliet.” This data is provided in Appendix. The parameters are given as $D_{\max} = 5$ and tol $= 0.001$, and then, the value of q is about 0.3652. In addition, time step $\Delta t = 0.2$ and $\gamma = 0.01$ are used. Figure 8 shows the process of finding the optimal network plot by applying the proposed algorithm. The proposed algorithm for “Romeo and Juliet” data reached the equilibrium state at 617 iterations.

Table 1: Name of the character corresponding to each number.

Number	Character name	
	The Merchant of Venice	Romeo and Juliet
1	Antonio	Montague
2	Bassanio	Lady Montague
3	Shylock	Capulet
4	Portia	Lady Capulet
5	Nerissa	Second Capulet
6	Lorenzo	Romeo
7	Jessica	Juliet
8	Duke	Nurse
9	Tubal	Peter
10	Salerio	Prince Escalus
11	Salanio	Paris
12	Gratiano	Mercutio
13	Leonardo	Benvolio
14	Launcelot Gobbo	Tybalt
15	Old Gobbo	Friar Laurence
16	Stephano	Balthasar
17	Prince of Morocco	Sampson
18	Prince of Arragon	Gregory
19	Servant	Abraham
20		Apothecary
21		Musician
22		Page
23		Servant
24		Watchman
25		Friar john
26		Citizen

Figure 9 shows the network plot of the equilibrium state of the iterative algorithm for the “Romeo and Juliet” dataset. From the results, we can quickly and easily grasp the relationship among characters and see the high assortativity. Furthermore, we can confirm that the objective function $\mathscr{E}$ decreases and converges during the process of finding the optimal position for all nodes. Figure 10 depicts the decreasing of the objective function.

4. Conclusion

We proposed a simple method for 3D network diagrams in this article. The proposed method had the advantage of being drawn only with information from a given network. We used the distance function based on the network information and objective function. To minimize the proposed objective function, we used the gradient descent method and confirmed the energy dissipation. According to the given relationship, the network diagrams own various characteristics: (i) different sizes of nodes and (ii) different thickness of edges. Through specific examples, we have known that these features help us to grasp network information at a glance. Furthermore, our algorithm is simple to understand and implement. Finally, the MATLAB source code of our algorithm is provided for the interested readers to try.

Social distancing and contact tracing can contribute to curbing the spread of the COVID-19 virus. Research on

SARS-COV-2 transmission control strategies in actual social networks using GPS or mobile data is actively being conducted. In future works, we plan to use the proposed algorithm to present a contact network diagram. Using this diagram, we can visualize the contact network for the dynamics of the COVID-19 spreading.

Appendix

In this appendix, we provide the MATLAB source codes of the proposed algorithm for novel data as "The Merchant of Venice" and "Romeo and Juliet," and Table 1 describes each character name and corresponding node number. The source code is available on the following website: http://elie.Korea.ac.kr/~cfdkim/codes/.

Here, the weights of each novel data were presented as W_1 and W_2 in the source code.

clear;

$$W_1 = \begin{bmatrix}
0 & 21 & 24 & 16 & 0 & 0 & 0 & 2 & 0 & 7 & 4 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
21 & 0 & 27 & 32 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 11 & 2 & 3 & 2 & 0 & 0 & 0 & 0 \\
24 & 27 & 0 & 40 & 0 & 0 & 7 & 0 & 12 & 6 & 0 & 2 & 0 & 5 & 0 & 0 & 0 & 0 & 2 \\
16 & 32 & 40 & 0 & 36 & 3 & 0 & 7 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 3 & 13 & 2 & 0 \\
0 & 0 & 0 & 36 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 7 & 3 & 2 & 0 & 15 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 11 & 2 & 10 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$W_2 = \begin{bmatrix}
0 & 3 & 3 & 0 & 0 & 1 & 0 & 0 & 0 & 9 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 18 & 5 & 0 & 8 & 17 & 0 & 1 & 14 & 0 & 0 & 9 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\
0 & 0 & 18 & 0 & 0 & 0 & 42 & 17 & 0 & 3 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 71 & 45 & 0 & 0 & 6 & 56 & 60 & 17 & 38 & 10 & 0 & 0 & 0 & 8 & 0 & 0 & 16 & 0 & 0 & 0 \\
0 & 0 & 8 & 42 & 0 & 71 & 0 & 82 & 0 & 0 & 17 & 0 & 0 & 0 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 17 & 17 & 0 & 45 & 82 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 1 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & 2 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 14 & 2 & 0 & 6 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 2 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 56 & 0 & 7 & 1 & 0 & 0 & 0 & 47 & 13 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 5 & 0 & 0 \\
9 & 1 & 0 & 0 & 0 & 60 & 0 & 0 & 0 & 4 & 0 & 47 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 9 & 0 & 0 & 17 & 0 & 0 & 0 & 0 & 0 & 13 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 5 & 2 & 0 & 38 & 17 & 8 & 0 & 4 & 9 & 0 & 0 & 0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 30 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 10 & 2 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

```
flag = 1;
if flag = = 1
W = W1;
else if flag = = 2
W = W2;
end
W = W/max (max(W)); minW = min(min(W(W > 0)));
N = size(W,1); lbD = 1; ubD = 5;
q = −log (ubD)/log(minW); rand("seed",3773);
t1 = rand(N,1); t2 = rand(N,1);
xy = ubD * [sin(pi * t1) * cos (2 * pi * t2), sin (pi *
t1) * sin (2 * pi * t2), cos (pi * t1)];
```

```
for i = 1:N
  for j = 1:N
    if W(i, j) > 0
    d (i, j) = 1/W(i, j)ˆq;
    end
  end
end
dt = 0.2; gam = 0.01; tol = 0.001; error = 2 * tol; n = 0;
while error ≥ tol
  n = n + 1; F = zeros(N, 3);
    for i = 1 : N
      for j = i + 1 : N
        if W(i, j) > 0
        vt = xy (i, :)-xy (j, :);
        F (i, :) = F(i, :) - (norm(vt)-d(i, j)) * vt/norm
(vt);
        F (j, :) = F(j, :) + (norm(vt)-d(i, j)) * vt/norm
(vt);
        end
      end
    end
  for i = 1:N
    for j = 1:N
      if j~ = i
      vt = xy (i, :)-xy(j, :);
      F(i, :) = F(i, :) + gam * vt/norm(vt);
      end
    end
  end
  xy = xy + dt * F; error = norm(dt * F)/sqrt(N);
  if mod(n, 30) = = 0 \Vert error < tol
  DrawNetwork(xy, W); pause(0.1)
  end
end
function DrawNetwork(xy,W)
n = 20; x = linspace(−1.1, 1.1, n); y = x; z = x;
[xx, yy, zz] = meshgrid(x, y, z); u(1:n, 1:n, 1:n) = 0;
for i = 1:n
  for j = 1:n
    for k = 1:n
    u(i, j, k) = 1-sqrt(x(i)ˆ2y(j)ˆ2z(k)ˆ2);
    end

  end

end
N = length (xy); clf;
for i = 1 : N
s = 0.2 * sum (W(i,:))ˆ0.5 + 0.1;
```

```
isosurface (s ∗ xx + xy(i,1),s ∗ yy + xy(i,2),s ∗ zz + xy(i,3),u,0); hold on
end
camlight; lighting phong;
for i = 1:N
   for j = i+1:N
   if W(i, j) > 0.4
   plot3 (xy ([i, j], 1), xy ([i, j], 2), xy ([i, j], 3), “r”, “linewidth,” 15 ∗ W (i, j)̂2 + 1);
   elseif W (i, j) > 0
   plot3 (xy ([i, j], 1), xy ([i, j], 2), xy ([i, j], 3), “b”, “linewidth”, 15 ∗ W(i, j)̂2 + 1);
   end
   end
end
for i = 1:N
s = 0.2 ∗ sum (W(i , :))̂0.5 + 0.4; a = 0.15; b = 0;
text (xy(i, 1)-a,xy(i, 2)-b, xy (i, 3) + s, num2str (i), “fontsize,”10);
end
axis off; axis image;
end
```

Data Availability

We have released the code and data of this paper in the Appendix and the corresponding author’s homepage.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The corresponding author (J.S. Kim) expresses thanks for the support from the BK21 FOUR Program.

References

[1] J. Scott, “Social network analysis,” *Sociology*, vol. 22, no. 1, pp. 109–127, 1988.

[2] S. Marcus, M. Moy, and T. Coffman, “Social network analysis,” *Mining Graph Data*, pp. 443–467, 2007.

[3] J. Lee, G. Noh, G. Oh, and C. K. Kim, “Trustor clustering with an improved recommender system based on social relationships,” *Information Systems*, vol. 77, pp. 18–128, 2018.

[4] P. De Meo, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, “Providing recommendations in social networks by integrating local and global reputation,” *Information Systems*, vol. 78, pp. 58–67, 2018.

[5] C. Zhan, D. Zhang, Y. Wang, D. Lin, and H. Wang, “IES-backbone: an interactive edge selection based backbone method for small world network visualization,” *IEEE Access*, vol. 7, pp. 166415–166427, 2019.

[6] W. Didimo, L. Grilli, G. Liotta, L. Menconi, F. Montecchiani, and D. Pagliuca, “Combining network visualization and data mining for tax risk assessment,” *IEEE Access*, vol. 8, pp. 16073–16086, 2020.

[7] H. Yuan, Y. Song, J. Hu, and Y. Ma, “Design of festival sentiment classifier based on social network,” *Computational Intelligence and Neuroscienc*, vol. 2020, Article ID 8824009, 10 pages, 2020.

[8] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[9] G. Jia, Z. Cai, M. Musolesi et al., “Community detection in social and biological networks using differential evolution,” *Lecture Notes in Computer Science*, pp. 71–85, 2012.

[10] J. Lee and J. Lee, “Hidden information revealed by optimal community structure from a protein-complex bipartite network improves protein function prediction,” *PLoS One*, vol. 8, no. 4, Article ID e60372, 2013.

[11] Y. Atay, I. Koc, I. Babaoglu, and H. Kodaz, “Community detection from biological and social networks: a comparative analysis of metaheuristic algorithms,” *Applied Soft Computing*, vol. 50, pp. 194–211, 2017.

[12] S. Shirakol, M. Kalyanshetti, and S. M. Hosamani, “QSPR analysis of certain distance based topological indices,” *Applied Mathematics and Nonlinear Sciences*, vol. 4, no. 2, pp. 371–386, 2019.

[13] J. Krasensky and C. Jonak, “Drought, salt, and temperature stress-induced metabolic rearrangements and regulatory networks,” *Journal of Experimental Botany*, vol. 63, no. 4, pp. 1593–1608, 2012.

[14] R. Ghazzali, “Modeling a rumor propagation in online social network: an optimal control approach,” *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 6724815, 12 pages, 2020.

[15] A. Bhih, “A discrete mathematical Modeling and optimal control of the rumor propagation in online social network,” *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 4386474, 12 pages, 2020.

[16] T. Zhao, N. Cui, Y. Chen, and M. Li, “Efficient strategy mining for football social network,” *Complexity*, vol. 2020, Article ID 8823189, 11 pages, 2020.

[17] J. Zhang, “Analysis and modeling of football team’s collaboration mode and performance evaluation using network science and BP neural network,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 7397169, 13 pages, 2020.

[18] M. Umar, Z. Sabir, M. A. Z. Raja, F. Amin, T. Saeed, and Y. Guerrero-Sanchez, “Integrated neuro-swarm heuristic with interior-point for nonlinear SITR model for dynamics of novel COVID-19,” *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 2811–2824, 2021.

[19] Z. Sabir, M. A. Z. Raja, H. A. Wahab, G. C. Altamirano, Y.-D. Zhang, and D.-N. Le, “Integrated intelligence of neuro-evolution with sequential quadratic programming for second-order Lane-Emden pantograph models,” *Mathematics and Computers in Simulation*, vol. 188, pp. 87–101, 2021.

[20] Z. Sabir, C. M. Khalique, M. A. Z. Raja, and D. Baleanu, “Evolutionary computing for nonlinear singular boundary value problems using neural network, genetic algorithm and active-set algorithm,” *The European Physical Journal*, vol. 136, no. 2, pp. 1–19, 2021.

[21] P. Eades, “A heuristic for graph drawing,” *Congruent Number*, vol. 42, pp. 149–160, 1984.

[22] S. C. Teja and P. K. Yemula, “Power network layout generation using force directed graph technique,” in *Proceedings of*

the 2014 Eighteenth National Power Systems Conference (NPSC) IEEE, pp. 1–6, Guwahati, India, 2014.

[23] E. R. Gansner, Y. Koren, and S. North, “Graph drawing by stress majorization,” in *Proceedings of the International Symposium on Graph Drawing*, pp. 239–250, New York, NY, USA, 2004.

[24] Y. Wang, Y. Wang, Y. Sun et al., “Revisiting stress majorization as a unified framework for interactive constrained graph visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 489–499, 2017.

[25] A. Dekker, “Visualisation of social networks using CAVALIER,” *ACM*, vol. 16, pp. 49–55, 2001.

[26] E. Ho, R. Webber, and M. R. Wilkins, “Interactive three-dimensional visualization and contextual analysis of protein interaction networks,” *Journal of Proteome Research*, vol. 7, no. 1, pp. 104–112, 2008.

[27] G. A. Pavlopoulos, S. I. O’Donoghue, V. P. Satagopam, T. G. Soldatos, E. Pafilis, and R. Schneider, “Arena3D: visualization of biological networks in 3D,” *BMC Systems Biology*, vol. 2, no. 1, p. 104, 2008.

[28] J. Paananen and G. Wong, “FORG3D: force-directed 3D graph editor for visualization of integrated genome scale data,” *BMC Systems Biology*, vol. 3, no. 1, p. 26, 2009.

[29] S. Batson, R. Score, and A. J. Sutton, “Three-dimensional evidence network plot system: covariate imbalances and effects in network meta-analysis explored using a new software tool,” *Journal of Clinical Epidemiology*, vol. 86, pp. 182–195, 2017.

[30] J. Park, S. Yoon, C. Lee, and J. Kim, “A simple method for network visualization,” *Mathematics*, vol. 8, no. 6, p. 1020, 2020.

[31] H. M. Baskonus, H. Bulut, and T. A. Sulaiman, “New complex hyperbolic structures to the Lonngren-wave equation by using sine-gordon expansion method,” *Applied Mathematics and Nonlinear Sciences*, vol. 4, no. 1, pp. 141–150, 2019.

[32] M. Modanli and A. Akgül, “On Solutions of Fractional order Telegraph partial differential equation by Crank-Nicholson finite difference method,” *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 1, pp. 163–170, 2020.

[33] E. Ilhan and I. O. Kıymaz, “A generalization of truncated M-fractional derivative and applications to fractional differential equations,” *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 1, pp. 171–188, 2020.

[34] E. I. Eskitasçıoglu, M. B. Aktas, and H. M. Baskonus, “New complex and hyperbolic forms for Ablowitz–Kaup–Newell–Segur wave equation with fourth order,” *Applied Mathematics and Nonlinear Sciences*, vol. 4, no. 1, pp. 105–112, 2019.

[35] H. Duru, O. Tasbozan, and A. K. de Griñó, “New analytical solutions of conformable time fractional bad and good modified Boussinesq equations,” *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 1, pp. 447–454, 2020.

[36] H. Duru, A. Kurt, and O. Tasbozan, “New travelling wave solutions for KdV6 equation using sub equation method,” *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 1, pp. 455–460, 2020.