Journal of Visual Languages and Computing 21 (2010) 194–208

Contents lists available at ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

# A new concentric-circle visualization of multi-dimensional data and its application in network security

Liang Fu Lu [a], Jia Wan Zhang [b,*], Mao Lin Huang [c], Lei Fu [b]

[a] Mathematics Department, Tianjin University, Tianjin, PR China
[b] School of Computer Science and Technology, Tianjin University, Tianjin, PR China
[c] Faculty of Engineering and IT, University of Technology, Sydney, Australia

ARTICLE INFO

*Keywords:*
Concentric-circle coordinate
Multi-dimensional data visualization
Crossing reduction
Network visualization
Security visualization
Network intrusion detection
Polycurve

ABSTRACT

With the rapid growth of networked data communications in size and complexity, network administrators today are facing more challenges to protect their networked computers and devices from all kinds of attacks. This paper proposes a new concentric-circle visualization method for visualizing multi-dimensional network data. This method can be used to identify the main features of network attacks, such as DDoS attack, by displaying their recognizable visual patterns. To reduce the edge overlaps and crossings, we arrange multiple axes displayed as concentric circles rather than the traditional parallel lines. In our method, we use polycurves to link values (vertexes) rather than polylines used in parallel coordinate approach. Some heuristics are applied in our new method in order to improve the readability of views. We discuss the advantages as well as the limitations of our new method. In comparison with the parallel coordinate visualization, our approach can reduce more than 15% of the edge overlaps and crossings. In the second stage of the method, we have further enhanced the readability of views by increasing the edge crossing angle. Finally, we introduce our prototype system: a visual interactive network scan detection system called CCScanViewer. It is based on our new visualization approach and the experiments have showed that the new approach is effective in detecting attack features from a variety of networking patterns, such as the features of network scans and DDoS attacks.


## 1. Introduction

Today networking systems and data communications are becoming more and more popular in our society [1]. However, there is no absolute way to secure the data and data transformations in large scale networking systems. The existing techniques and tools of securing a network system still rely heavily on human experiences. Most of them require human involvement in analyzing and detecting anomalies and intrusions. To enhance the human perception and understanding of different types of network intrusions and attacks, network visualization has become a hot research field in recent years that attempts to speed up the intrusion detection process through the visual analytics. Unlike the traditional methods of analyzing textual log data, visualization approach has been proven that can increase the efficiency and effectiveness of network intrusion detection significantly by the reduction of human cognition process. Visualization cannot only help analysts to deal with the large volume of analytical data by taking the advantage of computer graphics, but also help network administrators to detect anomalies through visual pattern recognition. It can even be used for discovering new types of attacks and forecasting the trend of unexpected events [50].

The aim of intrusion detection is to accurately detect as many attacks as possible. In early years, many researchers were focusing on the detection of the most common

* Corresponding author.
*E-mail address:* zhangjiawan@gmail.com (J. Wan Zhang).

1045-926X/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jvlc.2010.05.002

intrusions, such as network scans. However, in recent years there are many other types of network intrusions that have been created. In order to gain more knowledge about other types of network intrusions, it would be beneficial to start with the analysis of network scans. The purpose of scanning a network is usually to determine what exists on the network. Some visualization techniques and tools have been applied recently for detecting hostile attacks and scans [2–8,18]. However, these existing techniques are focusing on how to discover the abnormal visual structures through the monitoring of a large volume of network traffic data. They seldom consider the display of features of network events, intrusions and transactions through the network data analysis. These existing techniques only display the timing of network scans, and they are not specifically designed for detecting particular network attacks, such as DDoS (Distributed Denial of Service) attacks through the analytical visualization. In general, DDoS attacks are very difficult to be detected mainly because they are hidden in a large volume of network traffic data. Moreover, the network traffic data are usually represented in multi-dimensional manner and the visualization of multi-dimensional data is still a challenging problem in the visualization community. Among the existing multi-dimensional visualization techniques, parallel coordinate [9] is one of the most popular methods, in which the values of different dimensions are indicated one-to-one to an equal number of parallel axes and each dataset is mapped as a polyline intersecting the parallel axes at points which represent values of the individual data dimensions (see Fig. 1(a)). Unfortunately, as the volume of data increases, a large number of line crossings and overlaps among these polylines would produce unreadable images that suffer from excessive visual clutter. Several improvements [29,31,32] have been proposed to reduce the quantity of displayed elements. These methods either focus on the interactive techniques [10,47] or rely on the data preprocessing techniques such as clustering [11–14] and aggregation [15].

This paper proposes a novel multi-dimensional visualization method called concentric-circle visualization, in which axes are organized as concentric circles rather than parallel lines. We used it to display the main characteristics and patterns of network scans and DDoS attacks. The polylines are transformed into segments of curves to represent each dimensional space (see Fig. 1(b)). Through our theoretical analysis, it has been proved that comparing to parallel layout concentric-circle fashion can reduce 33% of the line crossings if the raw data could be transferred into complete bipartite graph, and for common data, some heuristics in horizontal crossing reduction are applied to solve the same problem in our concentric-circle fashion. Our experiment shows that the new approach can reduce extra 15% more of the line crossings in comparison with the traditional parallel coordinate scheme. Furthermore, the new method can increase the crossing angles partially that improves the readability of views. A visual interactive real time detection system called CCScanViewer has been developed based on the new visualization techniques. In our system, abnormal network activities are extracted from a large volume of network flows and their patterns. Experiments show that the new approach is effective in various applications for monitoring and detecting network intrusions. By using our approach, we can easily detect the unusual patterns from network scans, port scans, the hidden scans, and DDoS attacks etc.

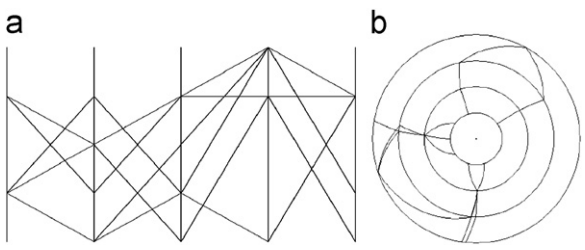


**Fig. 1.** The layout of parallel coordinates and concentric-circle coordinates: (a) Parallel Coordinates and (b) Concentric-circle Coordinates.

The rest of the paper is organized as follows. In Section 2 we give an overview of existing techniques in visual network security and the methods for reducing the overlapping and crossings in parallel coordinates technique. Section 3 provides the general layout of concentric-circle coordinates, and the new method is proposed theoretically to reduce the overlapping and crossings. Comparison between parallel coordinates and concentric-circle coordinates is discussed in Section 4. Case studies are presented in Section 5. Finally, we give the conclusions and future work in Section 6.

## 2. Related work

Network security visualization (or visual network analytics) is a subfield of information visualization. The main steps include: (1) data collection and preprocessing, (2) visual mapping, and (3) graphics generation. Many information visualization researchers are focusing on the last two steps [26,30,46,48]. To be able to map the multiple dimensional network data into the geometrical plane, various multi-dimensional visualization techniques have been developed.

Parallel Coordinate technique is one of the most popular methods has been widely used. By using parallel axes, *n*-dimensional data can be represented in a 2-dimensional plane. Although parallel coordinate method has been proven to be an effective tool for displaying multi-dimensional data, the edge crossings (or visual-clutters) are still the main problem of this approach that reduces readability of the underlying patterns in large datasets.

Several enhancements have been proposed to overcome the drawback of parallel coordinate visualization. For example, Zhou et al. [14] convert the straight-line edges into curves to reduce the visual clutter in clustered visualization. They also used the splatting framework [13] to detect clusters and reduce visual clutter. Yuan et al. [12] combined the parallel coordinate method with the scatter-plots method to reduce the visual clutter. It plots scattering points in parallel coordinates directly with a

seamless transition between them. The shapes of polylines are remodeled to cooperate with the scattering points, resulting in the diminution of their inherent visual effects.

Another popular multi-dimensional visualization method is Space-Filling Curve (SFCs) approach [7,24]. It maps the multi-dimensional space into one dimensional space. Mokbel et al. [24] introduced a description vector $v$ to encapsulate the properties of SFCs. The SFC is divided into a set of connected segments by this means, so quantificational experiments can be done to compare different SFCs. SFCs are utilized to map the collected statistics to images representing traffic activities in [7]. The enhanced locality of SFC clustering was used to identify anomalies such as large scale DDoS attacks.

To reduce the visual clutter in multi-dimensional visualization, Peng et al. proposed that the reordering of axes can reduce clutter and reveal relationships among the data in parallel coordinates [25].

To increase the readability of visualization, Ellis and Dix [27,28] proposed a sampling lens (focus+context) technique to reduce the display density. There have been some other efforts to improve the readability of multi-dimensional visualization through the change of straight-line based layouts into circular layouts, such as radial and concentric-circle layouts in graph visualization [49]. Bachmaier [34] extended Sugiyama's drawing into radial drawings of hierarchical information. Keim et al. [35] used hierarchic radial layouts to visualize and analyze network activities of computer hosts. Vliegen et al. transformed the classical treemaps into a concentric-circle drawing to visualize business data [36] and Brandes et al. map social network structure to geometric centrality [37]. Giacomo et al. analyze the advantages and disadvantages in radial drawings of graphs theoretically [38].

Although the above techniques can increase the readability of visualization, however, they cannot visualize multi-dimensional datasets.

This paper proposes a new multi-dimensional visualization technique. This method can be used to identify the main features of network attacks, such as DDoS attack, by displaying their recognizable visual patterns. To reduce the visual clutter, axes are arranged as concentric circles rather than the traditional parallel lines. In our approach, edges are drawn as segments of curves rather than polylines in the parallel coordinate approach. Some heuristics are applied in our new method in order to improve the readability of views. We discuss the advantages as well as the limitations of our new method in the following sections.

## 3. *n*-dimensional graph model and layout

In this section, we will describe the calculation of the geometrical layout of $n$-dimensional data in the concentric-circle visualization. Before describing the layout, the data attributes and their corresponding graph model will be defined. As the visualization consists of a set of data values, a set of curves and a set of axes, we may use the graph model to create a geometrical abstraction of the data.

### 3.1. Transformation between dataset and graph

We are dealing with an $n$-dimension set of discrete data attributes

$$A^n = \{a^{(i)}\}, \quad i = 1,2,\ldots,m \tag{1}$$

and each set of data attributes $a^{(i)}$ is mapped to a set of values $d^{(i)}$ at $n$- dimensions

$$f_a : a^{(i)} \rightarrow d^{(i)} = (d_1^{(i)}, d_2^{(i)}, \ldots, d_n^{(i)}) \tag{2}$$

where

$$d^{(i)} \in R^n \tag{3}$$

We use graph $G=(V^{m\times n}, E^{m\times(n-1)})$ to model our $n$-dimension visualization, where $V^{m\times n}$ is representing $m$ sets of values in $n$ dimensions, and $E^{m\times(n-1)}$ is representing $m$ sets of $n-1$ edges that are linking with $m\times(n-1)$ pairs of data values.

The vertex set consists of $n$ subsets, namely

$$V^{m\times n} = \bigcup_{j=1}^{n} V_j, \quad V_j \in R^{m\times 1} \tag{4}$$

Therefore, each value can be mapped to each vertex

$$f_v : d_j^{(i)} \rightarrow v_j^{(i)} \in V_j \tag{5}$$

Each set of the values $d^{(i)}=(d_1^{(i)}, d_2^{(i)}, \ldots, d_n^{(i)})$ can map to a corresponding set of edges

$$d^{(i)} \xrightarrow{f} E^{(i)} = \left\{ (e_{1,2}^{(i)}, e_{2,3}^{(i)}, \ldots, e_{n-1,n}^{(i)}) \middle| v_j^{(i)} = f_v(d_j^{(i)}), \quad e_{j,j+1}^{(i)} = f_e(v_j^{(i)}, v_{j+1}^{(i)}) \right\} \tag{6}$$

We define $E_i$ as a "polyline" or "polycurve". In parallel coordinate drawings, we use a polyline $pl^{(i)}$ that consists of a set of straight lines to represent $d^{(i)}=(d_1^{(i)}, d_2^{(i)}, \ldots, d_n^{(i)})$, where $pl^{(i)}=(e_{1,2}^{(i)}, e_{2,3}^{(i)}, \ldots, e_{n-1,n}^{(i)})$, and $e_{j,j+1}^{(i)}=(v_j^{(i)}, v_{j+1}^{(i)})$. The map between the value of each dimension of each data item and each vertex is

$$f_p : d_j^{(i)} \rightarrow v_j^{(i)} \in V_j, \quad j = 1,2,\ldots,n-1.$$

In concentric-circle coordinate drawings, we use a polycurve $pc^{(i)}$ that consists of a set of connected small curves to represent $d^{(i)}=(d_1^{(i)}, d_2^{(i)}, \ldots, d_n^{(i)})$, where $pc^{(i)}=(e_{1,2}^{(i)}, e_{2,3}^{(i)}, \ldots, e_{n-1,n}^{(i)})$, and the calculation of $e_{j,j+1}^{(i)}$ is described in Section 3.3.2. The map between the value of each dimension of each data value and each vertex is

$$f_c : d_j^{(i)} \xrightarrow{v(\rho,\theta)} v_j^{(i)} \in V_j, \quad j = 1,2,\ldots,n-1.$$

After the data attribute sets are transformed into an abstract graph, we can now apply a particular geometrical layout method to draw the abstract graph for visualization. In the classical parallel coordinate methods each edge (data item) is drawn as a polyline across all parallel axes, while each vertex is placed on a particular parallel axes. However, in our new approach we locate a vertex $v_j^{(i)} \in V_j$ on the circle $\rho_j$ with radius $r_j$, which represents the $j$th dimension. The location of the vertex depends on $\rho=r_j$ and the relative angle $\theta$ in polar coordinates, i.e. $v(\theta,\rho)(0 \le \theta < 2\pi)$. Under these mappings, each data item corresponds to a series of spirals intersecting with the

dimension circles, just as the polyline across the axes set in the classical parallel coordinate method.

The edges set of the graph is partitioned into $n-1$ subsets

$$E = E_{1,2} \cup E_{2,3} \cup \ldots \cup E_{n-1,n} \tag{7}$$

where

$$E_{i,i+1} = \{e_{u,v} | \mu \in V_i, v \in V_{i+1}\} \tag{8}$$

Then we use

$$\gamma_{i,i+1} = \frac{|E_{i,i+1}|}{|V_i||V_{i+1}|} \tag{9}$$

to describe how edges are connected between two neighboring vertex subsets, which also reflects the data density between two dimensions.

### 3.2. The positioning of circular axes

Similar to the classical parallel coordinate visualization, the positioning of axes in concentric-circle visualization affects directly the readability of visualization and is very important for users to percept date items and associated values, especially the order of these axes that is the essential issue in the design of visualization. Therefore, it is very important to optimize the arrangement of dimension axes. Since it has been proved in [39] that it is a *NP*-hard problem in finding an optimal order of the dimension axes based on the similarities between dimensions. Therefore, we need to import some additional factors for optimizing the order of dimension axes.

Since the length of dimension axes appearing as a number of concentric circles differs with each other, it is expected that the dimension with higher density of data should be positioned as an outer circle to reduce the edge overlaps and crossings while the dimension with lower density of data should be positioned as an inner circle.

### 3.3. The drawing of curves

In the classical parallel coordinate visualization, the end vertices (values of data item) are placed on parallel coordinates as dots in advance, and then the straight lines are drawn to connect these end vertices in pairs axe by axe. However, in our approach, there are some other factors needed to be considered in the drawing of curves, such as the winding direction, the curve shape etc.

#### 3.3.1. The direction of winding

Obviously, for each connection between a pair of vertices, there are two different winding directions: clockwise or counter-clockwise. A given winding direction will produce different lengths and shapes of the curve. Without loss of generality, we make rules that all the curves start from the inner circle and point outward. In such cases, clockwise means that the angle increases along the curves, and counter-clockwise inversely, see Fig. 2 for example. It is clear that in comparison between these two winding directions, the curve drawn in Fig. 2(a) is much shorter than the one drawn in Fig. 2(b). Therefore, in this case we should use curve shown in Fig. 2(a) in the actual visualization.

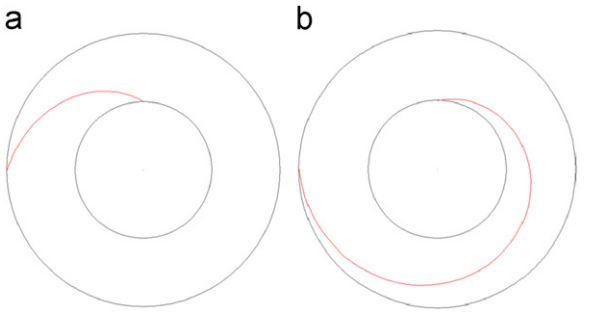


**Fig. 2.** Two winding directions: (a) Counter-Clockwise Winding and (b) Clockwise Winding.

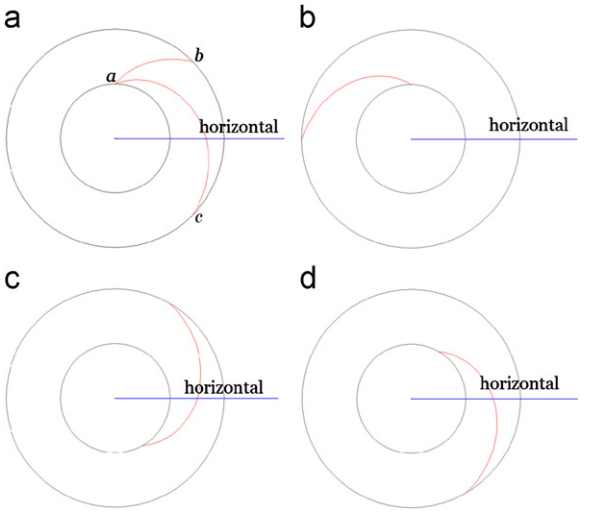


**Fig. 3.** Angle ambiguities: (a) Angle Ambiguity, (b) $\varphi = 0$, (c) $\varphi = -1$ and (d) $\varphi = 1$.

As explained in the above example, we can now define the winding direction as follows:

Suppose that the angle span of a given curve $|\theta_a - \theta_b|$ is less than $\pi/2$. Thus we have the winding direction $\delta: E \rightarrow \{-1,1\}$. We aim to shorten the length of the spiral curve as much as possible.

$$\delta(e) = \begin{cases} 1, & \text{if } gn(\Delta\theta)(|\Delta\theta| - \pi) \geq 0 \\ -1, & \text{if } gn(\Delta\theta)(|\Delta\theta| - \pi) < 0 \end{cases} \tag{10}$$

where sgn: $R \rightarrow \{1,-1\}$, $\Delta\theta = \theta_a - \theta_b$ $\rho_a < \rho_b$. It denotes "clockwise direction" when $\delta = 1$ and "counter-clockwise direction" when $\delta = -1$.

The periodicity of polar coordinates could produce some ambiguities when the curve crosses the horizontal line $\rho = 0$(see Fig. 3(a)). Both curves wind clockwise, but $\theta_a < \theta_b$ and $\theta_a > \theta_c$. Therefore, another variable called span $\phi: E \rightarrow \{-1,0,1\}$ is introduced to tag that character.

$$\varphi(e) = \begin{cases} 1, & \text{if } \delta(e) = 1 \text{ and } (\theta_a - \theta_b) > 0 \\ -1, & \text{if } \delta(e) = -1 \text{ and } (\theta_a - \theta_b) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

*3.3.2. Curve calculation*

Bezier curve is the common way to connect two points in the plane smoothly, and we use it to draw curve $e^{(i)}_{j,j+1}$. In polar coordinate, a curve connecting vertexes "a" and "b" can be calculated

$$\begin{cases} \theta(t)=\theta_a+(1-t)\theta_b \\ \rho(t)=\rho_a+(1-t)\rho_b \end{cases}, \quad (0<t<1) \tag{12}$$

where $\rho_a<\rho_b$. The point $p(\theta,\rho)$ on the curve generated by (12) should satisfy the conditions $\theta_a<\theta_p<\theta_b$ and $\rho_a<\rho_p<\rho_b$. After considering the issue of angle ambiguity expressed by (11), we can modify and rewrite the curve calculation expression (12) into the following new expression:

$$\begin{cases} \theta(t)=\theta_a+(1-t)(\theta_b+2\pi\cdot\varphi(e)) \\ \rho(t)=\rho_a+(1-t)\rho_b \end{cases}, \quad (0<t<1) \tag{13}$$

*3.3.3. Reduction of edge crossings*

In the beginning of this section, we have defined an $n$-dimensional graph model $G=(V,E)$ which is extracted from the multi-dimensional dataset $D^n=\{d^{(i)}\}$. In graph $G$, the distribution of vertices, which represent the values of data items, affects directly the readability of the views. For continuous data, the vertex can just fall on the circle increasingly or decreasingly. However, when disposing discrete datasets with weak monotonicity or the categorical data [33], the layout can be made more readable if breaking the monotonicity.

We aim to adopt graph drawing methods to reduce the crossings and overlaps of polylines. Therefore, we trade these polylines as edges in graph $G$. The edge crossing condition in graph drawing, which is slightly different from the polyline crossings in parallel coordinates, will be discussed first in this section. The new definition sets up the bridge to apply heuristics in horizontal edge crossing reduction to our concentric-circle fashion. On the other hand, we have proven that in the complete bipartite graph $K_{n,n}$(when $n$ is large enough), we can reduce up to 33% of edge crossings by using concentric-circle coordinator in comparison with the parallel coordinator. The detail of the proof is shown below:

**Theorem 1.** In a two-layer visualization of graph $K_{n,n}$, drawn in the concentric-circle coordinate, up to 33% of the total edge crossings can be reduced in comparison with the parallel coordinate layout when $n$ is large enough.

**Proof.** In the parallel coordinate layout, for a given edge $e_{u,v}$, each edge $e_{i,j}$ with the end vertices $i\prec u$ and $jmac_sc;v$ crosses $e_{u,v}$, and each edge $e_{i,j}$ with the end vertices $imac_sc;u$ and $j\prec v$ also crosses $e_{u,v}$ see Fig. 4. Thus the total number of edges crossing with edge $e_{u,v}$ is

$$pEdgeCr(e_{u,v})=\sum_{i=1}^{u} i \sum_{j=v+1}^{n} j+\sum_{j=1}^{v} j \sum_{j=u+1}^{n} i \tag{14}$$

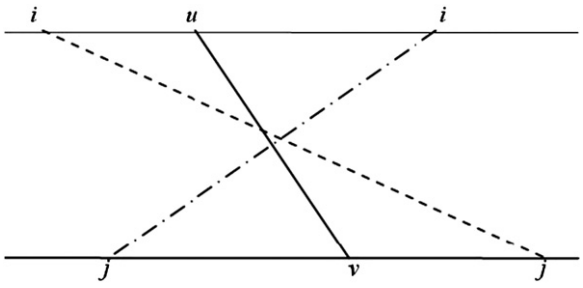


**Fig. 4.** An edge crossing in parallel coordinate layout.

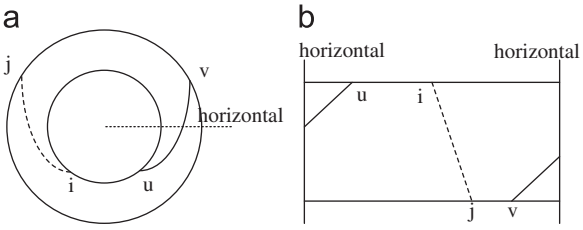


**Fig. 5.** (a) An edge crossing in concentric-circle layout, and (b) Unwinding the concentric circle to remove the edge crossing.

Therefore, the total number of edge crossings in the graph $K_{n,n}$ drawn in parallel coordinate approach is

$$pCr(K_{n,n})=\frac{1}{2}\sum_{u=1}^{n}\sum_{v=1}^{n} pEdgeCr(e_{u,v})=\frac{1}{4}n^4-\frac{1}{2}n^3+\frac{1}{4}n^2 \tag{15}$$

In Section 3.3 we defined some rules to avoid ambiguities while the angle span of curves is larger than $\pi/2$. If we unwind the circle along the *original line*, edges $e_{u,v}$ with $u-v>n/2$ will intersect *original line*, as showed in Fig. 5. Without losing generality, we assume that vertices are placed on axes or circles uniformly.

Suppose that $u\prec v$ and $v-u=s$, so the possibilities edge crossings with edge $e_{u,v}$ include four situations that are described below:

(a) If $i\prec u$ and $j\succ v$, then there will be $n/2-(u-i)$ edges from vertex $i$ crossing with $e_{u,v}$, see Fig. 6(a). Thus. the total number of edge crossings under this condition is

$$\sum_{v-(n/2)<i<u} [n/2-(u-i)] \tag{16}$$

(b) If $i\succ u$, $j\prec v$ and $i\prec j$, there will be $n/2$ edges from vertex $i$ crossing with $e_{u,v}$. See Fig. 6(b). Thus, the total number of edge crossings under this condition is

$$sn/2, \tag{17}$$

(c) If $i\succ u$, $j\prec v$, $i\succ j$ and $i\prec v$, there will be $v-i$ edges from vertex $i$ crossing with $e_{u,v}$, see Fig. 6(c). Thus, the total number of edge crossings on this condition is

$$\sum_{u<i<v} v-i \tag{18}$$

(d) If $i\succ u$, $j\prec v$, $i\succ j$ and $i\succ v$, then there will be $n/2-(i-v)$ edges from vertex $i$ crossing with $e_{u,v}$

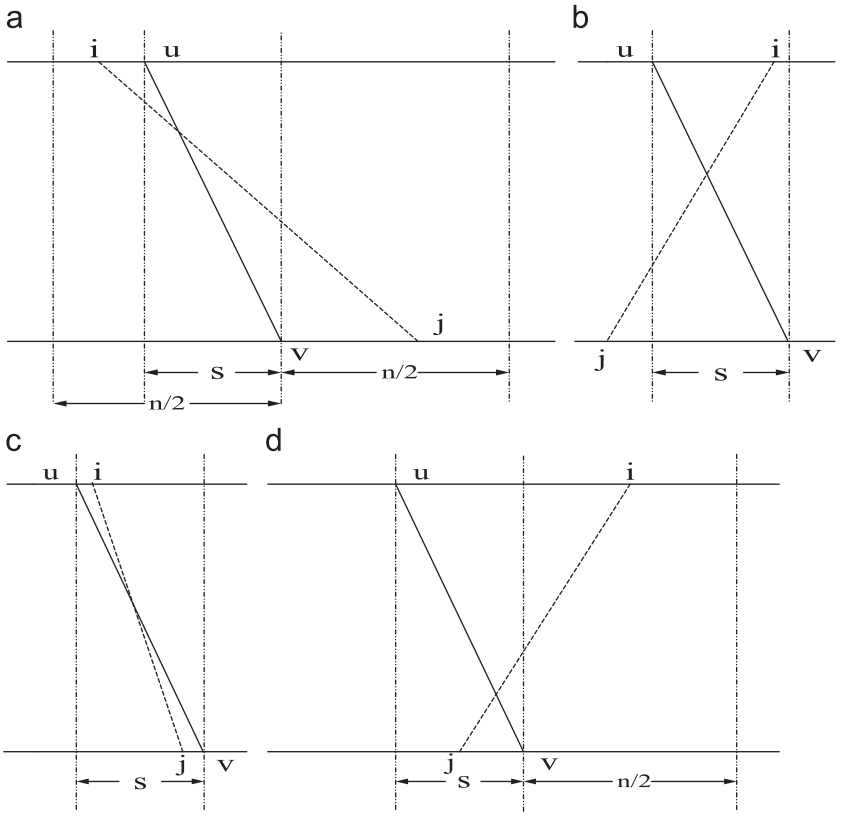


**Fig. 6.** Edge crossings with different conditions in concentric-circle layout.

(see Fig. 6(d)). Therefore, the total number of edge crossings on this condition is:

$$\sum_{v<i<v+(n/2)} [n/2-(i-v)] \tag{19}$$

From (16)–(19), we can calculate the total number of edge crossings for each edge $e_{u,v}$ is

$$cEdgeCr(e_{u,v}) = \sum_{v-(n/2)<i<u} \left[\frac{n}{2}-(u-i)\right] + s\cdot\frac{n}{2} + \sum_{u<i<v} (v-i) + \sum_{v<i<v+(n/2)} \left[\frac{n}{2}-(i-v)\right] \tag{20}$$

Considering the symmetry property in the concentric circular layouts that every edge has the same edge crossings number. So, the edges from a vertex $u$ in the first layer should produce the following number of edge crossings

$$cVertexCr(u) = \sum_{s=0}^{(n/2)} cEdgeCr(e_{u,v}) + \sum_{s=1}^{(n/2)} cEdgeCr(e_{u,v}) \tag{21}$$

Thus the total number of edge crossings in graph $K_{n,n}$ under the concentric-circle scheme is

$$cCr(K_{n,n}) = \frac{1}{2} n\, cVertexCr(u) = \frac{1}{6}n^4 + \frac{3}{8}n^3 + \frac{1}{12}n^2 \tag{22}$$

Combining (15) and (22), we can get the formula (23) below. It indicates that when $n$ becomes large enough, under the concentric-circle coordinate scheme we can reduce up to 33% of the edge crossings in the drawing of a complete bipartite graph $K_{n,n}$, in comparison with the parallel coordinate scheme

$$\lim_{n\to\infty} \frac{cCr(K_{n,n})}{pCr(K_{n,n})} = \lim_{n\to\infty} \frac{(1/6)n^4+(3/8)n^3+(1/12)n^2}{(1/4)n^4-(1/2)n^3+(1/4)n^2} = \frac{2}{3} \tag{23}$$

*Definition of the edge crossing in concentric-circle coordinate*: In parallel coordinates, an edge crossing occurs between two polylines depends purely on the positions of their end vertices. In concentric-circle coordinates, however, the edge crossings could be affected by some other factors, such as the direction of curve winding and the crossing with the horizontal line as discussed in Section 3.3.

If span $\varphi(e_1)\neq 0$ or $\varphi(e_2)\neq 0$, it implies that there is a gap of $2\pi$ or $-2\pi$ between the reality and the literal meanings of end vertices angles (see Fig. 7). Relative to the other three vertices, the angle of $b_2$ should be in $(2\pi,4\pi)$, so $\theta(b_2)'=\theta(b_2)+2\pi$. It can be decided whether two curves cross only by angles of their end vertices after such transformation.

$$Cross(e_1,e_2) = \mathrm{sgn}(\theta(a_1)\prime-\theta(a_2)\prime)\wedge \mathrm{sgn}(\theta(b_1)\prime-\theta(b_2)\prime) \tag{24}$$

where sgn: $R\to\{1,-1\}$. It denotes there exists a crossing when $Cross(e_1,e_2)=1$.

*View Adjustment*: After positioning the axes and drawing of polycurves, the improvement of readability

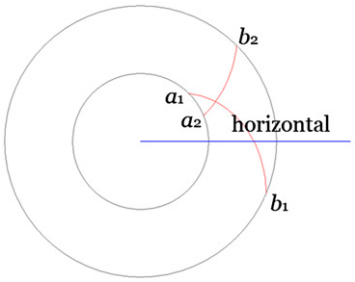


**Fig. 7.** The definition of edge crossings in concentric-circle coordinate.

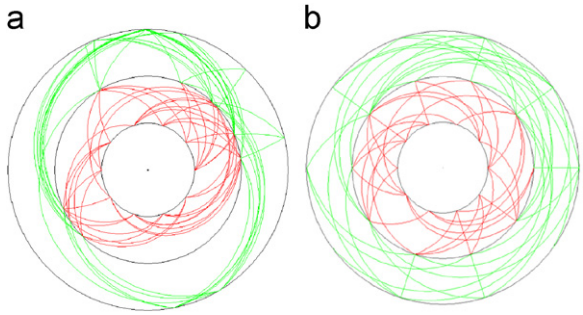


**Fig. 8.** A comparison between barycenter and greedy switch heuristics: (a) Barycenter Heuristic and (b) Greedy Switch Heuristic.

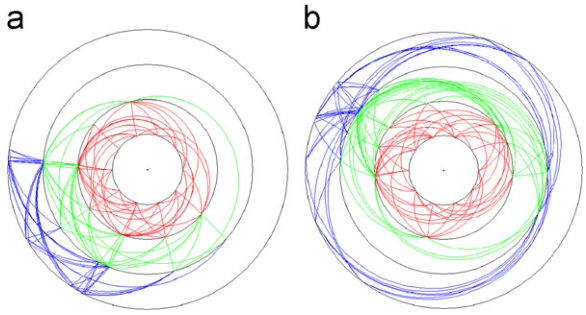


**Fig. 9.** An example of the distributions in (a) Median Heuristic and (b) Barycenter Heuristic.

of views becomes crucial. This can be done by optimizing the distribution of vertexes and the reduction of edge crossings. We attempted to minimize the edge crossings circle-by-circle. We initialize the vertexes ordering in the inner most circles first according to the monotonicity, and then gradually sweep to outer circles.

It has been proved that it is a NP-hard problem to minimize the edge crossings between two axes by changing the ordering of axes in parallel coordinate even if the ordering of vertices in one axe of the graph is fixed [40], and Bachmaier pointed out that the same problem in radial layouts is also NP-hard [34]. As a consequence, it has been advised that to find some heuristics to reach an efficient solution. Many methods have been proposed to resolve the problem in the parallel layout [40]. However, since the edge crossing definition has been expressed, we can derive from the heuristics in the parallel layout to solve the same problem in our concentric-circle layout apparently.

*Greedy Switch Heuristic*: In parallel layout, the greedy switch heuristic works in a similar way as bubble-sort [40]. Suppose that $u$ and $v$ are two consecutive vertices on the same layer, $u$ is prior to $v\mu \prec v$. Then, swapping their positions changes the total number of crossings by $C_{v\mu} - C_{\mu v}$, where $C_{\mu v}$ denotes the number of the crossings between edges from $u$ and edges from $v$. The algorithm scans all consecutive pairs and swaps them if this reduces the number of crossings. This process is repeated until no further swap is required.

Section 3.1 shows that the use of greedy heuristic swap can reduce crossings in the concentric-circle layout if two curves crosses are dependent on their vertices' angles. Obviously, the algorithm has the same running time as the horizontal version $O(|\rho|^2)$.

*Sifting Heuristic*: Sifting was originally introduced as a heuristic for vertex minimization of ordered binary decision diagrams [41], and it was later adapted for the (parallel) crossing minimization problem [42]. The algorithm determines optimal position of each vertex that positions of the other vertices are fixed.

As same as greedy switch heuristic, our algorithm can also be implemented in the concentric-circle layout without much change from Section 3.1. Since every vertex has to be set on every position, the time complexity is as $O(|\rho|^2)$, which remains the same as parallel layout.

*Barycenter Heuristic*: The barycenter heuristic in the parallel coordinate crossing reduction, which is also called averaging, is based on the intuition. There are fewer crossings when each vertex is close to its adjacent vertex. Because the method is simple and produces good result, it is widely applied to different applications.

In concentric-circle layout, we can locate a vertex $u$ on the outer circle whose angle equals to the average one of its adjacency on the inner circle.

$$\theta(u) = \frac{\sum_{e_{(u,v)} \in E_{(\rho_1,\rho_2)}} \theta(v)}{\deg(u)} + \Delta \tag{25}$$

where $\deg(u) = |\{(u,v)|(u,v) \in E_{(\rho_1,\rho_2)}\}|$ and $\Delta$ is just a very small random number to prevent overlapping when two nodes have the same average values. The layout ensures that vertices with same or similar adjacent vertices will be placed near each other. The pattern can uncover some relationship in the initial dataset (see Fig. 8).

*Median Heuristic*: Similar to the barycenter heuristic, median heuristic places vertices close to their adjacent vertices. However, median heuristic replaces the median with the average. Median heuristic generally distributes vertices in a denser way in comparison with barycenter heuristic, because it is more likely to have the same median than average for two data groups (see Fig. 9).

The running time for computing the average or the median is proportional to the degree of $u$. Therefore both barycenter heuristic and median heuristic can be calculated in linear time.

## 4. Comparison between parallel coordinates and concentric-circle coordinates

To measure the performance of concentric-circle drawings, we implemented our new technique by using Java programming language (Java Development Kit 1.5), running on an ordinary computer with 1.84 Ghz and 768 Mbs RAM memory. In order to compare the performance between these two methods in terms of the number of edge crossings and angle crossings, we use the heuristics to analyze both the parallel coordinate and concentric-circle coordinate drawings. We randomly select 160 different data groups in our experiment, and each group is a combination of the following parameters: $\gamma_{i,i+1}$={0.2,0.4,0.6.0.8} and $|E_{i,i+1}|$={10,20,30,40}, where $i$=1, 2, 3 and 4.

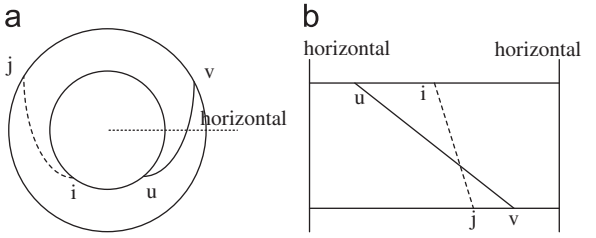


**Fig. 10.** An example shows that by using concentric-circle layout, some edges can be shorten that could reduce the possibility of causing edge crossings: (a) Concentric-circle Layout and (b) Parallel Layout.

### 4.1. The number of edge crossings

The experiments shown that with the same graph, the concentric-circle layout can reduce a certain number of the edge crossings in comparison with the parallel layout. With less dimensions or low density of data, the number of edge crossings can be reduced by around 25 percent. While with large number of dimensions or data items, it

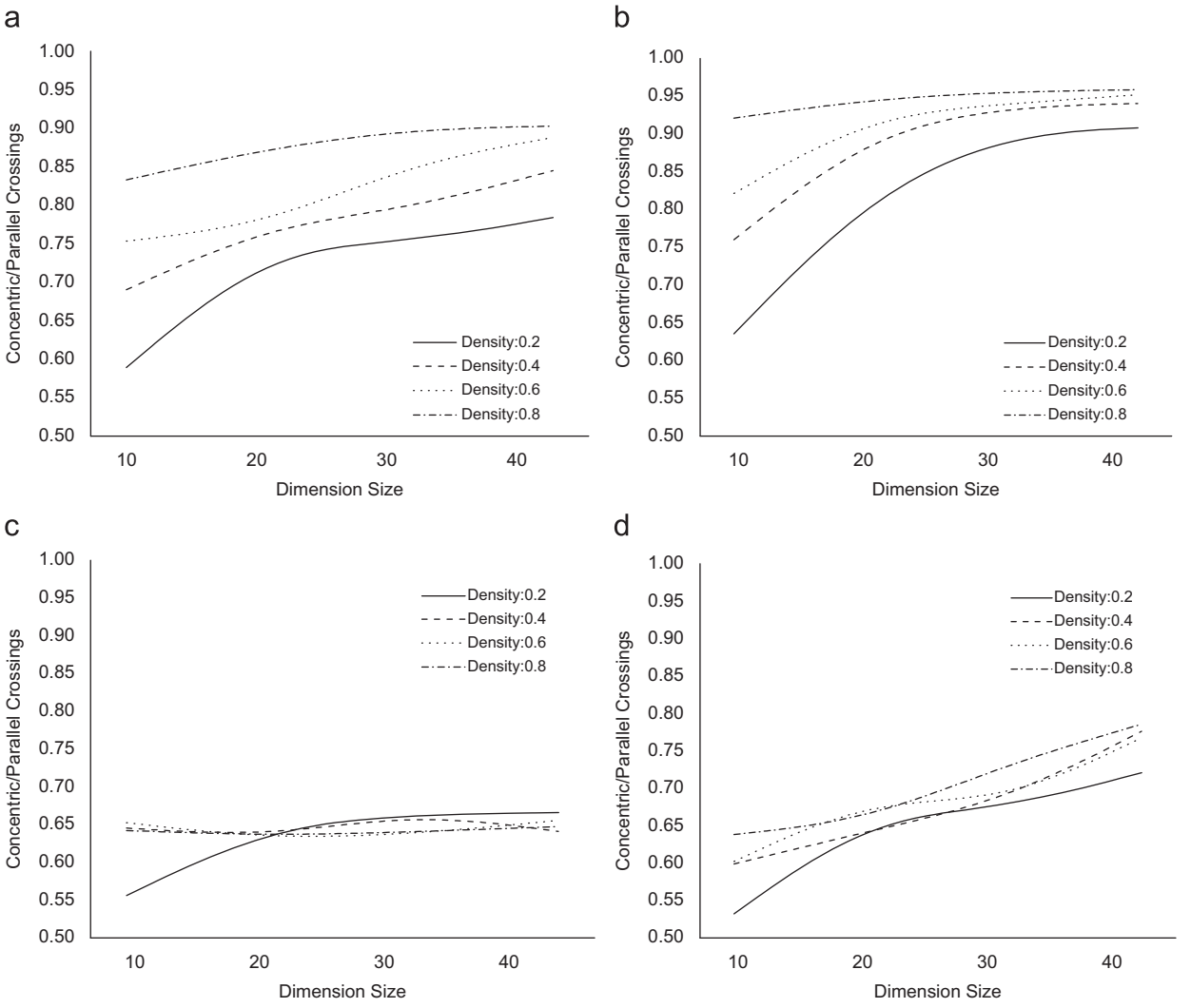


**Fig. 11.** The Comparison of the number of edge crossings with different heuristics: (a) Barycenter Heuristic, (b) Median Heuristic, (c) Greedy Switch Heuristic and (d) Sifting Heuristic.

can still reduce more than 15 percent of edge crossings (see Fig. 11). Based on the discussion in Section 3.3, the concentric-circle layouts can shorten the length of some edges, which could avoid some edge crossings (see Fig. 10). The experiment also shows that with the complete bipartite graph $K_{n,n}$, concentric-circle layouts could reduce up to 33% of edge crossings when $n$ is large enough.

Although by using concentric-circle layout we can reduce some edge crossings, different heuristics we use could produce different performances. When data density increases, the performance of barycenter heuristic (BH) and median heuristic (MH) is dropped down dramatically (see Fig. 12(a)). However, with the increase of data complexity, the performance of greedy switch heuristic (GS) and sifting heuristic (SH) remain stable (see Fig. 12(a)). On the other hand, the influence of the change in the number of dimensions is the same for all heuristics (see Fig. 12(b)). Therefore, we recommend that the barycenter heuristic and the median heuristic should be the first choice when visualizing relatively small dataset. If no real time requirement, the greedy switch heuristic and the sifting heuristic may provide more readable layouts in concentric-circle layouts.

### 4.2. The angle of edge crossings

As discussed in [43,44], when edges across orthogonally, there are less confused than crossing at an acute angles. In other words, an edge crossing with a large crossing angle makes much easier to be identified and more readable than those with a smaller crossing angles. Therefore, we measure the crossing angles in both parallel layouts (PL) and concentric-circle layouts (CL) in each heuristic discussed above: sifting heuristic (SH), greedy switch heuristic (GS), barycenter heuristic (BH), median heuristic (MH).

In Fig. 13, it clearly shows that the average edge crossing angles in concentric-circle layouts for all heuristics, except the median heuristic layout. We can see that there is no much improvement in comparison with the traditional parallel coordinate scheme. However, from Fig. 14, we can see that the concentric-circle layout is the winner in reducing edge crossing angles, if we only consider the first two dimensions of the data.

It is notified that the width of each dimension (the distance between two neighboring axes) in parallel layout is larger than the width in concentric-circle layout. From Fig. 14, we can see that the main advantages

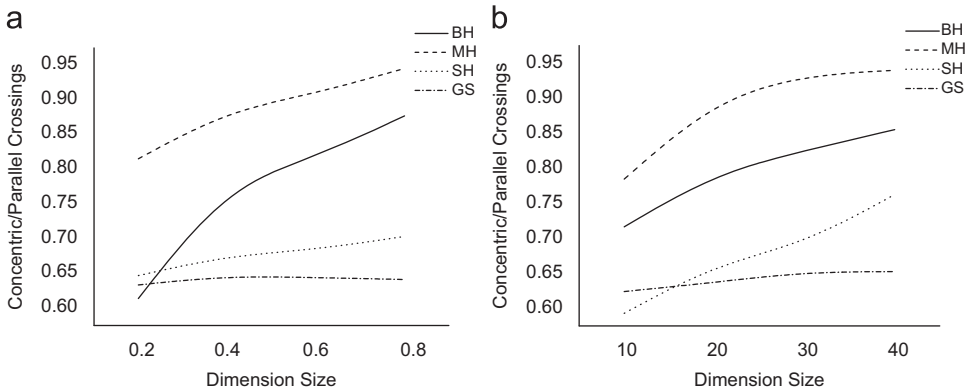


**Fig. 12.** The comparison of performance of all heuristics in terms of the number of dimensions and the size of data: (a) Adjustment/Density Compare, (b) Adjustment/Dimension Compare.

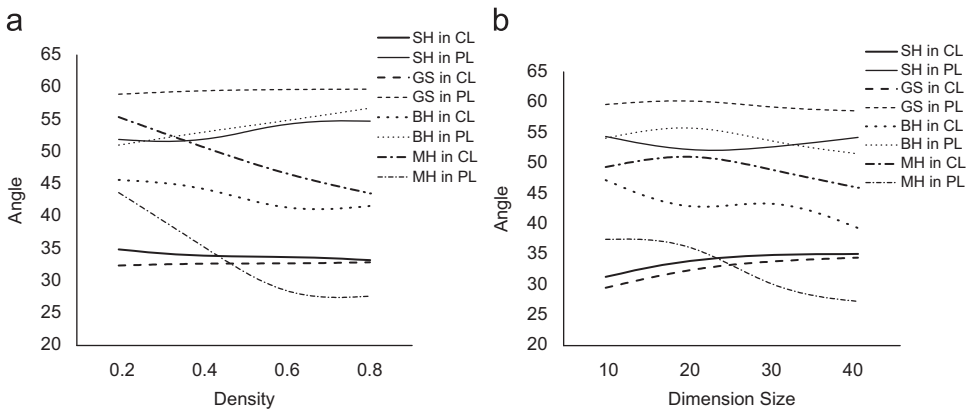


**Fig. 13.** The comparison of edge crossing angles in all dimensions: (a) With different Density and (b) With different Dimension size.

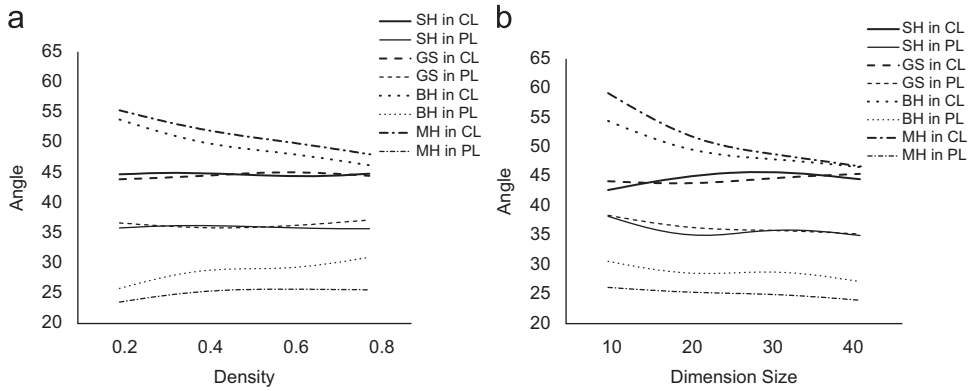


**Fig. 14.** The comparison of edge crossing angles between 1st and 2nd Dimensions: (a) With different Density, (b) With different Dimension size.

of concentric-circle layout are uniformed distribution of vertices and better edge crossing angles in inner circles whose degrees are enlarged when transformed from the parallel coordinate layout. These advantages improve the readability of the visualization even through the data items are located in the center of the circles. This feature could late help users in arranging the ordering of dimension axes; that is we could arrange sensitive and important data, those the user wants to view them with a better display quality, in the center of the circles.

## 5. Case studies

The study of network intrusion (attack) detection has been popular for the last decade. Many previous works have been done and most of them are focusing on detecting network attacks by analyzing large volume alerts. Some tools and visualization methods have also been developed to support this research.

The early work in networking visualization was conducted by Cheswick et al. [8]. It uses the force-directed graph visualization method to map and display the internet structures (graphs) showing routing and reachability information. This mapping technology late is used for network security visualization. Late, Makanju et al. [20] proposed an event (log) visualization tool, named LogView, for displaying clusters of event (log) data based on clustering structures produced by SLCT (Simple Log file Clustering Tool). This could be used in various log analysis tasks including profiling, selection, filtering and searching on event log datasets.

In 2005, Lee et al. [17] proposed a Visual Firewall to assist the configuration of firewalls that are monitoring the networks by providing four simultaneous views. These views display varying levels of detail, including time-scales and firewall reactions to individual packets.

In 2006, Simsek [19] proposed a L-BIDS technique (lattice-based intrusion detection system) to track the propagation of DDoS attacks, in which the structure of nodes are colored to give a concise intrusion signature. Certain features of distributive lattices were concluded.

In 2007, Samak et al. [7] proposed another methodology for representing network traffic using SFCs. Images generated using those curves preserve traffic properties while withstand aggressive compression. Traffic features and anomaly behavior are effectively detected using proposed mapping. They applied this technique to identify DDoS and Code Red attacks.

Pearlman [16] proposed a visualization method for network security by approaching the problem from a service-oriented perspective. This research provides a real time system for monitoring service activities, and enabling network administrators with the early stage detection of attacks, which include Denial of Service (DoS) attacks.

Although the above visualization techniques can assist analysis activities in the early stage of network intrusion detection. Most of them are based on network transaction and data volume analysis and they produce only an alarming service and in most cases the accuracy rate of the alarming of network attacks is very low. Thus, the actual identification and classification of a variety of attacks are still heavily relied on human brains. Furthermore, up to now there are no specific techniques and tools developed for detecting DDoS attacks. The above approaches focused mainly on the visualization of suspicious network activities and they do not help in the analysis of characteristics of network events.

Analysis of network scan is an alternative approach to detect network attacks. Many research works have been done in finding ways to deal with large volume of alerts produced by various visualization tools for detecting network scans. ScanVis [21] used the visualization and statistical techniques to analyze the patterns found in a large volume of periodical network scans. In 2004, Conti et al. [22] used a parallel coordinates technique to display scan details and characterized attacks. PortVis [23] proposed a three-semantic-level of visualization framework: timeline, hour (main), and port visualizations, for detecting different types of port scans.

Although the above port scan detecting methods are useful for the analysis of abnormal patterns, they are still based on single aspect (port scan) analysis. Therefore the accuracy of attack detection is still low. Our new proposed method uses a multiple-aspects analysis, which combines

the visual structure analysis and event (attack) feature based analysis.

Network scan is a common way for analyzing network intrusions. The purpose of scanning a network is to determine what was happened on the network. However, when the size of networking systems are grown rapidly and their structures are becoming more and more complex, the demanding of development of network security mechanism has exceeded the capacity of current network security tools.

Unfortunately, up to now, there is no perfect method to secure network communication completely. The currently techniques and tools of network security still heavily rely on human detection.

In this section, we evaluate our concentric-circle visualization by applying it into the network intrusion detection. Our experimental data are collected from a local network at The Image and Graphics Institute, Tianjin University, China. A visual network scan detection system called CCScanViewer is development that is based on our new visualization method. The system can visualize network traffic activities extracted from network flows and their patterns. The experiments shown that the system with our new visualization could easily detect the ports scans in the local area networks and these patterns are more identical and understandable than those generated in the parallel coordinate visualization.

### 5.1. Extract features of vertical/horizontal scans

We aim to detect as many attacks as possible. At the first step, we focus on detecting some popular network intrusions, such as port scans. Based on IP address and port number, there are three common port scan methods: (1) horizontal scan, (2) vertical scan and (3) block scan [45]. In our experiments, we use ScanPort 1.2 to test our new method. There attributes: source IP host address, destination port number and destination IP address are visualized from the inner circular axe to the outer circular axe in the concentric-circle coordinate visualization.

Fig. 15(1) displays the normal pattern of periodical networking communication between hosts 211.81.163.202 and 211.81.163.184 in our local network. In comparison with the normal pattern, patterns of vertical scan and horizontal scan have their own distinguished characteristics (see Fig. 15(2) and 15(3)).

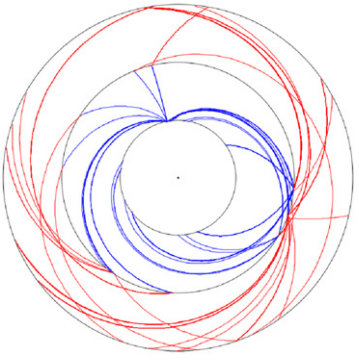

**Fig. 15(1).** A normal networking pattern displayed in CCScanViewer.

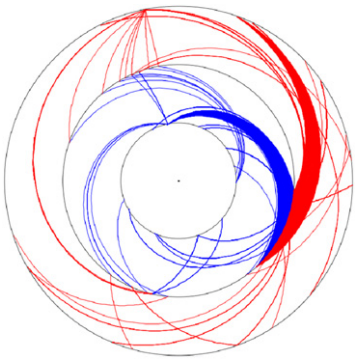

**Fig. 15(2).** Pattern of a vertical scan displayed in CCScanViewer.

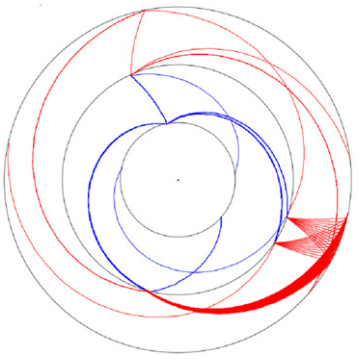

**Fig. 15(3).** Pattern of a horizontal scan.

As a common knowledge, a single host seldom connects with continuous ports of another host. From Fig. 15(2), we can see that there are many connections between 211.81.163.202 and 211.81.163.184 and all ports from 21 to 200 of the host 211.81.163.184 are scanned by the host 211.81.163.202. Therefore, all curves from the same point of the inner circle (see blue curves) converge finally to the same point in the outer circle (see red curves). This is the distinguishable characteristic of a vertical scanning. By recognition of this feature we can easily detect a vertical scan attack from the host 211.81.163.202.

Similarly, from Fig. 15(3), we can see that there are many connections between host 21.81.63.184 and hosts those partially from 192.168.2.2 to 192.168.2.254. From the middle circle, we can find that only few ports are involved in network communication. This is considered as the distinguishable feature of a horizontal scan.

From the above experiment, we can see that the shifting heuristic can distribute numerical points uniformly. Consequently, the curves can also cover circles evenly when the source IP and the destination IP addresses are displayed in two circular axes. Fig. 15(4) displays the logic characteristics of scanning.

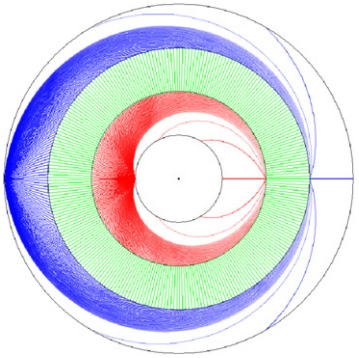

**Fig. 15(4).** Pattern of the logic characteristics of scanning.

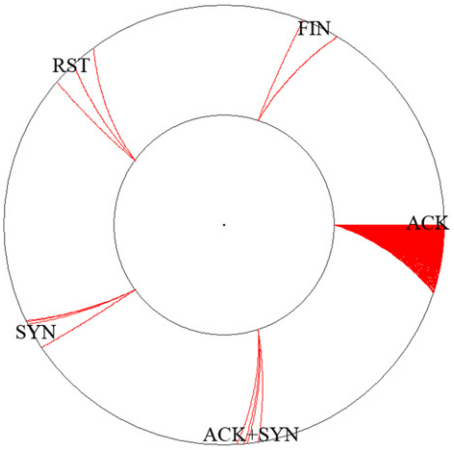


**Fig. 16(1).** The Pattern of normal TCP flags of network communication.

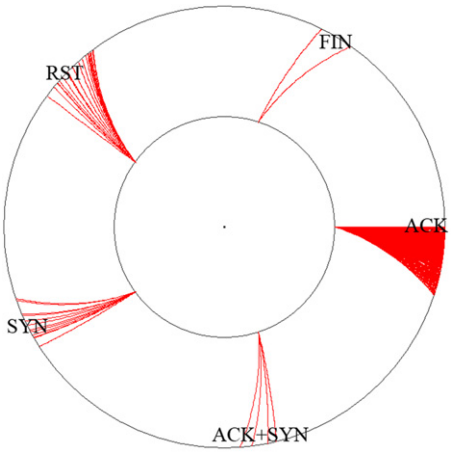


**Fig. 16(2).** The Pattern of a RST–SYN scan.

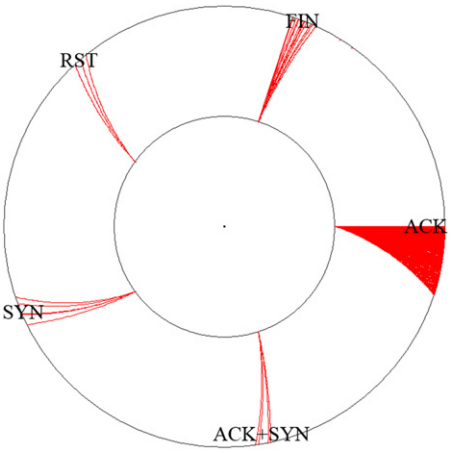


**Fig. 16(3).** The pattern of a FIN scan.

### 5.2. Distinguish the types of network scans

The aim of scanning a host or a cluster is to obtain the status of a network. There are several ways to achieve this. For example, the abnormal flag of the TCP data packet could cause an abnormal network connection. In order to establish a normal connection of two hosts, the majority of the flags in the TCP three-way handshakes should be acknowledged (ACK) to ensure the reliable and synchronized communications between two hosts. This is shown in Fig. 16(1). When a RST–SYN (Reset–Synchronize) attack occurs, the flags of SYN and RST increase accordingly. SYN data segments are sent to the destination hosts in the process of scans. If the response packages are with flag RST on, the ports are closed. On the other hand, if the response packages are with flags SYN and ACK on, the target ports are in the state of monitoring. Then the host sends RST flag to destination host, and the connection has not yet been established. Therefore, this pattern may be confused with the three-way handshakes. Fig. 16(2) shows the RST–SYN scan. Fig. 16(3) displays the statistics of TCP flags in the TCP packets. Obviously, the flags of FIN have a marked increasing in Fig. 16(3). However, the number in normal state should be rare. Consequently, we can easily see that a hidden FIN scan occurs. Our experiments showed that the new approach can also detect the types of network scan effectively.

### 5.3. Extract features of DDoS attacks in parallel and concentric-circle coordinate visualizations

In this section, we describe examples using concentric-circle visualization and parallel coordinate visualization for displaying and analyzing the patterns of DDoS attack. The characteristic of DDoS attacks is that within a short time period, the victim (host) may receive a large amount of requests and packets from a large number of strange IP addresses. Such as Smurf attacks, from which the victim's console is communicated with large number of IP addresses in a short period and most of IP addresses have never appeared previously. In our case study, we used Smurf attacks to demonstrate the advantage of our new

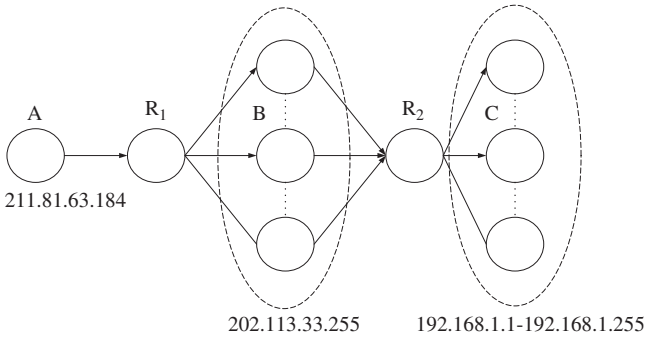


**Fig. 17(1).** The architecture of a DDoS attack.

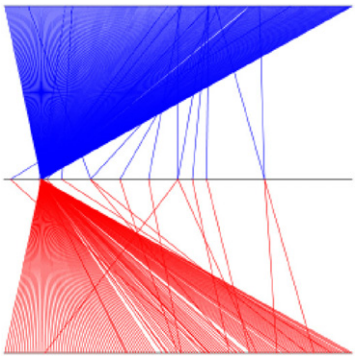

**Fig. 17(2).** The Pattern of a DDoS attack displayed in parallel coordinates.

method. In Smurf attack, the perpetrators send large numbers of IP packets from the source 211.81.63.184 to the victim 202.113.33.255, with a range of faked addresses from 192.168.1.1 to 192.168.1.255. Consequently, network's bandwidth will be occupied quickly, and the legitimate data packets to destination 202.113.33.255 will be blocked. The architecture of this attack is shown in Fig. 17(1).

In CCScanViewer, the above DDoS attacks can be easily detected. Figs. 17(2) and 17(3) give a comparison between two different visualizations. We can see that in concentric-circle layout edge crossings can be significantly reduced and display space can also be optimized.

As discussed in Section 3, concentric-circle layout is better in catching distinguishable attack features from large volumes of network data, in comparison with parallel coordinate layouts. Therefore, in the case of visualizing one-to-many or many-to-many data communications, our approach can be more efficient and accurate in producing recognizable visual patterns.

To periodically produce data transformation patterns (such as the one shown in Fig. 17(3)) for analysis, in our experiments we updated the graph every 10 min to generate a new graph. We then compare these graphs to enable us identifying the abnormal patterns. However, the selection of time interval is a challenge. This is because that the volumes of data transformation received from different time slots are different, e.g. time slots in peak/off peak time periods. We attempt to receive the most appropriate amount of data in a certain time period to ensure the best quality of image for feature detection. So as one of the future works, we will investigate algorithms for selecting optimized time interval.

## 6. Conclusion and future work

This paper proposed a new visualization method: concentric-circle coordinate for visualizing multi-dimensional datasets. We have successfully applied this method for network intrusion detection through the extraction and visualization of recognizable features of a variety of network attacks.

In Section 4, we claimed that our experiments shown that the new approach could reduce 15% of line crossings in comparison with the traditional parallel coordinate scheme. These experiments were conducted with the similar situation, as conducted in parallel coordinate approach, such as the data scalability and the heuristic methods. We have selected 160 different datasets randomly in our experiments. Even though these experiments are empirical, the datasets are chosen randomly. Therefore, we believe that our experiments are data-independent. In Section 3, we have given the theoretical proof that in comparison with the parallel coordinate, the concentric-circle coordinate can reduce up to 33% of the line crossings if the raw data could be transferred into complete bipartite graph.

Some additional heuristics are applied to our method in order to improve the readability of the views. The experiments showed that the new approach is effective in finding the patterns of network scans, port scans, the hidden scans, DDoS attacks and others.

In the future, we intend to investigate appropriate interaction mechanisms to solve the unequal dimension distribution problem that allowing users to view the detail of their interested part of the data through the interactive swapping of circular axes.

Furthermore, we will continuously evaluate our new method by conducting more case studies and apply this method into other application fields.

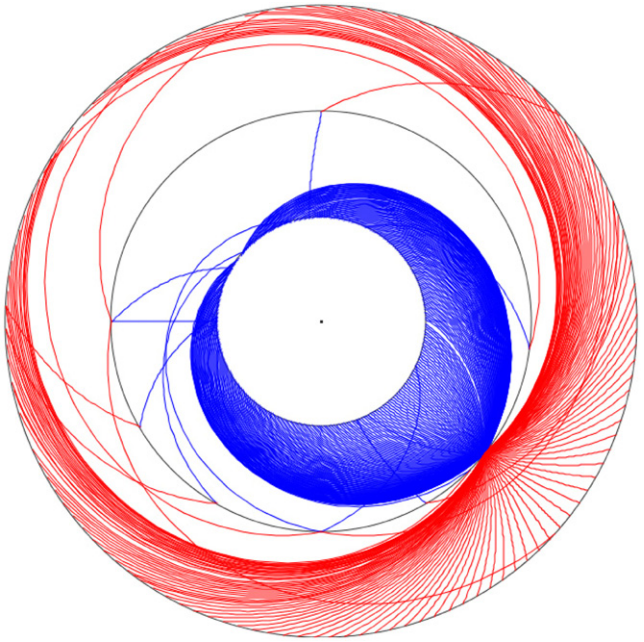

**Fig. 17(3).** The Pattern of a DDoS attack in concentric-circle visualization.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No.60673196 and the Natural Science Foundation of Tianjin, P.R. China, under Grant No. 07F2030.

## References

[1] X. Yin, W. Yurcik, M. Treaster. VisFlowConnect: NetFlow visualizations of link relationships for security situational awareness, in: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security. Washington, DC, USA, ACM Press, pp. 26–34.
[2] R.F. Erbacher. Visual traffic monitoring and evaluation, in: Proceedings of the Conference on Internet Performance and Control of Network Systems II, 2001, pp. 153–160.
[3] L. Girardin and D. Brodbeck. A visual approach for monitoring logs, in: Proceedings of the 12th Usenix System Administration conference, 1998, pp. 299–308.
[4] C. Muelder, K.L. Ma, T. Bartoletti, A visualization methodology for characterization of network scans, Visualization for Computer Security (2005) 29–38.
[5] T. Samak, S. Ghanem, and M.A. Ismail. On the Efficiency of using Space-Filling Curves in Network Traffic Representation, in: Proceedings of the INFOCOM Workshops 2008, IEEE, pp. 1–6.
[6] J. Wang, D.J. Miller, G. Kesidis, Efficient mining of the multidimensional traffic cluster hierarchy for digesting, visualization, and anomaly identification, IEEE JSAC on High-Speed Network Security 24 (10) (2006) 1929–1941.
[7] T. Samak, A. El-Atawy, E. Al-Shaer, M. Ismail, A novel visualization approach for efficient network-wide traffic monitoring, End-to-End Monitoring Techniques and Services (2007) 1–7.
[8] B. Cheswick, H. Burch, and S. Branigan. Mapping and Visualizing the Internet, in: Proceedings of the 2000 USENIX Annual Technical Conference, 2000, pp. 1–12.
[9] A. Inselberg, B. Dimsdale, Parallel coordinates: a tool for visualizing multi-dimensional geometry, in: Proceedings of Visualization '90, 1990, pp. 361–370.
[10] Y.-H. Fua, M.O. Ward, and E.A. Rundensteiner, Hierarchical parallel coordinates for exploration of large datasets, In: Proceedings of the IEEE Visualization, 1999, pp. 43–50.
[11] Jimmy Johansson, Patric Ljung, Mikael Jern, Matthew Cooper, Revealing Structure within clustered parallel coordinates displays, in: Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS'05),2005, pp. 125–132.
[12] Y. Xiaoru, G. Peihong, X. He, Z. Hong, Q. Huamin, Scattering points in parallel coordinates, IEEE Transactions on Visualization and Computer Graphics (InfoVis'09) 15 (3) (2009) 1001–1008.
[13] Z. Hong, C. Weiwei, Q. Huamin, W. Yingcai, Y. Xiaoru, Z. Wei, Splatting the lines in parallel coordinates, Computer Graphics Forum (EuroVis'09) 28 (3) (2009) 759–766.
[14] Z. Hong, Y. Xiaoru, Q. Huamin, C. Weiwei, C. Baoquan, Visual clustering in parallel coordinates, Computer Graphics Forum (EuroVis'08) 27 (3) (2008) 1047–1054.
[15] M. Novotny, H. Hauser, Outlier-preserving focus+context visualization in parallel coordinates, IEEE Transaction on Visualization and Computer Graphics 12 (5) (2006) 893–900.
[16] J. Pearlman, Visualizing network security events using compound glyphs from a service-oriented perspective, Visualization for Computer Security. VizSEC 2007: Proceedings of the Workshop on Visualization for Computer Security (2007) 131–146.
[17] C.P. Lee, J. Trost, N. Gibbs, R. Beyah, J.A. Copeland, Visual firewall: real-time network security monitor, IEEE Workshop on Visualization for Computer Security 2005 (VizSEC 05) (2005) 129–136.
[18] C. Papadopoulos, C.K. Alexander Sawchuk, Xinming He, CyberSeer: 3D audio–visual immersion for network security and management, in: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security. Washington, DC, USA: ACM Press, 2004, pp. 90–98.
[19] S. Simsek, Work in progress—tracking correlated attacks in enterprise intranets through lattices, Securecomm and Workshops (2006).
[20] A. Makanju, S. Brooks, A.N. Zincir-Heywood, E.E. Milios, LogView: visualizing event log clusters, Privacy, Security and Trust (2008) 99–108.
[21] C. Muelder, K.L. Ma, T. Bartoletti, A visualization methodology for characterization of network scans, Visualization for Computer Security, IEEE Workshops (2005) 4.
[22] G. Conti, Abdullah, K. Passive visual fingerprinting of network attack tools, VizSEC/DMSEC '04, in: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, 2004, pp. 45–54.

[23] J. McPherson, K.L. Ma, P. Krystosk, T. Bartoletti, M. Christensen Portvis: a tool for port-based detection of security events, In: Proceedings of the ACM VizSEC 2004 Workshop, 2004, pp. 73–81.
[24] M.F. Mokbel, W.G. Aref, and I. Kamel. Performance of multi-dimensional space-filling curves, in: Proceedings of the tenth ACM International Symposium On Advances In Geographic Information Systems, November 08-09, McLean, Virginia, USA,2002 pp. 149–154.
[25] W. Peng, M.O. Ward, and E.A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering, in: Proceedings of the IEEE Symposium on Information Visualization 2004 (INFOVIS'04), 2004 pp. 89–96.
[26] Y. Fua, M.O. Ward, E.A. Rundensteiner. Navigating hierarchies with structure-based brushe,. in: Proceedings of the 1999 IEEE Symposium on Information Visualization (INFOVIS'99), 1999, pp. 58–64.
[27] G.P. Ellis, A. Dix. Density control through random sampling: an architectural perspective, in: Proceedings of International Conference on Information Visualization (IV'02), 2002 pp. 82–90.
[28] G. Ellis, A. Dix, Enabling automatic clutter reduction in parallel coordinates plots, IEEE Transaction on Visualization and Computer Graphics 12 (5) (2006) 717–723.
[29] J. Johansson, M. Cooper, M. Jern. 3-Dimensional Display for Clustered Multi-Relational Parallel Coordinates, in: Proceedings of the Ninth International Conference on Information Visualisation (IV'05), 2005 pp. 188–193.
[30] R. Wegenkittl, H. Löffelmann, E. Gröller. Visualizing the behavior of higher dimensional dynamical system,. in: Proceeding of the IEEE Visualization 1997 (Phoenix, USA, 1997), ACM Press: New York, 1997 pp. 119–125.
[31] C. Forsell, J. Johansson. Task-based evaluation of multi-relational 3D and standard 2D parallel coordinates, in: Proceedings of the IS&T/SPIE's International Symposium on Electronic Imaging, Conference on Visualization and Data Analysis 2007 (San Jose, USA, 2007), SPIE: Bellingham and IS&T:Springfield; 64950C, 2007, pp. 1–12.
[32] A.O. Artero, M.C.F. de Oliveria, H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations, in: Proceedings of the 2004 IEEE Symposium on Information Visualization (INFOVIS'04), 2004, pp. 81–88.
[33] R. Kosara, F. Bendix, H. Hauser, Parallel sets: interactive exploration and visual analysis of categorical data, IEEE Transaction on Visualization and Computer Graphics 12 (4) (2006) 558–568.
[34] C. Bachmaier, A radial adaptation of the sugiyama framework for visualizing hierarchical information, IEEE Transaction on Visualization and Computer Graphic 13 (3) (2007) 583–594.
[35] D.A. Keim, F. Mansmann, J. Schneidewind, T. Schreck Monitoring network traffic with radial traffic analyzer visual analytics and technology, in: proceedings IEEE Symposium on Visual Analytics Science and Technology 2006 (VAST 2006), 2006, pp. 123–128.
[36] R. Vliegen, J.J. van Wijk, E.-J. van der Linden, Visualizing business data with generalized treemaps, IEEE Transaction on. Visualization and Computer Graphics 12 (5) (2006) 789–796.
[37] U. Brandes, P. Kenis, D. Wagner, Communicating centrality in policy network drawings, IEEE Transaction on Visualization and Computer Graphics 9 (2) (2003) 241–253.
[38] E. Di Giacomo, W. Didimo, G. Liotta. Radial drawings of graphs: geometric constraints and trade-off, In: Proceedings of seventh Graph Drawing Conference (GD'06), 2006, pp. 355–366.
[39] D.A. Keim, Designing pixel-oriented visualization techniques: theory and applications, IEEE Transaction on Visualization and Computer Graphics 6 (1) (2000) 59–78.
[40] M. Kaufmann, D. Wagner, in: Drawing Graphs, Springer, 2001.
[41] R. Rudell, Dynamic variable ordering for ordered binary decision diagrams. in: Proceedings of the 1993 IEEE/ACM International Conference on Computer-aided Design (ICCAD '93), 1993, pp. 42–47.
[42] C. Matuszewski, R. Schönfeld, P. Molitor, UsingSifting for *k*-layer straightline crossing minimization, in: Proceedings of seventh Graph Drawing Conference (GD '99), 1999, pp. 217–224.
[43] C. Ware, H. Purchase, L. Colpoys, M. McGill, Cognitive measurements of graph aesthetics, Information Visualization 1 (2) (2002) 103–110.
[44] W. Huang, P. Eades. How people read graphs, in: Proceedings Asia Pacific Symposium on Information Visualization (APVIS 2005), 2005, pp. 47–53.
[45] P. Ren, Y. Gao and Z. Li, IDGraphs: Intrusion Detection and Analysis Using Histographs, in: Proceedings of VizSEC 2005, Minneapolis, Minnesota, USA, 2005, pp. 39–46.
[46] E.R. Tufte, in: The Visual Display of Quantitative Information, Graphics Press, 1983.
[47] Harri Siirtolam, Kari-Jouko Raiha, Interacting with parallel coordinates, Interacting with Computers 18 (6) (2006) 1278–1309.
[48] D.A. Keim, Information visualization and visual data mining, IEEE Transactions on Visualization and Computer Graphics 8 (1) (2002) 1–8.
[49] Tominski C., Abello J., Schumann H. Axes-based visualizations with radial layouts, in: Proceedings of the ACM Symposium on Applied Computing 2004 (Nicosia, Cyprus, 2004), ACM Press: New York, 2004, pp.1242–1247.
[50] R.R. Kasemsri. A Survey, taxonomy, and analysis of network security visualization techniques. [Master thesis], 2005, Georgia State University.