Proceedings of the New Trends in Information Technology (NTIT-2017).
The University of Jordan, Amman, Jordan. 25-27 April 2017.

Enhancing the Visual Perception of Graph Communities

Arwa Aldabobi, Jamal Alsakran
The University of Jordan
Amman, Jordan
E-mail: arwadabobi88@gmail.com, j.alsakran@ju.edu.jo

Abstract

Graph G (V, E) is the mathematical answer for a system that consists of elements that have relationships among them. It is useful to visualize the communities within such systems, especially when they are large ones, that are the densely connected vertices after detecting them using one of the existing clustering algorithms and qualified by certain criteria like the commonly used index, modularity. This paper sheds the light upon an important problem that even a high modularity clustered graph visualization suffers from, which is the interleaving between clusters that yields overlapping areas confusing the user and his visual information grasp. We propose a solution for this problem enhancing the visual result, and an objective metric to assess the modification on the final visualization is introduced.

Keywords: Graph Community detection, Clustered graph visualization, Overlapping

1. INTRODUCTION

Many systems in our world tend to be represented as a graph G (V, E) such that vertices (V) represent elements in that system while edges (E) represent relationships among them, transportation networks and social networks are among these systems. Graphs can be visualized using node-link diagram as fig. 1 shows the graph representing the Zachary karate club in which vertices are the club members while edges are the friendship relations between them. To draw a graph there are many important issues must be taken into consideration to make sure that the graph represents the system with a nice view to the user, so many graph drawing algorithms exist. Among these issues are the aesthetics of the final graph layout that help the user to understand and discover the underlying structure of the represented systems, for example, it is nice to evenly distribute drawing of nodes and edges and to minimize the crossing edges [1]. These aesthetics can be easily applied in case of small graphs, but when graphs become larger in size and more complex then many problems will arise. A promising solution is to express the large and complex graph by its distinct communities; within each there are dense interacted elements instead of representing it by all its elements. Many community detection algorithms can detect the communities that reveal the real structure of a given system, to assess how much these communities can do so, many measurements are presented to evaluate the quality of the detected communities, modularity proposed by Newman and Girvan [2] is among these measurements. From the fact that community is a group in which the elements interact more frequently, so if there is a measure that gives the level of interaction within a community compared with the interaction outside it, it is supposed to have a high value with strongly connected communities, modularity does so.

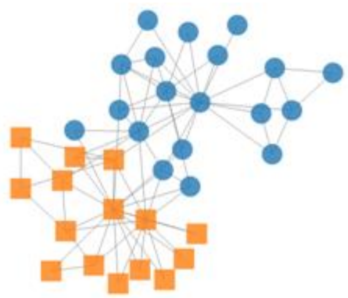


Fig. 1. A graph that represents Zachary karate club.

Because that interactions are represented by edges, modularity measures the quality of communities by comparing the edge density in a community with that outside the community, such that it is given by a numeric value between 0~1 and can be represented mathematically as:

$$Q = \frac{1}{2m} \sum [\, Aij - \frac{KiKj}{2m} \,] \delta(Ci, Cj), \qquad (1)$$

where Aij is the weight of the edge between nodes i and j, $Ki = \sum_j Aij$ is the sum of the weights of the edges attached to node i, C_i is the community of node i, $m = \frac{1}{2} \sum_{ij} Aij$ and the function $\delta(u,v)\, is\, 1\, if\; (u = v)\, and\, 0\, otherwise.$

As it is not expected, a high modularity of a graph clustering does not necessarily mean that their visualization will serve the user. In many systems there is no distinct structure, so it is better to represent such systems as a piece of information. On the other hand, there are systems that have a number of strongly connected communities and are measured by a high modularity, but their visualization are not clear to express them. A problem of interleaving between the detected communities and forming multiple overlapping areas may confuse the user perception of the real system structure, so degrade the purpose of visualization. Thus, a new measurement that detects the overlapping between the detected communities and a solution of this visually important problem is proposed in this paper.

The paper is organized as follows. A brief literature review is presented in section 2. The proposed solution is described in section 3. In section 4, results and discussion are presented. Finally, conclusions are made in section 5.

2. RELATED WORK

Many abstraction methods to simplify large graphs are suggested, so users can view them in a single picture and navigate through them, moreover the aesthetic rules can be easier applied to their final layout. Among these methods are the structure-based ones that exploit the graph topology and compress the similar parts of it. Given a large graph G with its adjacency matrix as $w = \{w_{ij}\}_{ij=1\sim n}$ and the row vector of the i^{th} node as $R_i = \{w_{ij}\}_{j=1\sim n}$, Shi et al. [3] exploit the Structural Equivalence Grouping method (SEG) [4] to simplify a large graph by grouping the node i with all other nodes having the same row vector into a single super node. PhraseNet [5] uses the similar idea to present the main subjects and relations among a document or set of documents by detecting the structure of a graph that consists of the document words as its nodes and a user-specified relation as its edges. Dunne and Shneiderman [6] present their morif simplification technique that replaces a group of nodes and edges with meaningful and compressed glyphs. To obtain a higher compression level for large graphs, community detection algorithms that use measurement other than the graph structure like modularity, can be used. FADE [7] by Quigley and Eades is a fast algorithm to draw a geometric clustering and multilevel viewing of large directed graphs. HiMap system [8] visualize a clustered graph of a large social networks using hierarchical summarization. ASK-GraphView by Abello et al. [9] is an overview+detail approach. Among abstraction methods also the context-based ones, that depend on the node attributes to build an explainer graph or a complementary one for the original large graph. As an example, Wettenberg proposes PivotGraph [10] that study type of graphs in which nodes can be associated with several attributes and groups the nodes that have the same value of one or two attributes. Shen et al. [11], propose a visual analytics tool, OntoVis, that aims to understand and analysis large and heterogeneous social networks that consist of concepts and relations as its nodes and edges by pruning them using the information in their associated ontology, allowing to use the structural abstraction to do filtering and facilitating the analysis process among them. OnionGraph [12], is a flexible framework that allows to group nodes based on the graph topology, nodes attributes or both. Transforming a large graph depending on its layout to minimize the visualization complexity caused by the large size by filtering and grouping the graph edges is another example of the abstraction methods. Jia et al. [13] suggest to filter weak edges according to an edge centrality measure, this can reduce the visual complexity preserving the topology properties of the original large graph. Although Van Ham et al. [14] suggest to construct a minimal spanning tree of the large graphs discovering their underlying structure, many topology information are lost. Edge bundling approaches [15, 16, 17, and 18] suggest to group similar edges within a cluster instead of removing them from the graph. Navigation interactions with an abstracted large graph into a hierarchical structure can be above traversal, below traversal, range traversal or unbalanced traversal as they are classified by Elmqvist and Fekete [19]. GrouseFlocks [20] exploits the attribute associated with the graph nodes and edges to provide the user with several possible hierarchies, users can create and modify the graph hierarchies manually or based on a provided attributes patterns. Van Ham and Perer support search through graphs viewing its context based on the user demand [21].

3. METHODOLOGY

The proposed solution of the overlapping problem between the detected communities of a graph is described as a flowchart in Fig. 2. Given a graph G (N, E) such that each node associated with the attribute "id" that identifies this node and each edge associated with the attributes source and target that are the nodes this edge connecting them together. The proposed solution can be divided into two basic stages;

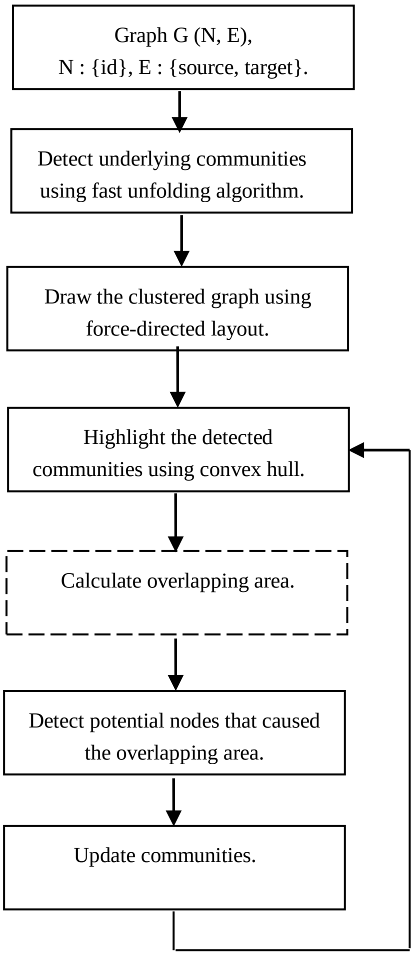


Fig. 2. Flowchart describes the proposed solution procedures.

3.1 Stage One

This stage consists of two main steps; detecting the underlying communities of the given graph and then plotting the visualization of the clustered graph. To detect the graph communities, fast unfolding of communities in large networks algorithm by Vincent et al. [22] is used in our proposed solution because its ability to detect good quality communities in a low computation time. Then, to draw the visualization of the clustered graph we employ the force-directed layout that was first proposed by Peter Eades [23] because its ability to draw a well-balanced layout. So, at the end of this stage each node has a new attribute which is the community it belongs to, such that N: {id, modClass}.

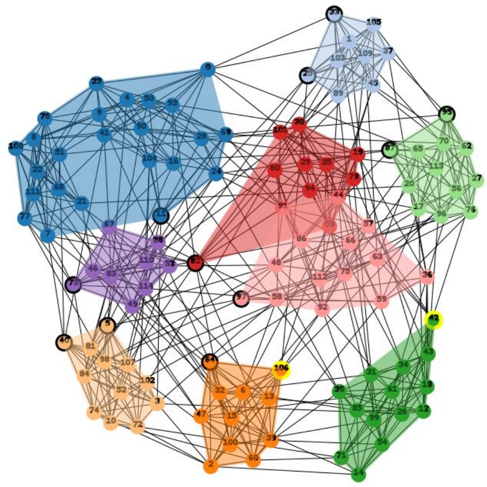
3.2 Stage Two

First of all, we need to visually highlight the detected communities within the given graph, so Graham scan algorithm for drawing a convex hull with O(nlogn) time complexity [24] is used, and a distinct color is given for each convex, see fig. 3 that shows a visualization of the clustered graph representing football dataset that has in its structure nine clusters. As it is obviously clear now the interleaving between the detected communities that will degrade the visualization process and confuse the user notion about the system underlying structure. Interleaving communities is calculated before any modification is done and after each modification to show that the proposed solution will eliminate the overlapping area in a number of studied cases and will at least decrease it in other cases for a reason that will be explained later on.

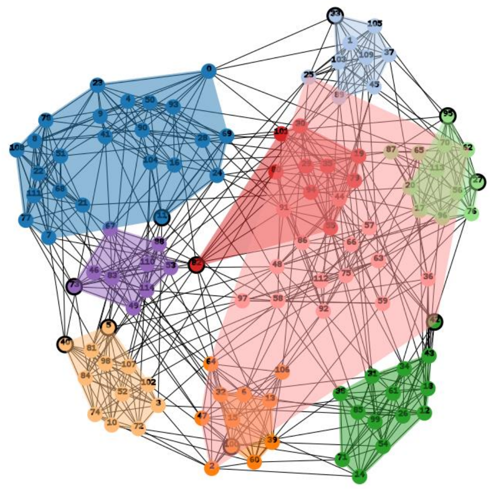
Overlapping area will be the objective measurement to assess the proposed solution, and it is calculated using triangle area equation if the overlapping area makes a triangle shape, otherwise, any other shapes will be divided into a number of triangles and the area will be the sum of this subdivision triangles. The proposed solution starts by detecting the potential nodes that cause the problem of overlapping, to detect them there are three choices, the first one is to compare the two sides attached to each head of each convex hull to the median length of all that convex hull sides, if both are larger than the median then this head will be added to the potential nodes set, otherwise no modification is needed. Fig. 3. (a) shows that the detected potential nodes according to the median choice will be the two highlighted nodes (42, 106) such that they have attached sides with a length exceed the median length of the convex hull sides. After that the potential nodes set has been detected, the modification is to move each potential node to a neighbour convex hull such that its both attached sides do not exceed the new convex hull median length. Why median is chosen; because it is supposed to produce a length exactly equals one of the tested convex hull sides length, and we assume that this choice will give a solution which is near to the reality. Despite this, many nodes that are far away from being a reason of the overlapping problem will be modified into a new cluster. As fig. 3. (b) shows that after five runs of modification the overlapping problem is not solved, but more destroy the underlying system visualization. To avoid unnecessary movements that may not solve the problem, but even more will destroy the visualization, the comparison with the median length is replaced with the average length of the convex hull sides, more acceptable modifications are obtained, and this is the second choice, see fig. 4. After studying the problem of overlapping, it is noticed that the overlapping problem is caused mainly by the points of the convex hull that are included inside another convex hull, but not as their heads, these heads are the potential nodes set that will be handled to eliminate or at least decrease the overlapping area problem in the third choice. This choice detects the potential nodes set such that:

Potential nodes set = all overlapping nodes ⋂ convex hull heads, suppose potential nodes set is Q = {P1, P2,..., Pn}.

Fig. 5. (a) shows that node 53 for example, is a head of the dark blue convex hull that represents the cluster number 2 and this node seems to be a reason of the overlapping occurred between the detected clusters, so if it is moved from its current convex hull to the orange convex hull that will include it inside it, the overlapping will be decreased, and if so is done to the node 44 then the overlapping area will be totally eliminated as it is shown in fig. 5. (c). Now, in the modification step to handle the overlapping problem a search about a new convex hull that includes the potential nodes and handle the overlapping problem is done by check all the potential nodes neighbours that belong to other distinct convex hulls, in case that the potential node does not have any neighbours except the ones it has in its current convex, no modification is done, and this is the reason that the proposed solution does not eliminate the overlapping area in some cases. In case that, for example take the potential node P0, has neighbours belonging to other convex hulls, say neighbour = {n1, n2, n3} such that the community of n1 is not the same as the community of n2 and n3 nor the community of P0, now the proposed solution is to move node P0 from its current community to the community of n1 if it will be included in the convex hull that represent n1 community, if it is not included then continue to the rest of the neighbours communities doing the same, finally if P0 is moved to a new community that include it inside, the convex hulls are re-drawn and the overlapping area is re-calculated. A problem that may result after this modification is shown in fig. 6. (c), that it is left a number of nodes of the community that are less than three points and cannot be drawn as a convex hull, this is solved in this case by adding them also to the P0 new community.

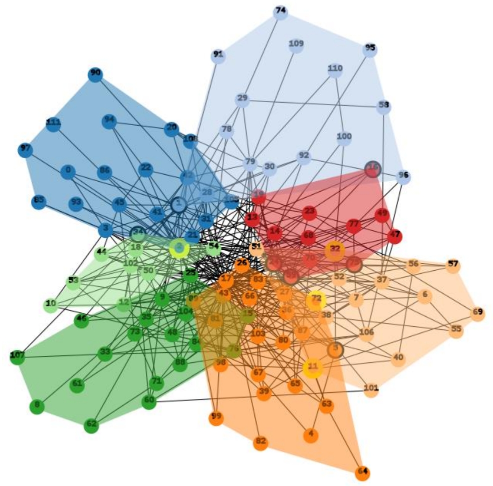


(a) Initial clustering result, modularity = 0.6027, overlapping area = 3,062.

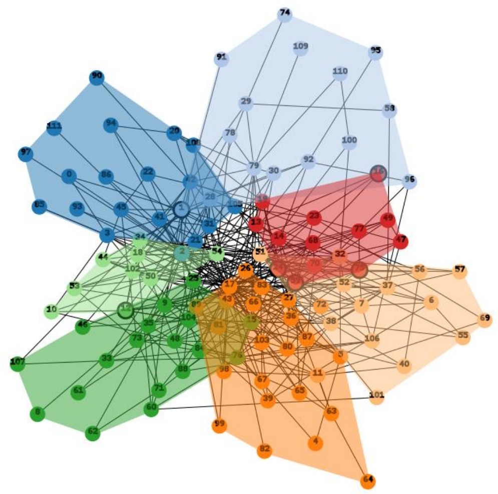


(b) Final modification result, modularity = 0.5237, overlapping area = 20,042.

Fig. 3. Football dataset, median choice, |N| = 115, |E| = 613.



(a) Initial clustering result, overlapping area = 14,229.



(b) Final clustering result, overlapping area = 8,328.

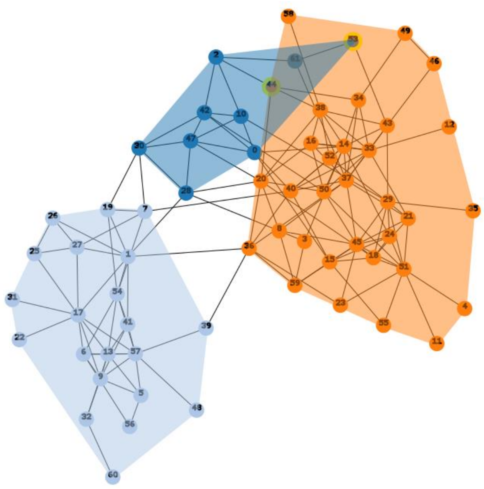
Fig. 4. Adjnoun dataset, modularity = 0.2745, average choice, |N| = 112, |E| = 425.

4. RESULTS AND DISCUSSION

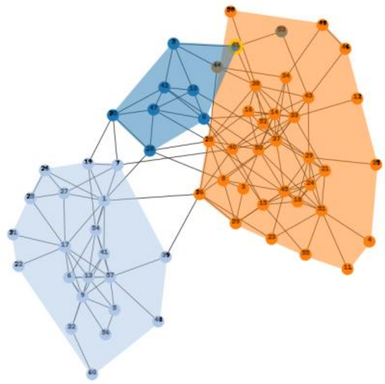
Overlapping problem between graph detected clusters is examined within four popular datasets that are described in TABLE 1. Modifications on Dolphin dataset is shown in Fig. 5. Fig. 5. (a) shows the first modification on the graph in which node 44 is moved from cluster 1 into cluster 2 and node 53 is moved from cluster 2 into cluster 1, modularity after this modification is reduced from 0.4607 to 0.4512 decreased by 2.06% while overlapping area is reduced from 5760 to 630 decreased by 89.04%. It is obvious that there is still an overlapping between communities, so fig. 5. (b) shows the second modification that moves node 6 from cluster 2 into cluster 1 decreasing modularity to 0.4532 by 1.62%, while the overlapping area is now totally eliminated to 0. Note that the modularity in the second modification is increased by 0.9% from the modularity after the first modification. Also modularity is decreased by 1.62%, the graph communities are now separated and visually clear to the user notion, the final result is shown in fig. 5. (c).

TABLE 1. Used dataset description.

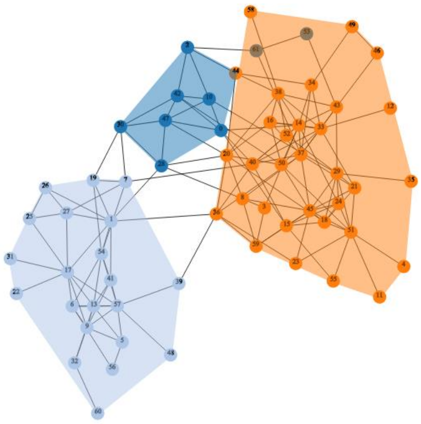
Dataset name	**Dataset description**	**Nodes number in the dataset**	**Edges number in the dataset**
Dolphin	Social network of frequent associations between dolphins in a community in Doubtful Sound, New Zealand.	62	159
Lesmis	Coappearance network of characters in the novel Les Miserables.	77	254
Adjnoun	Adjacency network of common adjectivies and nouns in the novel David Copperfield by Charles Dickens.	112	425
Football	Network of American football games between Division IA colleges during regular season Fall 2000.	115	613



(a) Interleaving problem between the detected clusters. Initial clustering result, modularity = 0.4607, number of overlapping nodes between clusters (1,2) = 4, intersection heads between clusters (1,2) are nodes (44,53), overlapping area = 5,750.



(b) Second modification, modularity = 0.4512, number of overlapping nodes between clusters (1,2) = 3, intersection heads between clusters (1,2) is node (61), overlapping area = 630.

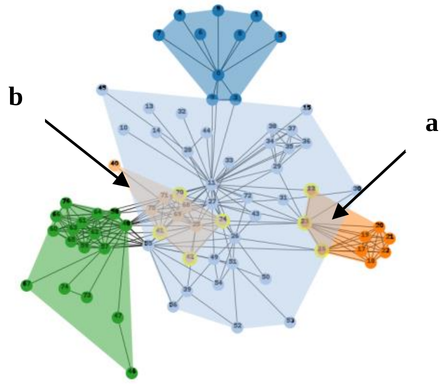


(c) Final result, modularity = 0.4532, number of overlapping nodes = 0, overlapping area = 0. Modularity is decreased by 1.62% and overlapping area is decreased by 100%.

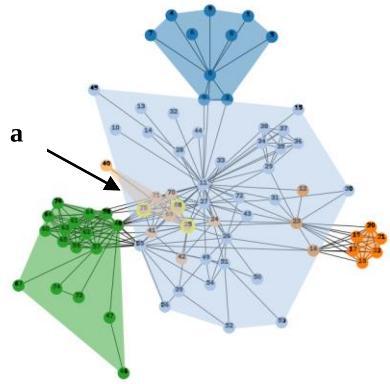
Fig. 5. Move overlapping coordinates choice, Dolphins dataset, |N| = 62, |E| = 159.

Fig. 6. shows the modification on Lesmis dataset, fig. 6. (a) shows that there are two overlapping areas referred to by arrow a and arrow b, the first overlapping area will be eliminated by moving the three overlapping convex heads while the second one will be decreased by moving the four overlapping heads, this will decrease the modularity by 15.56% while decreasing the overlapping area by 77.57% as fig. 6. (b) shows.

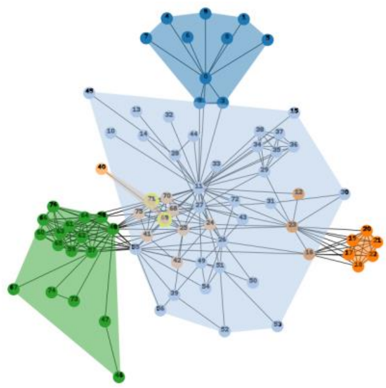
After this modification and as it is illustrated in fig. 6. (b) there is still an overlapping area that will be further decreased by moving the overlapping heads, but this modification will leave three nodes to form a convex hull as it is shown in fig. 6. (c), in the next level of modifications two of them are an overlapping heads that will be moved to a neighbor cluster according to the proposed solution and so a single node is now in their previous cluster that is definitely cannot form a cluster by its own, so in this case it will be added into its neighbors cluster despite it is not an overlapping head. So, the restriction is if there are left number of nodes less than three nodes that cannot represent a single cluster and cannot be included within a convex hull, it will be added to the cluster its neighbors are added into. The final visualization result that clearly expresses the underlying structure of the system by separated clusters is shown in fig. 6. (d) in which overlapping area is totally eliminated.



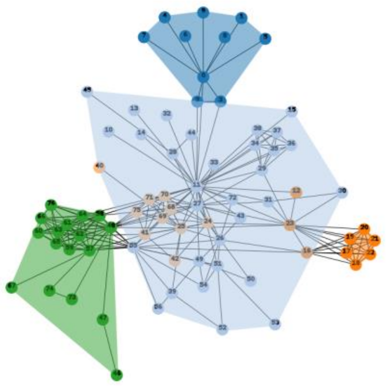
(a) Modularity = 0.5307, overlapping area = 13,538.



(b) Modularity = 0.4481, overlapping area = 3,036.



(c) Modularity = 0.4494, overlapping area = 677.



(d) Modularity = 0.4821, overlapping area = 0, node 40 will be left alone in its cluster and it is clear that it will not form a convex hull, so it will be moved also to the convex hull that includes it previous same cluster nodes.

Fig. 6. Lesmis dataset, move overlapping head choice, |N|= 77, |E|=254 .

5. CONCLUSIONS

As systems become larger in size, representing them as graphs become harder because the associated issues like edge density and crossing, also interleaving between communities that degrade visualization process and confuse users notion about the underlying systems structure. In this paper we study the problem of interleaving between communities and the overlapping area between them and propose a solution to eliminate or at least decrease these areas enhancing the final community visualization that will be assessed by measuring the overlapping area between them.

REFERENCES

[1] Tamassia, R. (Ed.). (2013). Handbook of graph drawing and visualization. CRC press.

[2] Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. Physical review E, 69(2), 026113.

[3] Shi, L., Liao, Q., Sun, X., Chen, Y., & Lin, C. (2013, October). Scalable network traffic visualization using compressed graphs. In Big Data, 2013 IEEE International Conference on (pp. 606-612). IEEE.

[4] Lorrain, F., & White, H. C. (1971). Structural equivalence of individuals in social networks. The Journal of mathematical sociology, 1(1), 49-80.

[5] Van Ham, F., Wattenberg, M., & Viégas, F. B. (2009). Mapping text with phrase nets. IEEE transactions on visualization and computer graphics, 15(6).

[6] Dunne, C., & Shneiderman, B. (2013, April). Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 3247-3256). ACM.

[7] Quigley, A., & Eades, P. (2000, September). FADE: Graph drawing, clustering, and visual abstraction. In International Symposium on Graph Drawing (pp. 197-210). Springer Berlin Heidelberg.

[8] Shi, L., Cao, N., Liu, S., Qian, W., Tan, L., Wang, G., ... & Lin, C. Y. (2009, April). Himap: Adaptive visualization of large-scale online social networks. In Visualization Symposium, 2009. PacificVis' 09. IEEE Pacific (pp. 41-48). IEEE.

[9] Abello, J., Van Ham, F., & Krishnan, N. (2006). Ask-graphview: A large scale graph visualization system. IEEE transactions on visualization and computer graphics, 12(5), 669-676.

[10] Wattenberg, M. (2006, April). Visual exploration of multivariate graphs. In Proceedings of the SIGCHI conference on Human Factors in computing systems (pp. 811-819). ACM.

[11] Shen, Z., Ma, K. L., & Eliassi-Rad, T. (2006). Visual analysis of large heterogeneous social networks by semantic and structural abstraction. IEEE transactions on visualization and computer graphics, 12(6), 1427-1439.

[12] Shi, L., Liao, Q., Tong, H., Hu, Y., Zhao, Y., & Lin, C. (2014, March). Hierarchical focus+ context heterogeneous network visualization. In Visualization Symposium (PacificVis), 2014 IEEE Pacific (pp. 89-96). IEEE.

[13] Jia, Y., Hoberock, J., Garland, M., & Hart, J. (2008). On the visualization of social and other scale-free networks. IEEE transactions on visualization and computer graphics, 14(6), 1285-1292.

[14] Van Ham, F., & Wattenberg, M. (2008, May). Centrality based visualization of small world graphs. In Computer Graphics Forum (Vol. 27, No. 3, pp. 975-982). Blackwell Publishing Ltd.

[15] Cui, W., Zhou, H., Qu, H., Wong, P. C., & Li, X. (2008). Geometry-based edge clustering for graph visualization. IEEE Transactions on Visualization and Computer Graphics, 14(6), 1277-1284.

[16] Lambert, A., Bourqui, R., & Auber, D. (2010, June). Winding roads: Routing edges into bundles. In Computer Graphics Forum (Vol. 29, No. 3, pp. 853-862). Blackwell Publishing Ltd.

[17] Holten, D. (2006). Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. IEEE Transactions on visualization and computer graphics, 12(5), 741-748.

[18] Gansner, E. R., Hu, Y., North, S., & Scheidegger, C. (2011, March). Multilevel agglomerative edge bundling for visualizing large graphs. In Pacific Visualization Symposium (PacificVis), 2011 IEEE (pp. 187-194). IEEE.

[19] Elmqvist, N., & Fekete, J. D. (2010). Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. IEEE Transactions on Visualization and Computer Graphics, 16(3), 439-454.

[20] Archambault, D., Munzner, T., & Auber, D. (2008). GrouseFlocks: Steerable exploration of graph hierarchy space. IEEE transactions on visualization and computer graphics, 14(4), 900-913.

[21] Van Ham, F., & Perer, A. (2009). "Search, show context, expand on demand": supporting large graph exploration with degree-of-interest. IEEE Transactions on Visualization and Computer Graphics, 15(6).

[22] Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10), P10008.

[23] Eades P (1984) A heuristic for graph drawing. Congressus Numerantium 42: 149–160.

[24] o'Rourke, J. (1998). Computational geometry in C. Cambridge university press.