A simple multiforce layout for multiplex networks

Zahra Fatemi, Mostafa Salehi, and Matteo Magnani

Abstract—We introduce multiforce, a force-directed layout for multiplex networks, where the nodes of the network are organized into multiple layers and both in-layer and inter-layer relationships among nodes are used to compute node coordinates. The proposed approach generalizes existing work, providing a range of intermediate layouts in-between the ones produced by known methods. Our experiments on real data show that multiforce can keep nodes reasonably aligned across different layers without significantly affecting the internal layout of each layer.

Index Terms—Layout, Multiplex Network, Force-directed, Visualization.

1 INTRODUCTION

Network analysis is a widely applied discipline, due to the fact that many realities can be modeled as sets of interconnected entities – for example, social networks, technology networks, transportation networks, utility networks and biological networks. In the following, we use the term monoplex to indicate a network consisting of nodes with a single type of relationship among them. The simplicity of monoplex network models is at the same time a strength, making them applicable to diverse contexts, and a limit, because it may hide many details of the modeled reality. Therefore, multiplex networks are often used as a richer but still simple and general model, as they allow multiple types of relationships between nodes. For example, two people in an online social network like Facebook can be friends, while being colleagues in a work environment.

Many aspects of multiplex networks from spreading processes to structural measures such as clustering coefficient and node centrality have been recently investigated [21], [3], [30], [2], [16], [27], [9], building on a long-standing literature in Social Network Analysis [32], but only a few works have specifically investigated how to draw multiplex networks. Several layouts have been proposed to visualize monoplex networks, arranging nodes so that users can easily identify special network structures like hubs or communities, and quickly locate important nodes [31], but drawing multiplex networks is significantly more challenging.

A multiplex network visualization should support two main types of tasks: the analysis of the single layers, each corresponding to a traditional monoplex network for which existing graph layouts can be used, and the analysis of the relationships between layers, for which it is useful to identify where the same nodes or network structures are located on the different layers [9].

• Zahra Fatemi and Mostafa Salehi are with the Department of New Science and Technology, University of Tehran, Iran.
  E-mail: fatemi.z@ut.ac.ir, mostafa.salehi@ut.ac.ir
• Matteo Magnani is with Uppsala University.
  Email: matteo.magnani@it.uu.se

Fig. 1: Two alternative ways of visualizing a multiplex network with 180 nodes and 863 edges of 6 types

Representing multiple relationship types in the same graph as in Figure 1(a) can quickly lead to a very dense representation hiding relevant network structures [26]. Therefore, different relationship types can be organized into different layers, with the same node replicated on multiple layers, as in Figure 1(b). This approach has been used in the literature to represent different types of multiplex networks, from traffic networks [19], [2] to biological [7] and social/historical [22], [21] networks.

In the literature, two main approaches for visualizing multiplex networks sliced into layers have been used: keeping the same layout in all layers, meaning that...
In our view, these approaches are just two specific cases of a more general method. To fill the gap between these two extremes, we propose to define a range of general layouts with a controllable balance between in-layer and inter-layer relationships. We call our general approach multiforce.

Multiforce is based on a force-directed algorithm (Fruchterman-Reingold) and uses two main types of forces: *in-layer* and *inter-layer*, that can be tuned to impact specific layers more or less than others. In-layer forces attract neighbors inside the same layer, making them closer, as in traditional layouts for monoplex networks. Inter-layer forces try to align instances of the same node on different layers. Figure 2 gives an intuition of how these forces operate. In addition, we also use repulsive forces as in the original algorithm.

Fig. 2: The effect of in-layer and inter-layer forces on node positions

To evaluate our approach, we have applied it to several real networks. A specific problem of multiplex network visualization is that different layers can have different structures, for example, two nodes connected on one layer may be disconnected on another. Therefore, in general, it is impossible to produce a good layout for each layer and keep nodes perfectly aligned across different layers at the same time. As an example, node 2 in the lower layer of Figure 2 would be attracted towards node 1 on the same layer because of the edge between them, but this would increase the distance between the position of node 2 on the lower layer and the position of node 2 on the upper layer. In this way, we would negatively affect our visual understanding of the relationships between the different layers, losing our ability to quickly locate the same node across layers.

Thus, we have defined two measures representing these two criteria (internal and external fit) and shown how existing approaches would optimize only one of them, while our algorithm can obtain good scores on both at the same time. In addition to this evaluation, we have also executed our algorithm on a simple dataset to characterize the resulting diagrams.

In Section 2, we quickly review existing layouts for monoplex and multiplex networks. In Section 3, our multiforce algorithm is introduced. Section 4 presents an example using a small synthetic dataset, to characterize the impact of different settings, while Section 5 proposes an evaluation on real data.

2 RELATED WORK

In this section, we describe previous works on monoplex and multiplex network visualization.

2.1 Monoplex Network visualization

Many layouts have been designed to visualize monoplex networks. Here we briefly review the ones that are more relevant for our approach. For an extensive review, the reader may consult [31].

*Multi-scale layouts* first create some core subgraphs, then they add other nodes until all nodes and edges are added [31], [17]. Random layouts [8] and circular layouts [20] are two categories of layouts which are appropriate for small graphs with few nodes and edges, because they do not consider aesthetic criteria: many edge crossings and node overdrawings can appear.

Among the most used visualization methods, *force-directed algorithms* consider a graph as a physical system where forces change the position of nodes. The two best known force-directed layouts are Fruchterman-Reingold [11] and Kamada-Kawai [15], [14]. In the Fruchterman-Reingold layout, nodes have repulsive power and push other nodes away, while edges attract neighbor nodes. In this layout, nodes are considered as steel rings having similar loads and edges are like springs attracting neighboring rings. This algorithm consists of three main steps. First, all nodes are distributed randomly. Second, repulsive forces separate all nodes. The value of repulsive force depends on the positions of the nodes. Third, for each edge and based on the position of nodes after repulsion, attractive forces are calculated [11]. In the Kamada-Kawai layout, an energy function is defined for the whole graph based on shortest paths between nodes, and positions are iteratively updated until the graph's energy is minimized [15].

Bannister et al. [11] proposed a force-directed layout to change the position of nodes so that more graph-theoretically central nodes pushed towards the centre of the diagram. In this algorithm, an additional force called gravity is used to change the position of more central
nodes. For each node \( v \) in a graph \( G \) the position of the node is influenced by the following force:

\[
I[v] = \sum_{u, v \in V} f_r(u, v) + \sum_{(u, v) \in E} f_a(u, v) + \sum_{v \in V} f_g(v) \tag{1}
\]

where \( f_r \) and \( f_a \) are respectively repulsive and attractive forces, and \( f_g \) is the gravity force, measured as:

\[
f_g(v) = \gamma_v M[v](\xi - P_v) \tag{2}
\]

In this equation \( M[v] \) is the mass of node \( v \), which can be set according to the node degree, \( P[v] \) is the position of \( v \), \( \gamma_v \) is the gravitational parameter and \( \xi = \sum_v P[v]/|V| \) is the centroid of all nodes. Notice that forces in the equations above are vectors.

Forcedirected methods have two main drawbacks. They are typically suitable for networks with at most 1000 nodes, because attractive forces result in hiding some relationships among nodes and node-overlapping increases. Moreover, their run time is high in comparison with other approaches. However, they are very popular, because they often practically succeed in separating clusters and increasing the graph readability.

Another family of layouts, that can also be combined with force-directed algorithms, are constraint-based layouts [10]. These layouts force nodes to appear at specific positions. For example, nodes are placed on a frame in a way that they do not overlap, or are horizontally and vertically aligned, as in the orthogonal layout [31], [10]. In these layouts, it is more difficult to isolate special structures like communities and time complexity is noticeably high. All these layouts are used in visualization software like Pajek, Gephi and Igraph.

One important assumption in graph drawing is that there is a correspondence between some aesthetic features of the diagrams and their readability. Therefore, some visualization algorithms explicitly target these features. One such criterion is that too many edge crossings make a graph more difficult to interpret. The crossing number, \( cr(G) \), of a graph \( G \) is the smallest number of crossings appearing in any drawing of \( G \) [28]. Several algorithms have been proposed to reduce edge crossing in monoplex networks. For example, Shabbeer et al. [29] developed a stress majorization algorithm. In [28] and [4] the concept of edge crossing is elaborated and equations for measuring the number of edge crossings in different graphs are reviewed. Another aesthetic feature impacting graph readability is node overlapping. Two popular methods to reduce node overlapping are proposed in [12], [18].

### 2.2 Multiplex Network Visualization

Different methods have been proposed for visualizing multiplex networks. We can categorize these methods into three main classes:

- **Slicing:** One way of visualizing multiplex networks is to show each layer or relationship type as a monoplex network and to connect these monoplex networks using inter-layer edges [7]. The layers can have aligned layouts or independent layouts. Aligned layouts help users find similar nodes in different layers by forcing the same node to have the same coordinates on all layers, but structures existing only on one layer (for example communities) may not be clearly visualized. Independent layouts can show specific structures of each layer, but may hide inter-layer patterns [26].

- **Flattening:** In these methods, all nodes and edges are placed on the same plane. In a node-colored network, nodes from different layers are shown with different colors [16], while for multiplex networks colors can be used to distinguish edges of different types. Apart from suffering from the same problems of aligned slicing, the disadvantage of this method is that for networks with high edge density relationships among nodes can be hidden by edges from non-relevant layers and readability quickly declines due to the network’s clutter [26].

In another study, Renoust et al. [24] proposed a system for visual analysis of group cohesion in flattened multiplex networks. This system, called Detangler, creates a so-called substrate network from unique nodes of the multiplex networks and a so-called catalyst network from edges of different types.

- **Indirect:** A third option, that we mention for completeness but is less related to our work, is not to visualize nodes and edges but other network properties, like the degree of the nodes in the different layers or other summary measures [7], [26], [23]. These approaches are complementary to graph drawing.

Our method belongs to the slicing class, and is different from existing approaches because it allows a balancing of the effects of in-layer and inter-layer relationships instead of focusing on only one of them.

### 3 The Multiforce Layout

Multiforce extends the Fruchterman-Reingold algorithm [11], and as mentioned in the introduction is based on two types of attractive forces. The nodes are positioned on a set of planes, one for each layer or type of relationship – this setting is sometimes called 2.5-dimensional, because it looks 3-dimensional but the z-coordinates of the nodes are fixed and limited to the number of planes/layers. In-layer forces, that can be weighted differently in each layer, attract pairs of nodes connected on the same layer. Inter-layer forces influence the position of nodes in different layers connected by inter-layer edges, or corresponding to the same node in the case of multiplex networks.

The pseudo-code of multiforce is presented in Algorithm [1]. The algorithm takes a multiplex network \( G = (N, L, V, E) \) as input, where \( N \) is a set of nodes, \( L \) a set of layers, \( (V, E) \) is a graph and the elements of
V are pairs (node, layer). We notate v.node the layer of an element v ∈ V and v.node the node corresponding to an element v ∈ V.

Lines 13-29 are the same as in the original algorithm, and compute the displacement of each node based on its neighbors (attractive forces) and other nodes (repulsive forces), with the addition of weights that can be used to specify on which layers the layout should be computed according to the original algorithm (27-28). Lines 30-37 extend the original algorithm and compute the displacement caused by the position of the node on other layers, to control node alignment. This is also weighted, to allow the user to turn this feature on and off for all or some layers (34-35).

Some details of the algorithm can be changed without affecting its underlying idea. First, we can modify lines 6-12 to assign the same initial random coordinates to the same node across different layers, anticipating line 8 before the for loop. A weighting factor INLA[v] can also be added at line 20, so that both attractive and repulsive forces are reduced or reinforced together — in practice, this does not seem to have a significant effect on the result; all diagrams in Section 4 have been computed without these weights. Finally, lines 41 and 42 have been retained from the original algorithm and ensure that the nodes do not exit the frame specified by the user, but are not necessary if the final coordinates are re-scaled to fit it.

3.1 Time Complexity

Separating nodes based on repulsive forces has time complexity O(|N|^2) for each layer. For a complete network with |N| nodes and |L| layers, there are at most (|L||N|(|N|−1)/2) in-layer edges and (|L|(|L|−1)|N|) inter-layer relationships in the whole network. So, the time complexity of multiforme without using indexes is O(|L||N|^2 + |L|^2|N|).

4 A SIMPLE EXAMPLE

Before providing a quantitative evaluation of our algorithm, we show the resulting layouts on a simple synthetic network and different weights, to give a visual intuition of it. As we have briefly discussed in Section 2, there is a connection between aesthetic features of graph diagrams and their readability [14]. Typical metrics are the number of edge crossings, the number of overlapping nodes, the separation of communities, and the representation of high degree centrality nodes in a specific position, for example in the centre of each layer. These features can be easily manually inspected in the following diagrams. In addition, the simple network used in this section allows us to check the impact of different settings for the in-layer and inter-layer weights.

The structures of the synthetic dataset used in the following is shown in Figure 3. This dataset contains two layers, each with 13 nodes; the whole network has 53 in-layer edges. Some nodes that are present in one

Algorithm 1 Multiforme

Require: G = (N, L, V, E); a multiplex network
Require: W: width of the frame
Require: L: length of the frame
Require: #iterations
1: f_r = function(z, k) { return k^2/z; }
2: f_a = function(z, k) { return z^2/k; }
3: area := W · L
4: k := \sqrt{\frac{\text{area}}{|N|}}
5: t := \sqrt{|N|};
6: for (n ∈ N) do
7: for (v ∈ V s.t. v.node = n) do
8: (x, y) := random coordinates;
9: z[v] := index(v.node);
10: end for
11: end for
12: for (i = 1 to #iterations) do
13: // calculate attractive forces inside each layer
14: (u, v) ∈ E do
15: if (u ≠ v and u.layer = v.layer) then
16: Δ := pos[v] − pos[u];
17: disp[v] := disp[v] + (Δ/|Δ|) * f_r(|Δ|);
18: end if
19: end for
20: // calculate attractive forces inside each layer
21: (u, v) ∈ E do
22: if (u ≠ v and u.layer = v.layer) then
23: Δ := pos[v] − pos[u];
24: disp[v] := disp[v] − (Δ/|Δ|) * f_a(|Δ|, k) * INLA[v];
25: disp[u] := disp[u] + (Δ/|Δ|) * f_a(|Δ|, k) * INLA[u];
26: end for
27: // calculate attractive forces across layers
28: (u, v) ∈ E do
29: if (u ≠ v and u.node = v.node) do
30: Δ := pos[v] − pos[u];
31: disp[v] := disp[v] − (Δ/|Δ|) * f_a(|Δ|, k) * INTERLA[v, u];
32: disp[u] := disp[u] + (Δ/|Δ|) * f_a(|Δ|, k) * INTERLA[u, v];
33: end for
34: // assign new positions
35: for (v ∈ V) do
36: pos[v] := pos[v] + (disp[v]/|disp[v]|) * min(disp[v], t);
37: end for
38: // reduce the temperature
39: t := cool(t);
community in one layer belong to another in the second layer, making this small example useful to show how these nodes are handled by varying the weights.

The results of drawing this multiplex network with multiforce and different combinations of weights are shown in Table 1. It is interesting to notice that in the bottom-right plot most of the nodes are aligned, with the nodes being part of different communities in the different layers having slightly different positions. In general, an inspection of the inter-layer connections may reveal which parts of the layers are similar and which parts are significantly different, being characterized by oblique connections.

5 EXPERIMENTAL EVALUATION

Our algorithm does not replace existing approaches: it unifies and complements them with new options. All the available options can be valuable, including those provided by existing methods: for a user it can be good to keep all the nodes aligned, while for another it can be better to visualize the layers one by one. Therefore, we will not try to prove the superiority of one approach with respect to the other, and the main value of our proposal is to have a single flexible algorithm that can emphasize different aspects of the data. However, we can unambiguously show that our algorithm can generate valuable visualizations that cannot be obtained using existing methods.

More precisely, the hypothesis we test is the following. We compare different layouts against a case that we assume being optimal, that is, when each layer is drawn independently of the others, and at the same time all occurrences of the same node across different layers are aligned. Notice that such a diagram is impossible in general, which is why our method allows different compromises to relax either the first or the second features of this optimal scenario. To compute the distance between the ideal case and our tests we can define two measures representing these two criteria: the sum of the forces still active on the nodes inside each layer (internal fit) and the displacement between nodes in different layers (external fit), also computed as the sum of the inter-layer forces still active at the end of the algorithm. Notice that an optimal diagram as defined above would contain stable and aligned nodes, so both measures would be 0. The hypothesis is that existing approaches optimize only one of these two criteria, while multiforce can obtain good scores on both at the same time when using both in-layer and inter-layer forces. In the following we call this setting balanced.

To give an intuition of how a balanced drawing may look like, consider Figure 4 where we have visualized a real 5-layer social network previously used in multiplex visualization research [26]. To make the diagram readable we have drawn each layer besides the others. The different layers are drawn according to their internal organization, showing peculiar structures: for example, the lower community indicated in the top-left diagram (continuous line) is not present in some of the other layers. However, we can see how the nodes belonging to this community have been visualized at similar locations.
TABLE 1: Visualizing the synthetic network using different weights

| In-layer Force=0 | Inter-layer Force=0.5 | Inter-layer Force=1 |
|------------------|-----------------------|---------------------|
| Inter-layer Force=0 | ![Image](image1.png)     | ![Image](image2.png)           |
| Inter-layer Force=0.5 | ![Image](image3.png)       | ![Image](image4.png)           |
| Inter-layer Force=1  | ![Image](image5.png)       | ![Image](image6.png)           |

in the other layers, providing information about inter-layer relationships. We can see, for example, that some of the nodes in this community are not present in the second layer, that this community is split into two sub-communities in the third layer, that it is not present in the fourth layer, where the same nodes are connected to different parts of the graph, and that they form a similar but less dense community in the fifth layer. Notice that in the fourth layer the nodes are still more or less in the same position as in the first layer, despite the fact that they do not form a community. In this way it becomes easier to locate them — in the real 2.5-dimensional visualization we would also have lines connecting nodes across layers, as in the previous section, making the task straightforward as long as the nodes are more or less aligned. In the figure we have also marked a second community with a fuzzy rectangle, so that the reader may observe another example.

To support our claim we have executed the algorithm on eight real datasets from different domains, summarized in Figure 2:

- A criminal network [25].
- A biological network in which every layer shows a synaptic junction (Electric, Chemical Monadic, Chemical Polyadic) [6].
- A transport network with flight connections in Europe [5].

For each network, Figure 5 shows how different layouts behave with respect to the two aforementioned measures, indicated as Intra (internal fit) and Inter (external fit) in the plots. A high value of the former means that some layers have not been visualized according to their internal structure — for example, the nodes in a community may have been spread around the frame instead of being close to each other. A high value of the latter means that nodes are not aligned across layers — for example, a node may have been visualized in the top left corner in one layer and in the top right in the other. In each figure, the first experiment corresponds to a balanced layout, where our algorithm is executed with weight 1 for all inter- and in-layer forces. The second experiment corresponds to the case where all layers are visualized independently of the others. In the remaining experiments the layout is computed based on one of the layers, and nodes are kept aligned on the other layers.

As expected, in the independent visualization (second case) nodes are not aligned (high Inter value), and in the following experiments we can see that computing the layout based on one layer prevents other layers from having good internal layouts. The balanced option (first
TABLE 2: Properties of Real-world Networks

| Network            | # Layers | # Nodes | # Edges |
|--------------------|----------|---------|---------|
| Social – Padgett   | 2        | 15      | 35      |
| Social – Kaptail   | 4        | 39      | 552     |
| Social – Wiring    | 6        | 14      | 108     |
| Hybrid – AUCS      | 5        | 61      | 1240    |
| Social – Physicians| 3        | 241     | 1370    |
| Criminal – Noordin | 4        | 74      | 318     |
| Biological – synapses | 3    | 279     | 3108    |
| Transport – airlines | 37   | 417     | 3588    |

Fig. 5: Results of experiments on real data
case) presents internal layouts that are less good than the ideal case, and node alignments that are also less good than the best possible option, but both are close to optimal and significantly better than the aspect not optimized in other experiments. This corresponds to the case exemplified in Figure 4.

6 CONCLUSION

An optimal layout for multiplex networks would be able to reveal the structure of each layer and the relationships between different layers at the same time. Unfortunately, this is not possible in general, because these two aspects may not be aligned in real data, with some layers being very uncorrelated with the others.

To address this problem, we proposed multiforce, a force-directed algorithm in which both in-layer and inter-layer forces can affect the position of nodes, and users can decide which aspect to emphasize more or less. In-layer forces keep together connected nodes and increase community isolation, while inter-layer forces help users finding nodes in different layers.

In the evaluation of the method we showed that while the algorithm supports more traditional layouts it can also generate what we call balanced visualizations where both internal properties and node alignments are present.

In this paper we only considered static multiplex networks, but nodes and edges can change over time and appropriate layouts considering dynamical features could be valuable tools. In addition, other aesthetic features of graph diagrams like symmetry and uniform edge length could be investigated in the context of multiplex networks.

REFERENCES

[1] Michael J. Bannister, David Eppstein, Michael T. Goodrich, and Lowell Trotter. Force-directed graph drawing using social gravity and scaling. In Proceedings of the 20th International Conference on Graph Drawing, pages 414–425. Springer-Verlag, Berlin, Heidelberg, 2013.

[2] Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. Physical Review E, 89(3):032804, 2014.

[3] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. Multidimensional networks: foundations of structural analysis. World Wide Web, 16(5-6):567–593.

[4] Christoph Buchheim, Markus Chimani, Carsten Gutwenger, Michael Jünger, and Petra Mutzel. Crossings and Planarization. CRC Press, 2013.

[5] Alessio Cardillo, Jesús Gómez-Gardeñes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco del Pozo, and Stefano Boccaletti. Emergence of network features from multiplexity. Scientific reports, 3:1344, 2013.

[6] Beth L. Chen, David H. Hall, and Dmitri B. Chklovskii. Wiring optimization can relate neuronal structure and function. Proceedings of the National Academy of Sciences of the United States of America, 103(12):4723–4728, 2006.

[7] Manlio De Domenico, Mason A. Porter, and Alex Arenas. MuxViz: a tool for multilayer analysis and visualization of networks. Journal of Complex Networks, pages 159–176, 2014.

[8] Josep Díaz, Jordi Petli, and Maria Serna. A survey of graph layout problems. ACM Comput. Surv., 34(3):313–356, September 2002.

[9] Mark E. Dickson, Matteo Magnani, and Luca Rossi. Multilayer Social Networks. Cambridge University Press, 2016.

[10] Tim Dwyer, Kim Marriott, Falk Schreiber, Peter Stuckey, Michael Woodward, and Michael Wybrow. Exploration of networks using overview+detail with constraint-based cooperative layout. IEEE transactions on visualization and computer graphics, 14(6):1293–300, January 2008.

[11] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. Softw. Pract. Exper., 21(11):1129–1164, November 1991.

[12] Xiaodi Huang, Wei Lai, A.S.M. Sajeev, and Junbin Gao. A new algorithm for removing node overlapping in graph visualization. Information Sciences, 177(14):2821 – 2844, 2007.

[13] Herbert Menzel James Coleman, Elihu Katz. The diffusion of an innovation among physicians. Sociology, 20(4):253–270, 1957.

[14] Skye Bender-Moll James Moody, Daniel McFarland. Dynamic network visualization. American Journal of Sociology, 110(4):1206–1241, 2005.

[15] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. Inf. Process. Lett., 31(1):7–15, April 1989.

[16] Mikko Kivelä, Alexandre Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer Networks. Journal of Complex Networks, 2(3):203–271, sep 2014.

[17] Yehuda Koren, L. Carmel, and D. Harel. Ace: a fast multiscale eigenvectors computation for drawing huge graphs. In Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on, pages 137–144, 2002.

[18] Pushpa Kumar and Kang Zhang. Visualization of clustered directed acyclic graphs with node interleaving. In Proceedings of the 2009 ACM Symposium on Applied Computing, SAC ’09, pages 1800–1805, New York, NY, USA, 2009. ACM.

[19] Maciej Kurant and Patrick Thiran. Layered Complex Networks. Physical Review Letters, 96(13):138701, apr 2006.

[20] Ding Ma. Visualization of social media data: mapping changing social networks . Master’s thesis, the Faculty of Geo-Information Science and Earth Observation of the University of Twente, 2012.

[21] Matteo Magnani and Luca Rossi. The ML-Model for Multi-layer Social Networks. In International conference on Social Network Analysis and Mining (ASONAM), pages 5–12. IEEE Computer Society, 2011.

[22] John F Padgett and Paul D McLean. Organizational Invention and Elite Transformation: The Birth of Partnership Systems in Renaissance Florence. American Journal of Sociology, 111(5):pp. 1463–1568, 2006.

[23] Denis Redondo, Arnaud Sallaberry, Dino Ienco, Faraz Zaidi, and Pascal Poncet. Layer-Centered Approach for Multigraphs Visualization. In International Conference on Information Visualisation (IV), pages 50–55, 2015.

[24] B. Renoust, G. Melancon, and T. Munzner. Detangler: Visual analytics for multiplex networks. Computer Graphics Forum, 34(3):321–330, 2015.

[25] Nancy Roberts and Sean F. Everton. Roberts and Everton Terrorist Data: Noordin Top Terrorist Network (Subset), 2011.

[26] Luì Rossi and Matteo Magnani. Towards effective visual analytics on multiplex and multilayer networks. Chaos, Solitons & Fractals, 72:68 – 76, 2015.

[27] Mostafa Salehi, Payam Siyari, and Danilo Montesi. Spread processes in multilayer networks. Network Science and Engineering, IEEE Transactions on, 2(2):65–83, 2015.

[28] Marcus Schaefer. The graph crossing number and its variants: a survey. The electronic journal of combinatorics, 1000:DS21–May, 2013.

[29] P. Shabbeer, C Ozcaglar, M Gonzalez, and KP Bennett. Optimal embedding of heterogeneous graph data with edge crossing constraints. In Presented at NIPS Workshop on Challenges of Data Visualization, page 1, 2010.

[30] Albert Solé-Ribalta, Manlio De Domenico, Sergio Gómez, and Alex Arenas. Centrality rankings in multiplex networks. In Proceedings of the 2014 ACM Conference on Web Science, WebSci ’14, pages 149–155, New York, NY, USA, 2014. ACM.

[31] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J.J. van Wijk, J.-D. Fekete, and D.W. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. Computer Graphics Forum, 30(6):1719–1749, 2011.

[32] Stanley Wasserman and Katherine Faust. Social Network Analysis: Methods and Applications, volume 8 of Structural analysis in the social sciences. Cambridge University Press, 1994.