PAPER *Special Section on Knowledge, Information and Creativity Support System*

Anchored Map: Graph Drawing Technique to Support Network Mining

Kazuo MISUE[†a)], ***Member***

SUMMARY Because network diagrams drawn using the spring embedder are not easy to read, this paper proposes the use of "anchored maps" in which some nodes are fixed as anchors. The readability of network diagrams is discussed, anchored maps are proposed, and a method for drawing anchored maps is explained. The method uses indices to decide the orders of anchors because those orders markedly affect the readability of the network diagrams. Examples showing the effectiveness of the anchored maps are also shown.

key words: *network visualization, graph drawing, anchored map, knowledge mining*

1. Introduction

Information with network structures is observed in various scenes of the real world. Such information has attracted the attention of a lot of researchers, and graphic tools that make network structures visible have become important for analyzing network information.

The purpose of this research is to develop visualization techniques to give overviews of networks with high readability for mining knowledge. The purpose of giving the overviews is to support the initial extraction of useful knowledge from the network information. When we process large-scale numeric data, we usually draw charts (e.g., scatter charts and histograms) that help us grasp overall tendency of the data. When we process network information, however, we notice that we do not have enough techniques and tools to manipulate the network information.

We propose to improve readability of the diagrams by introducing viewpoints that are dependable points of the observation. It is possible to visualize network information as diagrams, but it is often difficult to read any useful knowledge from them. Introducing viewpoints into network diagrams is expected to enable an observer to read useful information from the diagrams adequately.

As an approach to introduce viewpoints on diagrams drawn by using the spring embedder [4], which is often used to layout networks, we chose to fix some nodes at predetermined positions. The fixed nodes have an effect like that of coordinate systems and perform functions as viewpoints. The author named this drawing style "anchored map." The anchored map first proposed had just three anchors, but four or more viewpoints are often necessary. General techniques for drawing anchored maps with four or more fixed nodes are therefore desired.

This paper describes a method for drawing anchored maps with four or more fixed nodes and presents some examples illustrating the effectiveness of the anchored maps.

2. Basic Idea of Anchored Map

An anchored map is a node-link diagram based on the spring embedder and following the conventional straight-line representation. It restricts some nodes, called "anchors," to certain positions or areas and allows the others, called "free nodes," to be arranged freely.

2.1 Readability of a Free Layout

The easily implemented spring embedder is the most widely used of the graphing tools that use force-directed placement to produce an undirected graph [7]. Many people, however, find it difficult to read the network diagrams produced when the positions of nodes are not constrained. One reason for this is that spatial relationships like above and below or right and left on the diagrams are meaningless when no coordinate system is evident.

Most layout techniques for directed graphs, on the other hand, use the feature of directed edges; examples are upward/downward drawings [1] and hierarchical drawing [13]. These techniques orient the directions of edges to one common direction. These kinds of layout do not provide coordinate systems in the strict sense, but the common direction gives us a kind of viewpoint for the global structure of the graph. We can grasp the overall structure of the networks and the position of each element in the global structure.

2.2 Implementation of Early Anchored Maps

We first modified the spring embedder to introduce proxies of coordinate systems into undirected graphs by fixing three nodes (anchors) chosen by the observers on the three vertices of a regular triangle [9]. In the spring embedder model, any nodes can be fixed at any positions by simply ignoring the spring and the repelling force for those nodes.

The anchors adjacent to an unfixed (free) node pull it in different directions. By the nature of the spring embedder, the free node will move to a position that expresses its relation to the adjacent anchors. When we consider the anchors viewpoints, we can see the positions of free nodes

Manuscript received March 31, 2008.
Manuscript revised July 22, 2008.

[†]The author is with the Department of Computer Science, University of Tsukuba, Tsukuba-shi, 305–8573 Japan.
a) E-mail: misue@cs.tsukuba.ac.jp

DOI: 10.1093/ietisy/e91–d.11.2599

as expressions of the relations between those nodes and the viewpoints. The anchors of an anchored map thus give viewpoints to readers of the network diagram.

2.3 Problems in Drawing Anchored Maps

The reason that three anchors were used in the early anchored maps was to keep anchors in symmetry. When the number of anchors is three, the network diagrams are independent of the arrangement of the anchors. The diagrams with any anchor arrangements are essentially the same, which means that we do not need to worry about the arrangement of the anchors.

The number of nodes to be viewpoints is not always three, however. When there are four or more anchors, the arrangements of the anchors are not symmetrical. So we have to consider the arrangement of the anchors.

By the way, anchors might have orders by their meanings. In such case, the anchors can be arranged in the orders, even when four or more anchors are used. For example, when we choose the days of the week as anchors, we may arrange the anchors in the order of "Sunday," "Monday," . . ., "Saturday."

3. Formalization of Anchored Maps

When the number of anchors is four or more and the anchors do not have proper orders, the orders of the anchors must be determined appropriately. Because this problem cannot be solved by simple extension of the spring embedder, We reorganized the method for drawing anchored maps.

3.1 Drawing Objects

The drawing objects are bipartite graphs. A bipartite graph is a graph whose the set of nodes can be divided into two disjoint sets such that no edge has both endpoints in the same set.

A bipartite graph is formally described as $G = (A \cup B, E)$. Here, A and B are finite sets of nodes, and $A \cap B = \emptyset$. E is a finite set of edges, and $E \subseteq A \times B$.

Although this paper simplifies the drawing of anchored maps by restricting the drawing objects to bipartite graphs, so we need not consider edges between anchors, the concept of the anchored map can be applied to general graphs.

3.2 Conventions

Node arrangement

- Compound coordinate system: for a bipartite graph $G = (A \cup B, E)$, the elements of set A depend on a circular coordinate system; they are arranged on the circumference at equal intervals, while the elements of set B are independent of the coordinate system.

Edge routing

- Straight-line wiring: adjacent nodes are connected by straight line segments.
- Edges are independent of the coordinate system.

The elements of set A are called "anchors" and the elements of set B are called "free nodes."

3.3 Aesthetic Criteria

The following rules are used for drawing an anchored map.

R1 Nodes are separated mutually more than the lowest distance.
R2 Adjacent nodes are laid out as closely as possible (minimize the total length of edges).
R3 The number of edge crossings is as small as possible.
R4 Anchors adjacent to common free nodes are laid out as closely as possible.
R5 Free nodes adjacent to common anchors are laid out as closely as possible.

The drawing rules R1, R2, and R3 are used in many other graph drawing methods, but the rules R4 and R5 are specific to the anchored maps and are described formally in 3.3.2 and 3.3.3.

3.3.1 Preparation for Formalization

Suppose that M is the number of anchors; that is, $M = |A|$. The anchors are arranged on the vertices of a regular M-gon (a polygon with M vertices). The vertices of the M-gon are labeled clockwise from 1 to M. Which vertex is chosen to be 1 does not matter.

Suppose that $p(a)$ is the position of anchor a and that $A(b)$ is a set of anchors adjacent to free node b; that is, $A(b) = \{a \in A | (a, b) \in E\}$.

3.3.2 (R4) Closeness of Anchors Adjacent to Common Free Nodes

Rule R4 is expressed by

$$d(B) = \sum_{b \in B} d(b), \tag{1}$$

where $d(b)$ is the closeness of the anchors adjacent to free node b. It is defined formally after the explanations of "clockwise distance" and "gap."

Clockwise distance

The **clockwise distance** l_M between the i-th vertex and the j-th vertex is the number of vertices we meet when we trace the vertices of the M-gon from the i-th to the j-th clockwise. It is given by

$$l_M(i, j) = (j - i + M) \bmod M. \tag{2}$$

Gap

Suppose that b is a free node and $A(b) = \{a_1, a_2, \ldots, a_k\}$ and that $a_1, a_2, \ldots, a_k$ have been sorted by the numbers of vertices of the M-gon. In other words, $p(a_s) < p(a_t)$ if $s < t$. The gap for each anchor is the clockwise distance between one node and the next node (the next after a_k is a_1). The **gap** $g(a_s)$ of node a_s is given by

$$g(a_s) = l_M(p(a_s), p(a_{(s+1) \bmod M})). \tag{3}$$

Closeness of anchors

When k anchors are adjacent to a certain free node b, a sequence of k gaps is obtained. Removing the maximum gap from the sequence leaves $k-1$ smaller gaps. The closeness of anchors is defined by the sum of powers of $k-1$ gaps and represented by $d(b)$. Suppose that b is a free node and $g_1, g_2, \ldots, g_{k-1}$ is the remainder sequence that excludes the maximum gap from the gap sequence of the anchors adjacent to b. The closeness of the anchors adjacent to free node b is represented by $d(b)$ and is given by

$$d(b) = \sum_{i=1}^{k-1} g_i^q, \tag{4}$$

where the parameter q is a positive number [10].

A small $d(b)$ indicates that the anchors adjacent to the same free node b are close to each other.

3.3.3 (R5) Closeness of Free Nodes Adjacent to Common Anchors

It is an intuitive and simple rule to make free nodes that are adjacent to the same anchors close to each other. When there are two free node b_1 and b_2 adjacent to the same anchor set, the two free nodes should be arranged close to each other. However, if there is another free node b_3 adjacent to a similar anchor set, we may want to place b_3 a little closer to b_1 and b_2 than the other free nodes that are adjacent to completely different anchors. The similarity between two free nodes is therefore defined by using Jaccard index. The similarity of free nodes b_1 and b_2 is represented by $s(b_1, b_2)$ and is given by

$$s(b_1, b_2) = \frac{|A(b_1) \cap A(b_2)|}{|A(b_1) \cup A(b_2)|}. \tag{5}$$

Rule R5 can be formalized so as to increase the strength of the negative correlation between similarities and the Euclidean distance between the free nodes.

4. Drawing of Anchored Maps

4.1 Procedure

An anchored map is laid out in two steps:

(Step 1) Arrange anchors on the circumference at equal intervals. The size (i.e., radius) of the circumference is decided according to the size of the drawing area (i.e., window), and the order of the anchors on the circumference is decided.

(Step 2) Fix the anchors and arrange the free nodes at positions appropriately expressing their relationships to the anchors by the spring embedder.

The size of the circumference influences the size of the drawing but does not influence the quality of the layout. The order of the anchors, on the other hand, has a large effect on the quality of the layout. We do not need to worry about routing of edges because the edges are drawn as straight line segments.

4.2 Problem of Deciding Anchor Order

The distance between adjacent nodes (length of the edges) and the number of edge crossings are influenced but not determined by the order of the anchors. The spring embedder in Step 2 and the initial positions of the free nodes also influence the distance, but the influence of the spring embedder is negligible in comparison with that of the anchor order.

Deciding the order of the anchors is therefore the most critical problem. The goodness of a certain order can be evaluated only after the spring embedder has been processed for a certain anchor arrangement, but to do this for all the possible candidate orders would require too much computing time. We thus need an alternative index that is computable in a deterministic way and with a low computing cost.

4.3 Penalies: Indices to Decide Anchor Order

The work presented in this paper used numeric indices (penalties) as substitutes for the aesthetic criteria and evaluated the effectiveness of the indices experimentally.

The penalties P2, P3, and P4 described here respectively correspond to rules R2, R3, and R4. Penalty P5 is based on another idea (i.e., is irrelevant to R5).

P2 *Total length of edges* when all free nodes are placed at the barycenters of anchors adjacent to the free nodes (R2).

P3 *The number of edge crossings* when all free nodes are placed at the barycenters of anchors adjacent to the free nodes (R3).

P4 *The distance along the circumference* between anchors adjacent to the same free nodes (R4).

P5 *The eccentricity of free nodes*: the sum of distances between every free node and the center of the circumference.

The drawing rules R2 and R3 as they are cannot be used as indices because both can be evaluated only after the positions of free nodes have been decided. So in the definitions of penalties P2 and P3 we instead use the values (edge

```
p0 := current penalty;
d := ⌊|A|/2⌋;
while (d > 0) begin
      repeat
            c := false;
            for i := 0 to |A| − 1 begin
                  j = (i + d) mod |A|;
                  swap i-th node and j-th node;
                  p1 := current penalty;
                  if (p1 < p0) begin
                        p0 := p1;
                        c := true;
                  end else begin
                        swap i-th node and j-th node;
                  end
            end
      until (not c);
      d := ⌊d/2⌋;
end
```

Fig. 1 Algorithm searching for a good anchor order (the vertices run from 0 to $|A| - 1$ in this algorithm).

lengths and number of crossings) calculated assuming that all free nodes are placed at the barycenters of anchors.

4.4 Searching for a Good Order

A simple way to find the anchor order that minimizes the penalty is to compute the penalties for all the possible orders (isomorphism cause by rotation and mirroring is negligible) and then use the order with the minimum penalty. This procedure gives the optimal solution for the penalty but its computational cost is very large.

The graph-drawing method presented in this paper therefore uses a quasi-optimal solution for the penalty because the procedure for finding it has a lower computational cost [10]. The procedure begins with a random anchor order, exchanges two anchors, and compares the penalties before and after the exchange. It continues to choose pairs of anchors and it exchanges them if the penalty decreases by doing so. The procedure is listed as an algorithm in Fig. 1.

5. Evaluation of the Aesthetic Criteria

To evaluate the technique that we described in the previous section for aesthetic criteria, we implemented the technique in Java and performed an evaluation experiment.

5.1 Purpose of the Evaluation

The purpose of the evaluation was to answer the following questions: (1) Do the four penalties effectively evaluate their respective aesthetic criteria? (2) Which of the four penalties is the most useful?

5.2 Generation of Random Graphs

Random bipartite graphs for the experiment were generated assuming the following pairs of numbers of anchors and free nodes: (5, 50), (7, 70), and (10, 100). For each pair, five bipartite graphs were generated for each of the edge-occurrence probabilities 0.1, 0.2, and 0.3 (a total of 45 bipartite graphs). The sizes of these graphs are listed in Table 1, where $|A|$ is the number of anchors, $|B|$ the number of free nodes, and $|E|$ the number of edges. The number of free nodes varies because those with a degree less than 2 were deleted.

Table 1 Size of the graphs used for the evaluation.

$\|A\|$	$\|B\|$	$\|E\|$
5	4 ~ 17	8 ~ 46
7	5 ~ 38	10 ~ 112
10	15 ~ 80	33 ~ 289

5.3 Outline of the Evaluation

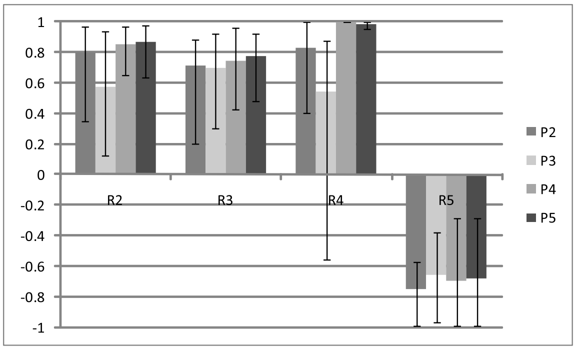
All the possible orders of anchors were generated for each of the 45 graphs. The number of possible orders is 12 when $|A| = 5$, 360 when $|A| = 7$, and 181,440 when $|A| = 10$. For every possible order, the penalty values were calculated and the layout of free nodes was computed using a spring embedder. The aesthetic criteria evaluation values were also calculated for each layout, and their relations with the penalty values were examined.

5.4 Evaluation Results

Figure 2 shows experimental data. We obtained all detailed data for all 45 graphs but show only the compiled data because of space limitations. Figure 2 (a) shows the averages of the results for the graphs with 5 anchors. The bar chart shows the effects of the four penalties P2, P3, P4, and P5 for each of the rules R2, R3, R4, and R5. For the rules R2 (total length of edges), R3 (the number of edge crossings) and R4 ($d(B)$), the bars show the correlation coefficients calculated for the 15 graphs. The bar for P2 in R2 group, for example, shows the mean of the correlation coefficients between P2 and R2 for all 360 anchor orders of combination. The main bar shows the average for 15 graphs, and the error bars show the minimum and maximum values among 15 graphs. For rule R5, the bars show the average, for 15 graphs, of the values of R5 (the correlation coefficient between the similarities $s(b_i, b_j)$ and Euclidian distances between nodes b_i and b_j for all pair of the free nodes b_i and b_j) in the layouts with the optimum penalty P2, P3, P4 or P5.

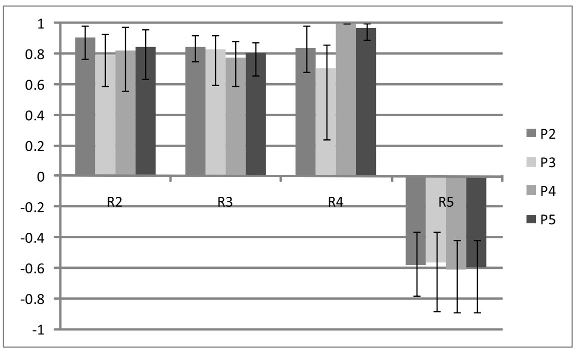
Strong positive correlation is expected for R2, R3, and R4, and strong negative correlation is expected for R5. Penalties P4 and P5 have a good effect generally. It is trivial that penalty P4 coming from anchor distance is effective for R4, but it is new knowledge that P5 which does not come from an aesthetic criterion has a similar effect.

Figure 2 (b) shows the results for the graphs with 7 anchors, and Fig. 2 (c) shows the results for the graphs with 10 anchors. They show almost the same tendency.



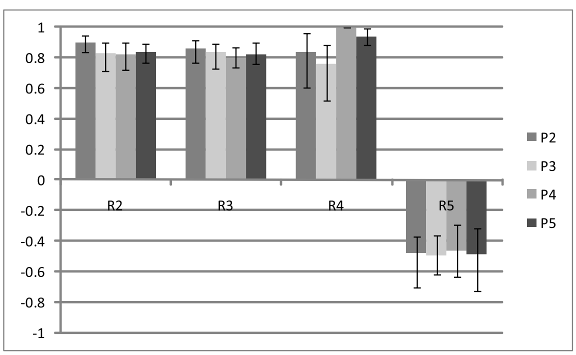
(a)

Results for graphs with $|A| = 5$.



(b)

Results for graphs with $|A| = 7$.



(c)

Results for graphs with $|A| = 10$.

Fig. 2 Evaluation results.

5.5 Time Complexities of Penalty Functions

The computational complexities of the penalties are listed in Table 2. All penalty values were computed for the given positions of the anchors. The positions of the anchors are easily computed by using size of the circumference to arrange the anchors if the order of the anchors is decided. If the order of the anchors is decided, the positions of the anchors can be easily computed by using size of the circumference. Once the order of anchors is decided, the computational complexity of computing the positions of the anchors is $O(|A|)$.

When we think the way anchored maps are ordinarily used, we would expect $|B|$ to usually be less than $|E|$. We would therefore expect the computational complexities of P2, P4, and P5 to all be $O(|E|)$. We incidentally remark that for various graphs there were no clear differences between the computation times needed to calculate these penalties.

Table 2 Time complexities of penalty functions.

Penalty	Complexity
P2	$O(\|E\|) + O(\|B\|)$
P3	$O(\|E\|^2) + O(\|B\|)$
P4	$O(\|E\|)$
P5	$O(\|E\|) + O(\|B\|)$

5.6 Summary of the Evaluation

All penalties have some effect for all aesthetic criteria, and only penalty P3 has an effect considerably different from the effects of the other penalties. Although the effects of all of the penalties were usually similar with regard to the quality of the layout of graphs, sometimes they were not. Something about a penalty seems to affect the quality of the layout. The study of such effects is a future project.

6. Overview of Network Information

The effectiveness of the anchored maps is shown here by two examples of overviews of network information extracted from actual databases.

6.1 Overview of Sales Database

A sales database was to examine the relations between items and the times they were sold.

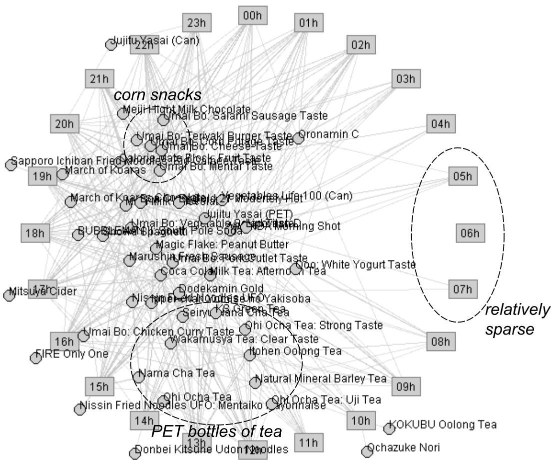
6.1.1 Data Source

We could not get any databases of actual shops, so we used a sales database managed in our laboratory. While not the database of a real shop, the transactions whose records are in it are real. We started to use the database in June 2006 and recorded the data for 4513 purchases by the end of May 2007.

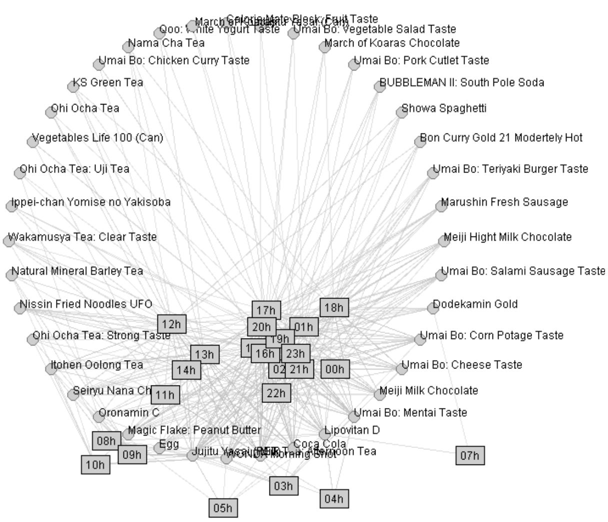
One transaction is represented as one record, and one record has the purchase date and time, the customer name, the item name, the item category, and the price. Various relations can be extracted from the database. In this paper, we focus on relations between items and time zones (every hour; 00 h ∼ 23 h) when the items were sold. We picked up top 48 items at sales amount ranking and then removed items which were sold less than two times in any time zones to avoid noise.

6.1.2 Overviews Obtained Using Anchored Maps

Figure 3 (a) shows networks representing relations between items and the times they were bought. The 24 hours of the day zones are placed as anchors. Because time zone has a cyclic order, so the hours placed in their proper order.



(a) 24 hourly intervals placed as anchors (contours and their labels added manually)



(b) items placed as anchors

Fig. 3 Anchored maps showing relations between items and time zones.

We can see at once that the activity of the students is relatively low in the morning. More careful inspection reveals that some flavors of corn snack were bought intensively at night (mainly around 21:00) and the PET bottles of tea were bought intensively in the early afternoon; that is, at lunch time.

In Fig. 3 (b), items are placed as anchors and the items with degree 1 have been omitted for the sake of the readability. Since items have no proper orders, we use the technique described in Sect. 4 to decide the order of anchors. We see the following clusters in the diagram.

1. Early morning 1: 03 h, 04 h, and 05 h
2. Early morning 2: 06 h (no node = no activities)
3. Morning 1: 07 h
4. Morning 2: 08 h, 09 h, and 10 h
5. Lunch time: 11 h to 14 h
6. Evening – night: 15 h to 02 h

6.2 Overview of Paper——Author Networks

Coauthorship networks are treated as one of the subjects of Social Network Analysis [6], [11]. Here we observe bipartite graphs comprising relations between authors and articles. The bipartite graphs are fundamental structures of coauthorship networks.

6.2.1 Data Source

The data shown here was obtained from the DBLP[†]. Maintained by Michael Ley, it is a database server providing bibliographic information on major computer science journals and proceedings. On October 29, 2007 the server indexed about 950,000 articles.

Visualization of the whole of the DBLP is a challenging task, but now we focus only on detailed structures in it. The following example was extracted in a breadth-first search, starting from "Kazuo Misue," with depth 2.

6.2.2 Overviews Obtained Using Anchored Maps

Figure 4 (a) shows the graph drawn as an anchored map with the author nodes fixed as anchors. The order of the author nodes (anchors) has been decided so as to make author nodes that are adjacent to common paper nodes (free nodes) close to each other. If every paper were written by authors in a local community, the coauthors of a paper would be placed close to each other and the paper would be close to the coauthors; that is, the circumference. Some paper nodes, however, are placed far from the circumference. This means that they may have been written by authors from different communities.

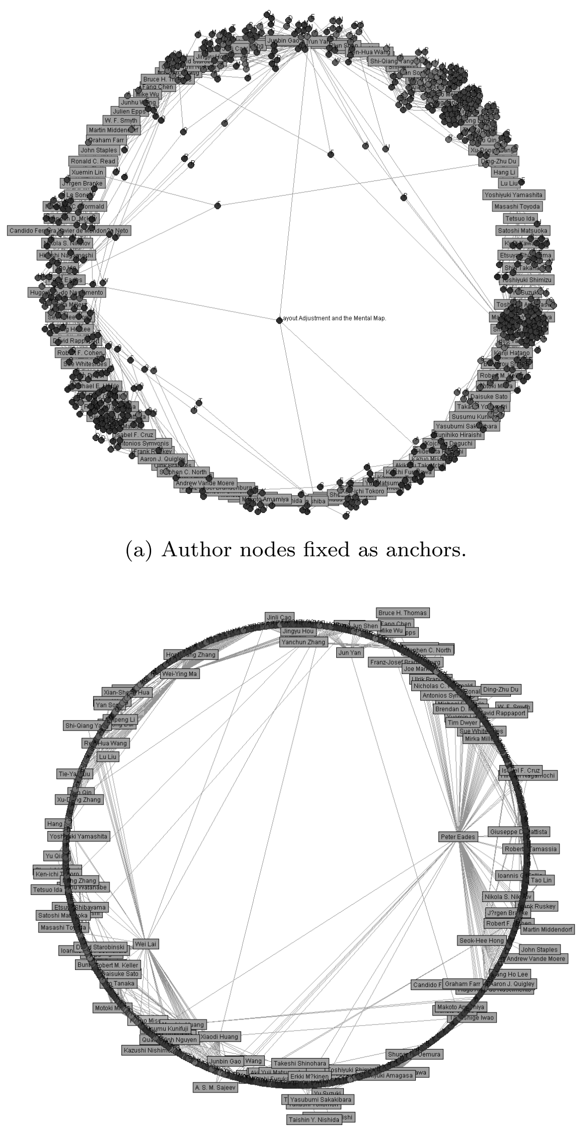
For instance, the paper placed just below the center of Fig. 4 (a) was written by four coauthors[††] when two of them worked at a private company and the other two were at a university. Later they moved to different universities and continued research activities. From the viewpoint of their current positions, this paper is crossing communities. An anchored map helps us discover such papers.

Figure 4 (b) is the same network shown in Fig. 4 (a) but with the paper nodes fixed as anchors. The order of the paper nodes (anchors) has been decided in order to make paper nodes that are adjacent to common author nodes (free nodes) close to each other. Nodes of authors who wrote a lot of papers with various coauthors are toward the center of the circle.

For example, we see such one node in the central right side of the figure and another in the lower left. Furthermore, we can assume that the two authors do not have many joint papers because the two nodes are placed apart from each other.

[†]http://dblp.uni-trier.de

[††]K. Misue, P. Eades, W. Lai and K. Sugiyama: Layout Adjustment and the Mental Map, Journal of Visual Languages and Computing, vol.6, no.2, pp.183–210, 1995.



(a) Author nodes fixed as anchors.

(b) Paper nodes fixed as anchors.

Fig. 4 Anchored maps showing relations between authors and papers.

6.3 Discussion

The anchored map seems an effective tool for gaining insight into network information because it can reveal clusters, like those seen in Fig. 3. While other clustering techniques might also reveal such clusters, the advantage of the anchored map is that we can see grounds and the background that compose the clusters as some connecting lines between nodes. By using such the diagrams we can first recognize the existence of the clusters and then we understand the reasons for the clusters by observing connecting lines between nodes.

The anchored map is also effective for finding distinctive nodes as illustrated in Sect. 6.2. It can help us discover important features that we cannot find with feature quantities such as the degree of nodes. It is possible to obtain an overview of graph structures in a free layout, but it is not always easy to find distinctive structures. Actually, it is difficult to find the characteristic paper nodes and author nodes mentioned above only from a free layout of the network.

The anchored map restricts the placement of anchors to a circumference. This may cause some confusing situations. We might expect free nodes adjacent to most anchors to be in the central part of the figure, as in Fig. 3 where items whose sales do not depend on specific times swarms are in the central part of the figure, but the converse is not always true. Some items located near the central part might depend only on certain times, such as items bought only at noon and midnight. Searching for good anchor order avoids such confusing situations but is not perfect.

7. Related Work

Representation techniques similar to "anchored maps" have been used in some other tools, such as Visual Who [3] and SQWID [8]. Visual Who is a tool whose purpose is to visualize communities. The tool visually expresses the appearance of the communities by using statistical information extracted from the text data of mailing lists. The positions of the nodes are computed by the spring model. The user can arrange an arbitrary mailing list as anchors, and the layout of the nodes represent member changes. This tool is interactive, and the user can arrange the anchors manually. It does not provide automatic layout facilities for the anchors.

SQWID is a Web search tool that expresses WWW retrieval results by using anchored maps. Terms used in the query are placed on the vertices of a triangle as anchors. Web pages (or sites) are placed according to the level of their relationship to those terms. The number of anchors is limited to 3 so that the relationships between the arranged Web pages and the fixed terms do not become vague. One would think, however, that there are a lot of situations in which four or more anchors are needed and that an automatic technique to find the arrangement of anchors should be developed.

There are some research results related to techniques for drawing bipartite graphs [2], [5], [12], [14], but most of them put their focus on theoretical aspects such as planar layout and edge-crossing minimization. Application-oriented techniques for drawing bipartite graphs have not been studied much.

8. Concluding Remarks

This paper described the readability problem due to the lack of coordinate systems in topological diagrams and proposed that a drawing style called "anchored maps" would improve readability. On anchored maps, some nodes called "anchors" are fixed as viewpoints and have an effect similar to that of coordinate systems. This paper described a method for drawing anchored maps with four or more anchors and also demonstrated the effectiveness of anchored maps by showing examples of overviews of network information extracted from actual databases.

Among the topics that need to be addressed in future work is the reconceptualization of network readability. The readability of networks or graphs is usually described as something determined by drawing conventions and drawing rules. The viewpoint introduction proposed in this paper might be higher-level concept that should be one of the aesthetic criteria for understandability.

References

[1] G. Di Battista and R. Tamassia, "Algorithms for plane representations of acyclic digraphs," Theor. Comput. Sci., vol.61, no.2-3, pp.175–198, 1998.

[2] E.D. Giacomo, L. Grilli, and G. Liotta, "Drawing bipartite graphs on two curves," Proc. Symposium on Graph Drawing (GD 2006), LNCS 4372, pp.380–385, 2007.

[3] J.S. Donath, "Visual who: Animating the affinities and activities of an electronic community," ACM Multimedia 95 – Electronic Proc., ACM Press, New York, 1995.

[4] P. Eades, "A heuristic for graph drawing," Congressus Numerantium, vol.42, pp.149–160, 1984.

[5] U. Fosmeier and M. Kaufmann, "Nice drawings for planar bipartite graphs," Proc. 3th Italian Conference on Algorithms and Complexity, pp.122–134, 1997.

[6] T.-H. Huang and M.L. Huang, "Analysis and visualization of co-authorship networks for understanding academic collaboration and knowledge domain of individual researchers," Proc IEEE Computer Graphics, Imaging and Visualisation, pp.18–23, 2006.

[7] M. Kaufmann and D. Wagner, eds., "Drawing graphs," Lecture Notes in Computer Science, vol.2025, p.312, Springer, 2001.

[8] S.D. McCrickard and C. Kehoe, "Visualizing search results using SQWID," Proc. Sixth International World Wide Web Conference (WWW6), 1997.

[9] K. Misue and I. Watanabe, "Visualization of keyword association for text mining," IPSJ SIG Technical Report 1999-FI-55-8, Information Processing Society of Japan, pp.65–72, 1999.

[10] K. Misue, "Drawing bipartite graphs as anchored maps," Proc. Asia-Pacific Symposium on Information Visualization (APVIS2006), pp.169–177, 2006.

[11] M.E.J. Newman, "Who is the best connected scientist? A study of scientific coauthorship networks," Phys. Rev., E64, 2001.

[12] M. Newton, O. Sykora, and I. Vrto, "Two new heuristics for two-sided bipartite graph drawing," Proc. Symposium on Graph Drawing (GD 2002), LNCS 2528, pp.312–319, 2002.

[13] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for visual understanding of hierarchical systems," IEEE Trans. Syst. Man Cybern., vol.11, no.2, pp.109–125, 1981.

[14] L. Zheng, L. Song, and P. Eades, "Crossing minimisation problems of drawing bipartite graphs in two clusters," Proc. Asia-Pacific Symposium on Information Visualization (APVIS 2005), pp.33–38, 2005.

Kazuo Misue received a BSc and an MSc from Tokyo University of Science in 1984 and 1986, and he received a PhD in engineering from the University of Tokyo in 1997. He is an Associate Professor in the Department of Computer Science, University of Tsukuba, Japan. His research interests include human-computer interaction, information visualization, automatic graph drawing, and supporting human creative activities. He is a member of the ACM, IPSJ, JSAI, and JSSST.