ENERGY MODEL FOR DRAWING OF CLUSTERED GRAPH

Quoc Dinh TRUONG (*,**), **Taoufiq DKAKI** (*,***), **Pière-Jean CHARREL** (*,***)
truong@univ-tlse2.fr , dkaki@irit.fr , charrel@univ-tlse2.fr
(*) IRIT, Institut de Recherche en Informatique de Toulouse, 118 Route de Narbonne, 31062 Toulouse Cedex 9, France
(**) Cantho University, 1 Rue Ly Tu Trong, Cantho, Vietnam
(***) University Toulouse II, 5, Allées A. Machado 31058 Toulouse, France

Mots clefs :
Graphe, Graphe groupé, Visualisation de graphe
Keywords:
Graph, Clustered graph, Graph drawing
Palabras clave :
Gráfico, Gráfico arracimado, Dibujo del gráfico

Résumé

Dans cet article, nous présentons un modèle d'énergie pour le dessin de graphes dont les nœuds sont regroupés en plusieurs classes. La structure de tels du graphes est souvent utilisée dans plusieurs champs d'applications tel que les réseaux sociaux, les schémas conceptuels pour la construction de base de données, le génie logiciel … Peu de travaux de recherche abordent la problématique de la visualisation de tels graphes et les modèles traditionnels d'énergie ne satisfont pas les contraintes importantes que le dessin des graphes groupés sous-tend: Les sommets d'un groupe doivent être placés dans une zone convexe ; Toutes zones convexes doivent être mutuellement disjointes -sans surface d'intersection commune. La proposition d'un algorithme pour le dessin et la visualisation de telle structure de données semble donc importante Nous proposons un modèle d'énergie basé sur celui présenté dans [6] avec deux améliorations notables concernant les notions de distance optimale entre sommets et concernant l'influence des forces exercées sur les sommets. La méthode que nous proposons étend la notion de groupe qui est défini dans la majorité des travaux comme résultant de la structure du graphe comme un ensemble de sommets ayant beaucoup d'arcs internes et peu d'arcs externes. Dans notre approche, un groupe peut être définit indépendamment de la structure du graphe. Il peut être définit comme un ensemble de sommets induit par une quelconque information qui peut être exogène. Cette définition nous permet d'étendre le champ d'application de la visualisation des graphes groupés. L'avantage le plus important est qu'avec notre modèle, nous pouvons clairement visualiser en les distinguant les différents groupes même si ces groupes ont peu voire pas du tout d'arcs internes.

Abstract

In this paper, we introduce an energy model for clustered-graph drawing. Nowadays, clustered graph structure is often used in a lot of fields related to computer science such as social network analysis, database design, software engineering… This stresses the need for good algorithms that can successfully draw such graphs. These algorithms must capture and highlight the node clusters in a way that the resulting graph visualization conveys both inter-nodes and inter-cluster relationships. Most known force and energy model do not clearly fulfill these perquisites. We propose a drawing model based on [6] that redefines the optimal distance between vertices and the forces which are exerted on vertices. These changes are performed in order to satisfy the following two constraints: vertices of a same cluster must

be placed in a convex area and all convex areas related to the clusters must be mutually disjoint. In traditional energy methods the only clusters that are well drawn are the ones containing vertices with many internal edges and few outside edges. Our method can handle and successfully draw clusters where nodes have few or even no connecting edges, This enhances the potential of graph visualization, as an exploratory analysis mean, by allowing the understanding of both node and cluster relationships even in the case of graphs with sparse clusters.

Introduction

Automatic graph drawing has been a dynamic research area over the last decades. It has been boosted by the increasing interest for graph data structure and for visualization as an important part of exploratory data analysis. Moreover graph visualization has proven it efficiency in many applications fields, such as social network, software engineering, electronic circuit design and database design. These fields handle data that can be seen as sets of objects sharing some relationships which can be effortlessly and efficiently be modelized by graphs. There are many strategies which can be used to draw general graphs such as: circular drawing [4, 14], planar drawing [10] and straight-line drawing. One of the most popular methods for straight-line drawing of graphs is force directed method [6, 9]. Most used algorithms in this context are rather simple and give satisfactory drawing results for average graph size of 100 nodes. There are also multi-scale methods that demonstrate their ability to handle larger graphs [7, 8, 11].

When the amount of information to be visualized grows to be big or the visualization constraints become more complex, the basic graph structure turns out to be inappropriate. This is, for example, the case of electronic circuit design where circuit boards usually comprise a large number of components which have to fit geographic constraints. In this case, a more suitable graph structure -clustered graph- is needed. This more sophisticated data structure is also appropriate for tackling the complex problem that can occurs in many domains related to computer sciences such as social networks analysis. There is, therefore, a need for methods that can visualize clustered graph without hiding the cluster structure within the data. These methods mainly deal with the problem -that traditional methods often fail to deal with- of distinctly presenting the clusters of a given graph..

In this paper, we introduce an energy based model working on clustered graph drawing. The model we propose redefines the constraints upon which energy models rely. It manly changes the optimal distance between vertices and the force exerted on vertices. This is done so that the drawing satisfies the following two criteria: vertices of a same cluster must be placed in a convex area and all convex areas related to each cluster must be mutually disjoint. The main advantage compared to other related works is that our method can successfully separate, when they exist, each cluster event if clusters density is small while at the same time it draws each cluster independently following the basics of energy driven methods.

The rest of paper is organized as follow. In section 1, we present the structure of clustered graphs, some related works and explain the motivation behind our proposal. In section 2, we describe our main contribution: an energy model for clustered graph drawing. This section mainly describes and explains the new constraints that are applied to clustered graph. Section 3 gives some examples. Several open problems are discussed in last section.

1 Background

1.1 Clustered graph structure

A graph G is a couple (V, E), where V is a finite set of vertices and E a finite set of edges with $E \subseteq V \times V$. A clustered graph CG is a triplet G(V, E, P), where V is a finite set of vertices, E a finite set of edges with $E \subseteq V \times V$ and P is a partition over V. The Number of elements in P corresponds to the number of clusters in G. The following figure illustrates structure of clustered graphs.

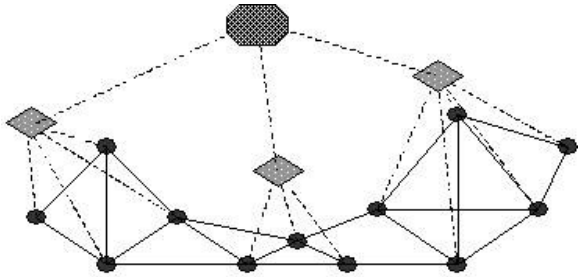


Figure 1 Structure of clustered graph associated to a cluster dendrogram

1.2 Related works

In the last decades, there were enormous works that has been proposed in the field of graph drawing in order to generate intelligible graph layouts. These works consist of placing the vertices of a graph in a 2D or 3D space. Each vertex is considered as a single and "autonomous" object. Some representative approaches are depicted in [3, 6]. These approaches are effectively appropriate for simple graph structure.

When considering the case of clustered graphs, few works can be cited. Among these works we can mention the ones in [1, 5, 12]. These works are related to different backgrounds and are relevant to different specific domains but they all address the problem that the well-known energy models [6, 9] and their extensions do not deal with and that consists in clearly highlighting the groups that compose a clustered graph especially for graphs with small diameter. [5] presents a method for straight-line drawing of clustered graph by applying Tuttle's algorithm [16] on a triconnected-skeleton clustered graph. Noack [12] set up an approach that clearly displays different clusters of graph by using a new energy model he called LinLog which use a cut ratio as a measure for the coupling of two disjoint sets of vertices. Noack proves that a graph having the minimum energy according to LinLog has its clusters finely displayed. Each cluster is separated from the remaining graph vertices. The distance of each cluster from each of the remaining graph vertices is inversely proportional to the coupling.

Unlike the above approaches, [1] propose a method based on "Level of detail visualization" for the drawing of clustered graph where clusters are created in a recursive way –e.g ascendant/descendant clustering. The obtained clustered graph forms a hierarchy. The resulting approach allow the changing of views from very abstracted to very detailed. This approach only modifies the details of representation of the graph but does not alter its structure. Figure 2 gives an example of this approach

Figure 2 Example of Level of detail visualization of clustered graphs

What distinguishes our work from the works presented above is how clusters are defined. The authors of these works take clusters as sets of vertices with many internal edges and few outside edges: the clusters are graph relationship dependant. They are induced by endogenous information. The clusters are often dense and connected subgraph. In other words, the density of cluster is generally greater than this of the entire graph. This is a quite restricted definition of clusters since graph vertices can group together according to some exogenous information. For example, in a citation graph, the clusters may be built according to some geographic similarity between graph vertices which has nothing to do with the citation relationship. The previously cited approaches do not apply well to this kind of clustered graphs. Our motivation is then to propose a method for the drawing of clustered graph where clusters are "freely" defined. We consider that a cluster can be any set of vertices. So, in this context, a cluster has not to be a dense subgraph. It is even possible to consider clusters containing only pairwise disconnected nodes –clusters without internal edge.

2 Contribution

2.1 Drawing rules

In this subsection, we present the set of heuristic rules that abstracts our approach for clustered graphs drawing. These rules are listed in priority order.

- Each cluster must be positioned in a convex zone.
- The drawing zones associated to the clusters must not overlap.
- Internal layout of each cluster is optimized first (edge crossings are reduced as much as possible within a same cluster)
- Overlapping of vertices is reduced as much as possible.

2.2 Model

Our method is based on force directed placement method proposed in [6]. Like traditional methods, our method iterates two steps: computations of the resulting force of attraction and repulsion among vertices; the positioning adjustment of each vertex and the computation of their new coordinates following the temperature model. We borrow the notion of temperature used in [6]. Using the temperature and "cooling" function, the displacement of a vertex is majored with the maximum displacement value decreasing over time. This translates the idea that, as the layout becomes better the amount of adjustment becomes smaller.
Before presenting our algorithm, let us introduce the notions and functions we use.

2.2.1 Optimal distance between vertices

To compute attractive and repulsive forces, it is necessary to define an optimal distance between graph vertices. In our algorithm, we distinguish two types of distance: one between vertices of different clusters –optDist- and relevant to vertices of a same cluster -optDistCluster-. We believe that the optimal distance between vertices of a same cluster must be less than the global optimal distance because the size of a cluster drawing area is smaller than the size of the total drawing area (optDistCluster = a * optDist where $a < 1$)
The idea behind using two different optimal distances is not only to keep all the vertices of a same cluster within a same convex zone but also to preserve a satisfactory display inside the area related to this cluster In the case of clustered graph drawing, we consider that the layout of a cluster is "satisfactory" if it resembles the layout of that cluster when it is considered as a single graph. Conducted experimentations s how that the optimal proportion between the two distances is 2: optDist = 2*optDistCluster.

2.2.2 Used functions

- proportion (Ci): computes the proportion between the number of possible edges of cluster Ci and the number of actual edges in cluster Ci (only edges that connect between vertices of cluster Ci is considered). Proportion(Ci) is the inverse of density of i^{th} cluster .

```
proportion(Ci)
begin
        {number of vertices of cluster Ci}
        nv := numberVertices0f( Ci);
        {number of actual edges of cluster Ci}
        ne := numberEdgesOf(Ci);
        return ((nv - 1) * nv) / ne;
end.
```

- calculForceRepul (dist): returns the nodes repulsive force

```
calculForceRepul(dist)
begin
        return (optDist * optDist) / dist;
end.
```

- calculForceAttract (dist) returns the nodes attractive force

```
calculForceAttract(dist)
begin
        return (dist * dist) / optDist;
end.
```

- calculForceRepulClsuter (dist): returns the clusters repulsive force

```
calculForceRepulCluster(dist)
begin
        return (optDistCluster * optDistCluster)/dist;
end.
```

- calculForceAttractCluster (dist) returns the clusters attractive force

```
calculForceAttractCluster(dist)
begin
        return (dist * dist) / optDistcluster;
end.
```

2.2.3 Algorithm

Our algorithm has a clustered graph G = (V, E, P) as input. We use array C to represent the cluster id of each vertex. Each vertex is associated to two vectors: .pos and .disp which represent respectively the position and the displacement of each vertex. The detail of algorithm is summarized by the following pseudocode

```
for i := 1 to iterations do begin
        for v in V do begin
                v.disp := 0;
                for u in V do
                        if (u ≠ v) then begin
```

```
                                        {D  is short hand for the difference vector between the positions of the two vertices}
                                        D  := v.pos - u.pos;
                                        If (C[u] ≠ C[v]) then begin
                                                {calculate the effect of repulsive force on every two vertices of different clusters}
                                                u.disp := u.disp - ( D /| D |) * calculForceRepul (| D |);
                                                v.disp := v.disp + ( D /| D |) * calculForceRepul (| D |);
                                        end
                                        else begin
                                                {calculate the effect of forces on every two vertices of same cluster}
                                                if ((v,u) ∉ V) then begin
                                                        {add the additional attractive forces to keep vertices close together}
                                                        u.disp := u.disp + ( D /| D |) * calculForceAttractCluster (| D |);
                                                        v.disp := v.disp - ( D /| D |) * calculForceAttractCluster (| D |);
                                                end
                                                {calculate the effect of repulsive force on every two vertices in cluster context}
                                                u.disp := u.disp - ( D /| D |) * calculForceRepulCluster (| D |);
                                                v.disp := v.disp + ( D /| D |) * calculForceRepulCluster (| D |);
                                        end
                                end
                        end
                end
        for e in E do begin
                {each edge is an ordered pair of vertices .u and .v}
                D  := e.u.pos – e.v.pos
                if (C[v] ≠ C[u]) then begin
                        {calculate the effect of attractive force in graph context}
                        e.v.disp := e.v.disp – ( D/| D |) * calculForceAttract (| D |);
                        e.u.disp := e.u.disp + ( D /| D |) * calculForceAttract (| D |);
                end
                else begin
                        {calculate the effect of attractive force in cluster context}
                        pro := proportion(C[u]);
                        e.v.disp := e.v.disp – ( D/| D |) * calculForceAttractCluster (| D |) * pro^2;
                        e.u.disp := e.u.disp + ( D /| D |) * calculForceAttractCluster (| D |) * pro^2;
                end
        end
        {limit the maximum displacement to the temperature t and then prevent from being displaced outside frame}
        for v in V do begin
                if (|v.disp| > temperature) then
                        v.pos =  v.pos + v.disp/(|v.disp| / t);
                else
```

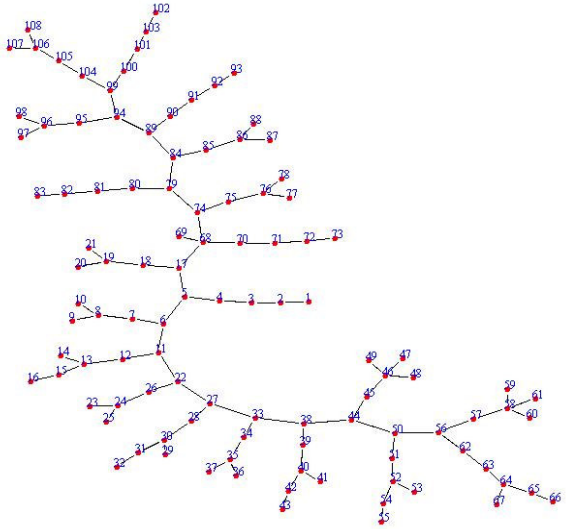
```
                v.pos = v.pos + v.disp;
        end
        {reduce the temperature as the layout approaches a better configuration}
        t := cool(t)
end
```

In comparison to other energy model approaches, our method, considers two kinds of optimal distances to distinguish between the forces implying two nodes of the same cluster -intra-clusters forces- and those implying two nodes of two distinct clusters -inter-clusters forces. Those forces apply differently to the nodes according to the clusters they belong to. Also, when considering involved forces between two disconnected vertices, traditional models only take into account repulsive forces. In our model, we also take into account the attractive forces a node may exercise over disconnected vertices of the same cluster –because they belong to the same group. In other words, we add the virtual edges -which do not appear in representation- to keep vertices of same cluster close together. Second, when considering the forces produced between vertices connected by an edge, in the case of vertices of same cluster, we increase the strength of attractive forces so that the attractive forces produced by virtual edges in first step do not over affect the general layout of cluster.

Another aspect of graph drawing algorithms is the initial configuration. This issue is still under work; in other words we do not propose any approach for finding best initial configurations; rather, vertices are placed randomly in the drawing space.

3 Results

As we mentioned above, the purpose of our work is to provide a new model to handle clustered graphs visualization. Our proposal is based on new definitions of both optimal distances and exerted forces. We argue that our goal of clustered graphs display enhancement somewhat achieved especially when considering sparse clusters. To assess this claim and evaluate the performances of our model we present several graphs and their respective representation following a traditional approach and following our approach. These examples show, even if they do not demonstrate, that our method performs quite well.



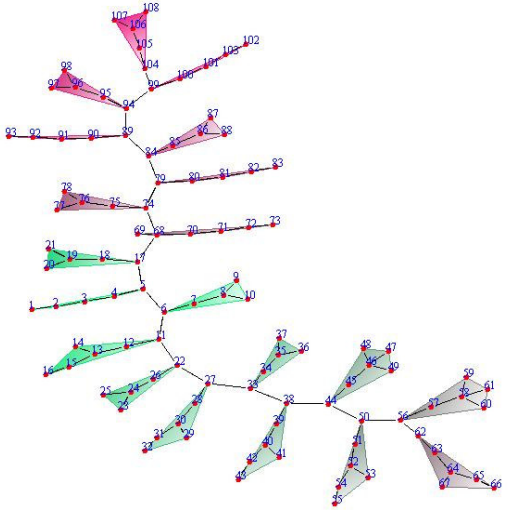
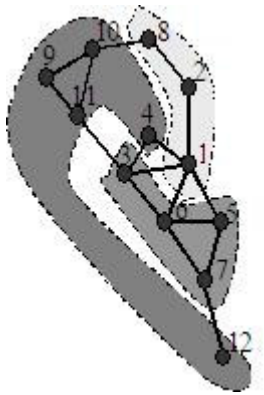
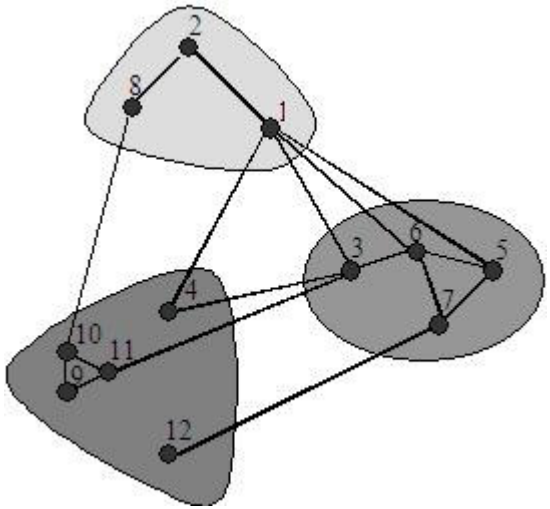


Figure 3 the example concerns a clustered graph where clusters are relatively dense. the figure at the left shows the result of a "normal" graph drawn using force directed method. At the right: the same clustered graph is drawn using our method. The performances of the two methods are equivalent

As we mentioned earlier, our model works also for clustered graphs where clusters may contain non connected vertices. This is what the following examples show.

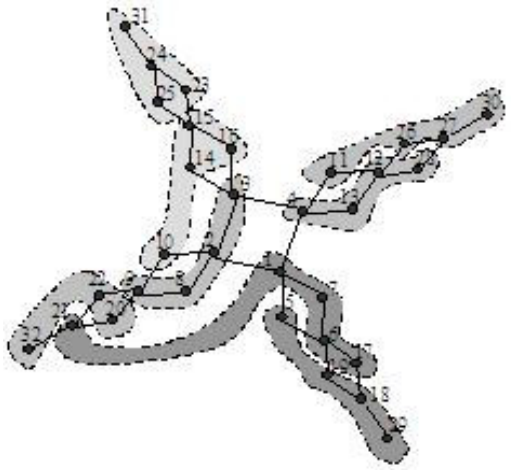




Cluster1: 1, 2, 8
Cluster2: 3, 5, 6, 7
Cluster3: 4, 9 10, 11, 12 (12 is a disconnected node inside the cluster)

Figure 4 this example shows that our methods outperforms the traditional approach when some clusters contain isolated nodes

The prerequisite associated to good clustered graph drawing are guaranteed when using our model even in the case of clusters having no internal edge at all. In this case the use of basic directed force placement is inappropriate as shown in the following example.



Cluster1: 23, 24, 25, 31
Cluster2: 10, 14, 15, 16 (not connected)
Cluster3: 20, 22, 32 (there is no internal edge in this cluster)
Cluster4: 2, 3, 8, 9
Cluster5: 4, 12, 13, 28
Cluster6: 11, 26, 27, 30 (not connected)
Cluster7: 1, 6, 7, 17, 21 (not connected)
Cluster8: 5, 18, 19, 29 (not connected)

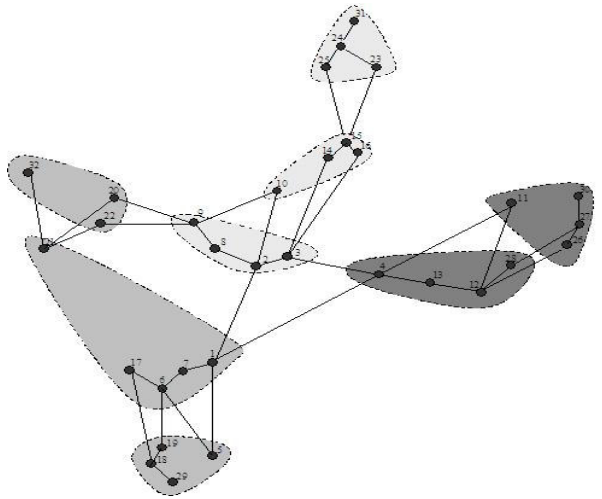


Figure 5 this example shows that our method also takes into account the case of clusters that have only disconnected nodes - having no internal edge at all

4 Conclusion and perspectives

In this paper, a new energy model for clustered graph drawing was introduced. The proposed method can handle any partitioned graph. In other word the user may choose any partition of graph vertices independently from graph proper relationship characteristics. This expands the application field of clustered graph visualization as optimal cluster-oriented visualization is achieved whatever the properties of the graph and the clusters are. Our model extends traditional force and energy driven models such as the one portrayed in [6] and outperformed models specialized in clustered graph drawing [5, 12]. Our main contribution is to efficiently separate the clusters of graph even those with low density or those having disconnected nodes.

In this paper, we presented an energy model for clustered graph drawing in two-dimensional space. We are presently considering the extension of this work to the three-dimensional (3D) clustered graph drawing.

The work is also being prolonged -as enhancement of one of our previous attempt to tackle the problem of geographic graph visualization [15]- to suit the problem of clustered graph drawing where the drawing area of each cluster is predetermined. This is mainly the case of graphs where the clusters are related to some geographic constraints –e.g. when graphs vertices are grouped according to some of their geographic properties such as the authors' countries or locations in science citation networks. Such graph drawing approach will help to understand the implications –if any- of geographic characteristics over the vertices relationships. Nevertheless, it poses challenging problems such as small drawing areas that contain a lot of vertices while huge areas may contain few.

We are also considering the reshaping our model in order to fit the requirements of intersecting clustered graph drawing [13]. Intersecting clustered graphs are clustered graphs with intersections among clusters. This is, for example, the case of conceptual graphs as in ontology representation [2] .The energy model must deal with the problem of forming intersection areas between clusters and placing shared vertices within these areas. We believe that this reshaping of our model can be achieved within reasonable delay.

5 References

[1] BALZER M., DEUSSEN O., *Level-of-detail visualization of clustered graph layouts. 6th International Asia-Pasific Symposium on Visualization*, APVIS07, pp. 133-140, Sydney Austrailia, 2007.

[2] Barthelemy, M., E. Chow, and T. Eliassi-Rad, *Knowledge Representation Issues in Semantic Graphs for Relationship Detection*, Papers from the 2005 AAAI Spring Symposium on AI Technologies for Homeland Security, Stanford, CA, March 2005.

[3] BATTISTA G DI., EADES P., TAMASSIA R., TOLLIS I. G., *Graph Drawing: Algorithms for the Visualization of Graphs,* Prentice Hall, 1999.

[4] BAUR M., BRANDES U., *Crossing Reduction in Circular Layouts,* Proc. Graph-Theoretic Concepts in Computer-Science (WG '04), 332-343, 2004.

[5] FENG Q.W., EADES, P., COHEN, R.F., *Planar Drawings of Clustered Graphs,* Technical Report 05-95, Department of Computer Science, The University of Newcastle, Australia, 1995.

[6] FRUCHTERMAN T., REINGOLD B., *Graph drawing by force-directed placement,* Softw. Pract. Exp., 21(11):1129{1164, 1991.

[7] GAJER P., KOBOUROV S. G., *GRIP: Graph dRawing with Intelligent Placemen,*. In Marks, Joe, Eds. Proceedings Graph Drawing, pages pp. 222-228, Colonial Wiliamsburg. 2001

[8] HAREL D., KOREN Y., *A fast multi-scale method for drawing large graphs,* In J. Marks, editor, Proc. 8th International Symposium on Graph Drawing (GD 2000), LNCS 1984, pages 183–196, 2001. Springer-Verlag.

[9] KAMADA T., KAWAI S., *An algorithm for drawing general undirected graphs,* Information Processing Letters, 31, (l), 7–15 (1989).

[10] KOCAY W., PANTEL C., *An algorithm for finding a planar layout of a graph with a regular polygon as outer face,* Utilitas Mathematica 48 (1995), 161-178

[11] KOREN Y., CARMEL L., HAREL D., *Ace: A fast multiscale eigenvector computation for drawing huge graphs,* In Proc. IEEE Symposium on Information Visualization (InfoVis 2002), pages 137–144, 2002

[12] NOACK A., *An Energy Model for Visual Graph Clustering,* In Liotta, Giuseppe, Eds. Proceedings Graph Drawing, pages pp. 425-436, Perugia, 2004.

[13]] Omote, H. and Sugiyama, K., *Method for Drawing Intersecting Clustered Graphs and Its Application to Web Ontology Language*, in Proc. Asia-Pacific Symposium on Information Visualisation 2006, pp. 89-92, Tokyo, Japan, February 1-3, 2006. CRPIT

[14] SHAHROKHI, F., SYKORA, O., SZEKELY, L.A., VRTO, I., *Book embedding and Crossing Numbers*, in Proceedings of 20th International Workshop on Graph-Theoretic Concepts in Computer Science, LNCS 903, Springer-Verlag, 1995, 112-122.

[15] TRUONG, Q. D. and DKAKI, T., *ViAGraph: a Tool for Graph Visualization and Analysis,* In Proceedings International Workshop on Webometrics, Informetrics and Scientometrics & Seventh COLLNET Meeting, Nancy (France)

[16] TUTTLE W.T., *How to draw a graph*, Proceeding of the London Mathematical Society, 3(13): 743 – 768 1963