http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (8): 1491-1501, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.1491.1501
© 2013 Asian Network for Scientific Information

Drawing Clustered Graph Using Modular Decomposition Tree

[1]Wan-Yu Deng , [1]Kai Zhang, [1]Qing-Hua Zheng and [2]Wei Wei
[1]School of Computer, Xi'an University of Posts and Telecommunications,
Xi'an, Shaan'xi , RPC, 710121, China
[2]School of Computer Science and Engineering, Xi'an University of Technology,
Xi'an, 710048, China

Abstract: Compared with macro visualization like small-world structure in WWW, some fields such as knowledge visualization need such layout that can show detailed information of nodes and at the same time can reveal clustered structure of the Graph. Based on modular decomposition, energy model and adjustable center distance, one hierarchical layout algorithm was proposed. Through modular decomposition, the graph was firstly represented by a tree hierarchically. The local positions were then obtained from bottom to top and then the global positions are obtained from top to bottom. The experimental results on various datasets showed that the algorithm can achieve artistic and nearly non-overlapping appearance.

Key words: Graph layout, modular decomposition, information visualization

INTRODUCTION

Given a graph G = (V, E) consists of a set of vertices V and a set of edges $E \subseteq V^{(2)}$, graph layout $p{:}V \rightarrow \Re$ is a function that assigns to each vertex a position in the d-dimensional space where d is the number of dimensions and $\Re$ is the set of real numbers. For graph drawing, nodes are usually represented as disks or boxes while the edges are always represented as straight lines or curves.

Graph layout is an assistant tool for exploratory observing the data inherent characteristics in many fields, such as social network (Ghoniem *et al*., 2005), sensor nets (Wei *et al*., 2010a, b; Wei and Zhou, 2012) web macrostructure, bioinformatics (Battista *et al*., 1994), etc. Additionally, it also can serve as a knowledge map to support Information visualization. There is not absolute criterion to measure the quality of layout. It is expected to reveal clusters in web macrostructure mining, while for navigation it is more important to prevent overlapping of nodes. The proposed algorithm will hold three properties:

- Nodes are non-overlapping such that the contents have unobstructed view for user
- Can reveal the clustered structure and render them using different color
- Can highlight center vertices

There have been many graph layout algorithms available and the most popular are energy-based algorithms. These algorithms treat graph layout as energy system and computes positions through energy minimization. The vertex-repulsion LinLog energy model (Noack, 2004) was the first energy model to compute clustering layouts for graphs with uniform edge degree, and the edge-repulsion LinLog model can tackle this limitation. Hierarchical clustering layout algorithm (Clemencon *et al*., 2011) reveals cluster structure through hierarchical maximal modularity clustering.

Motivated by modular decomposition-based layout algorithm (MDLayout) (Charis and Constantinos, 2006), an improved modular decomposition-based layout algorithm (IMDLayout) is designed. It first decomposes the graph into the modular decomposition tree (MD-tree) and then computes the positions hieratically according the tree. IMDLayout has more artistic sense and control ability for different applications. In IMDLayout P-nodes and S-nodes are drawn using modified circle layout algorithm which has lower complexity and more artistic than the grid. Fatherly, P- and S-node layout are optimized when it has only one adjacent to reflect core role of the adjacency node and improve aesthetic quality. One parameter λ is introduced for extending traditional center distance to prevent overlapping.

Corresponding Author: Wan-Yu Deng, School of Computer, Xi'an University of Posts and Telecommunications,
Xi'an, Shaan 'xi , RPC, 710121, China Tel: (086)13379284746 Fax: 029-88166055

RELATIVE WORK

A module is a set of vertices that have the same neighborhood A module does not overlap any other is called strong module. In the following, the model is considered with respect to the strong module if not specify particular. The family of modules can be represented by a tree which is called modular decomposition tree (MD-tree) (Gallai, 1967) for the graph. The nodes are divided into three types:

- **Parallel node:** If the graph is disconnected the root is called parallel node
- **Serial node:** If the graph's complement is disconnected, the root is called serial node
- **Prime node:** In all other cases the root is called prime node

As one example, One graph G and its MD-tree T(G) are displayed in Fig. 1. Modular decomposition is very important for graph theory. The first Modular decomposition algorithm is due to Cowan, James and Stanton (Battista *et al*., 1999) and runs in O(n^4). One cubic time algorithm is proposed by Habib and Maurer (1979). The first linear time algorithms appeared in 1994 (Cournier and Habib, 1994). Since then a series of simplified algorithms has been published. Some of them running in linear time, some others in almost linear time (Dahlhaus *et al*., 2001; McConnell and Spinrad, 2000) . In this study, the linear time algorithm proposed by Tedder *et al.* (2008) was adopted. Graph layout based on MD-tree has the following structure (Charis and Constantinos, 2006):

- Compute the MD tree T(G)of the input graph G using one of decomposition algorithms
- In a bottom-up fashion, traverse the MD-tree T(G) and calculate the relative drawing for every internal node t∈T(G)
- Set the position of the root and traverse the MD-tree T(G) from top to bottom. For every internal node t, the displacement dis(t) is added to its children t_i∈ch(t) and obtains the final position

THE PROPOSED IMDLAYOUT ALGORITHM

To make the node non-overlapping and the content tag on it has clear view for user, the radius and the content's length of node are both considered in IMDLayout algorithm. As shown in Fig. 2, b(t) = (w(t), h(t))∈$\Re^2$ is the size of the box of t, where, w(t) depends on the content' s length while h(t) = 2*r(t). The edges are drawn using straight-line while its width is proportional to the closeness degree among vertices. The nodes in

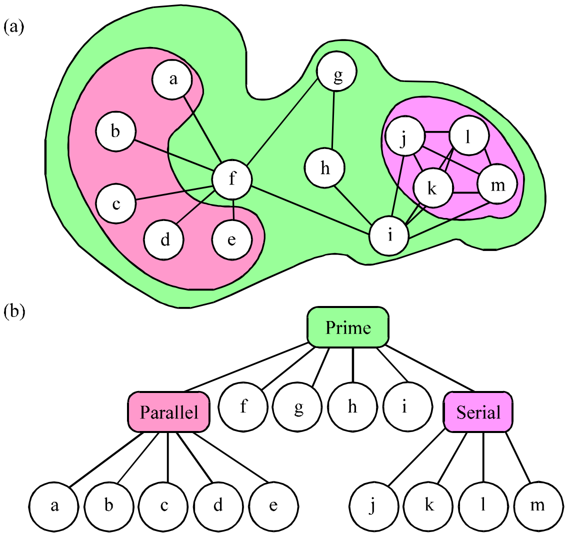

Fig. 1(a-b) : Graph and its modular decomposition tree (a) Graph (b) MD tree

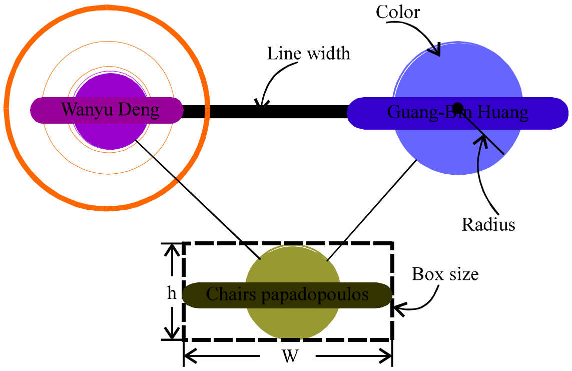

Fig. 2(a-b): Description of node and edge (a) Charis Papadopoulos, etc. (b) the proposed algorithm

cluster are rendered using the same color. p(t) =(x(t), y(t))∈$\Re^2$ represents the coordinates of. If one vertice is focused, it is surrounded by three orange circles with increasing radius.

The main algorithm: The MD-tree T(G) are first obtained by one of the known linear-time algorithms(Tedder *et al*., 2008). Then in bottom-up fashion, the relative drawing (b(t), r(t), dis(t_i),pos(t_i)) is computed for every internal node t∈T(G) where, b(t) is box size of node t, r(t) is the radius of node t, dis(t_i) dis(t) is displacement from t_i∈child(t) to t and pos(t_i) is relative position of t_i∈child(t). The position p(t), displacement dis(t), box size b(t) and radius r(t) are first initialized to zero for each internal node t∈T(G). For every leaf nodes, the center position p(t), displacement disp(t) is initialized to zero while the box size b(t) and radius r(t) are computed by predefined size since they are the vertices of G.

In bottom-up fashion, the relative coordinates $p(t_i)$ for every node $t_i \in child(t)$ are computed taking into account the dimensions of the box $b(t_i)$. Then compute the boundaries to decide box size b(t) and position p(t) in proportion to the frame boundaries of the relative drawing. The relative displacements $disp(t_i)$ from t_i to the parent can be obtained. For P-, S-node, circle layout is applied while for N-node energy model is adopted, respectively.

In up-bottom fashion, root node $t \in L_0$ position is first set as b(t)/2, The MD-tree T(G) is traversed in a top-down fashion and for every internal node $t \in T(G)$, the displacement $dis(t_i)$ is added to the position of $t \in T(G)$. In this way, all the vertices of G(M(t)) obtain the right coordinates relative to the center of their ancestor node t. The IMDLayout algorithm can be summarized as following:

```
Algorithm 1: IMDLayout Algorithm
Input: Graph $G = \{V, E|V = \{v_1, v_2, ..., v_m\}, E = \{e_1, e_2, ..., e_n\}\}$
Output: Position of all leaf nodes $p(G) = \{<t, p(t)>|t \in leaf(T)\}$
        Construct the MD tree T of the graph G;
        Initialize the center position $p(t) \leftarrow (0, 0)$ and displacement $dis(t) \leftarrow (0,0)$
        for every node $t \in T$ (including both internal nodes and leaf nodes); set
        box size $b(t) \leftarrow (0,0)$, radius $r(t) \leftarrow 0$ for every internal nodes, while
$b(t) \leftarrow (w, h)$, $r(t) \leftarrow w^2+h^2$ for every leaf nodes of T;
3       for i = h-1 up to 0 do {bottom-up}
3.1     Get the node sets Li of the i-th level;
              3.2 for each internal node $t \in L_i$ do
                   if t is a Parallel-node or Serial-node then
                     $(b(t), r(t), dis(t_i), pos(t_i)) \neg CircleLayout(t)$
                   else
                     $(b(t), r(t), dis(t_i), pos(t_i)) \neg ? -LinLogLayout(t)$
4. Set root node $t \in L_0$ position as $p(t) \leftarrow b(t)/2$
for i = 0 down to h-1 do {up-bottom}
 for each internal node $t \in L_i$ do
 for each child $t_i \in child(t)$ do
                     $p(t_i) \neg dis(t_i) + p(t)$
return $p(G) = \{<t, p(t)>|t \in leaf(T)\}$
```

The combine and divide: The Fig. 3 and 4 is the layout result of before and after combine. The Q node is surrounded by children of P- or S-node and it is obvious that Q is their center. Moreover, the adjusted layout has more aesthetic feeling than that of layout before adjustment. This adjustment is implemented by two operations: combine and divide. In order to improve efficiency, the adjustment operation is embedded into circle layout algorithm and prime layout, i.e., before circle layout and prime layout, these node pairs that need combine are firstly found and combined them. After layout, they will be divided again and compute the layout for them respectively. Obviously, the first work is to find which nodes should be combined.

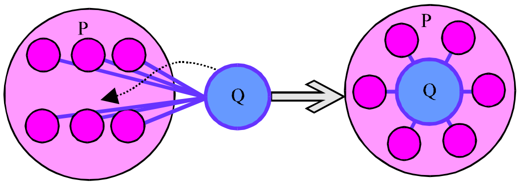

Fig. 3: Appearance of P node combined or not (a) Charis Papadopoulos, etc. (b) the proposed algorithm

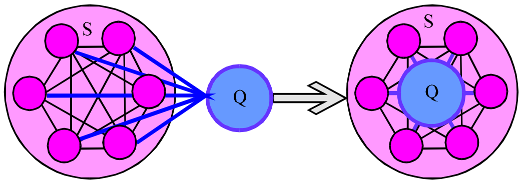

Fig. 4: Appearance of S node combined or not

As shown in Fig. 5, The parent may be S-, P- and N-node. Since there are no edges among children of P-node, P can't adjoin with Q, so they don't need to combine. When the parent is S-node, any two different nodes are all connected and thus the case need to combine only occurs when the sum of nodes in the serial subgraph is two. If the parent is N-node, P is checked whether has only one adjacent. From the above analysis, the combining condition can be summarized as: 1) the parent is S-node and the number of children is two; or 2) the parent is N-node and S-node or P-node exist who have only adjacent.

Combine operation mainly includes: Firstly, combine the children of P and the node Q to produce a new node S. In order to divide them in the divide step, the P, Q and P, S are stored into a map. Secondly, like circle algorithm, children of P are set around the Q node. The difference is that the radius can't be smaller than Q-node;s radius, i.e., if the obtained radius is smaller than radius sum, then it should be extended. Finally, compute the displacement and the box size for P and his children. Because the children of P are often outsiders of S, the position of P is same to S node. Therefore, the displacement from P to S is zero and the b(p) is same to b(s). The displacement of the children is $pos(t_i)$. For Q node, the position is (0, 0) and therefore its displacement is $dis(t_i)$. Its box doesn't change. The P and Q are removed from children set of P-node and the new node S was inserted. The combine operation can be summarized as followings.

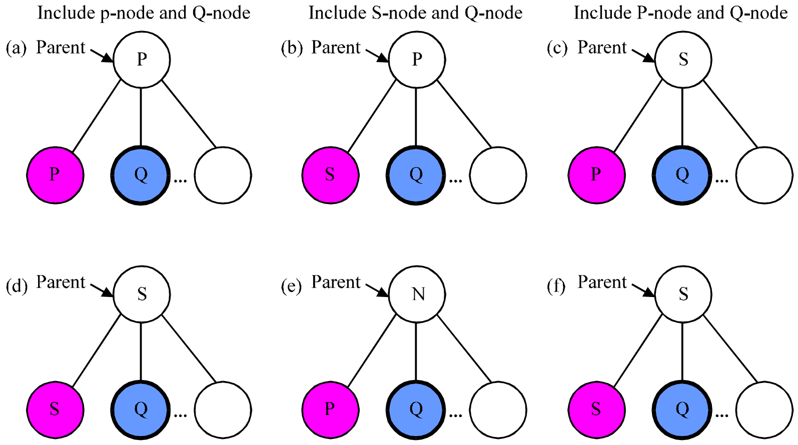

Fig. 5: Six cases in modular decomposition tree

```
Algorithm 2: Combine
Input: Internal node t and child(t)
Output: child(t)
1     if (is Serial-node and |child(t)| = 2) or (is Prime-node) then
 for each p∈child(t) do
 if p is Parallel-node and only connect with one another node q then
<p, q>→$M_{p\text{-}q}$;
s→p⊕q;
<p, s→<$M_{p\text{-}s}$;
2     for eachindo
2.1 q←$M_{p\text{-}q}$;
2.2 s←$M_{p\text{-}s}$;
2.3 Put |child(p)| nodes aronnd q and set α→2π/|child(p)|;
2.4 for each child ($t_i$)∈child(p) do
        $\rho_{pre} \neg \frac{r(t_i)+r(t_{i-1})}{2\sin(a/2)}$ and $\rho_{aft} \neg \frac{r(t_i)+r(t_{i+1})}{2\sin(a/2)}$
if $b(t_j) = b(t_{i-1})$ then $p(t_j) \leftarrow p_{pre}$
else if $b(t_i)=b(t_{i-1})$ then $\rho(t_i) \neg \rho_{pre}$ else $b(t_i)=b(t_{i-1})$
if $\rho(t_i) \neg \rho_{pre}$ then $b(t_i)=b(t_{i-1})$ ;
2.5 if $b(t_i)=b(t_{i-1})$ then $b(t_i)=b(t_{i-1})$ else $\rho(t_i) \neg \rho_{pre}$
 2.6 $b(s) \neg \max(pos(t_i)-r(t_i))-\min(pos(t_i)-r(t_i))$
2.7 $p(s) \rightarrow (\max(pos(t_i)-r(t_i))+\min(pos(t_i)-r(t_i)))/2$
2.8 $r(s) \rightarrow \|b(s)\|/2$
2.9 $\mathbf{p(p) \rightarrow p(s)}$
2.10 $\mathbf{b(p) \leftarrow b(s)}$
2.10 $\mathbf{dis(p) \leftarrow p(p) - p(s) = (0,0)}$
2.11 $\mathbf{dis(t_i) \leftarrow pos(t_i) - p(p)}$
2.12 $\mathbf{dis(q) \leftarrow -p(s)}$
2.13 $child(t)' \leftarrow child(t) \dot{-} \{p,q\} \dot{+} \{s\}$
return $\{child(t)', M_{p\rightarrow q}, M_{p\rightarrow s}\}$
```

After get the processed children set, graph layout was carried according to their type through the corresponding algorithm. The displacement of P is dis(t_i). The divide operation can be summarized as follows:

```
Algorithm 3: Divide
Input: $\{child(t)', M_{p\rightarrow q}, M_{p\rightarrow s}, dis(t_i'), dis(p), dis(q)\}$
Output: dis($t_i$)
for each p in M do
2.1  $s \leftarrow M_{p\rightarrow s}(p)$
2.2  $q \leftarrow M_{p\rightarrow q}(p)$
2.3  $\mathbf{dis(p) \leftarrow dis(s) + dis(p)}$
2.4  $\mathbf{dis(q) \leftarrow dis(s) + dis(q)}$
 $dis(t_i) \leftarrow dis(t_i') \dot{-} dis(s) \dot{+} dis(p) \dot{+} dis(q)$
return $\{child(t)', M_{p\rightarrow q}, M_{p\rightarrow s}\}$
```

Circle layout for P- and S-node: Circular drawing satisfies the aesthetic criterion of symmetry and is the usual way of representing complete graphs and edgeless graph. The vertices are placed in an arbitrary order on equal arcs, on the circumference of a cycle centered at (0, 0). For each node t_i∈child, two radiuses p_{pre} and p_{aft} that define two concentric circles centered at (0, 0) are computed. In the circle defined by p_{pre}, vertex t_i and its previous t_i-1 are placed on a α degrees arc, so that the minimum distance from their boundaries, is k. The radius p_{aft} defines a circle where t_i and t_i+1are placed in the same way. To prevent overlaps, p(t_i) is set to be the maximum of these two radiuses. In the special case where the box b(t_i) of t_i has the same dimensions with any of the two adjacent vertices in the circle, is set to be the radius defined by the equal sized box adjacent vertex. Finally, every vertex t_i are drew on the circle defined by radius p(t_i), with angle α. Obviously, for a complete graph with uniform nodes the drawing is a perfect circle. The formal description of circle layout is given in details as.

Algorithm 4: Circle layout

Input: internal node $t \in T$, inner circle radius δ

Output: $(b(t), r(t), dis(t_i), pos(t_i))$ where $t_i \in child(t)$

```
1.    Child (t)'←Combine(t,child(t))
2.    if |child (t)'| = 1 then
 p(t)←(0,0);
s←child (t)'b(t)←b(s); r(t)←r(s);
 else
            for each child (t)' ∈child (t)' do
              Set border distance of two nodes as k (default 0);
                  Compute and $\rho_{pre} \neg \frac{r(t_i)+k+r(t_{i\,1})}{2\sin(a/2)}$ and $?_{aft} \neg \frac{r(t_i)+k+r(t_{i+1})}{2\sin(a/2)}$
                   where $t_{i-1}$ and $t_{i+1}$ are the adjacent vertices of $t_i$ on the
                  circle;
              if $(bt_i) = b(t_{i-1})$ then $\rho(t_i) \leftarrow _{pre}$;
             else if $(bt_i) = b(t_{i-1})$ then $\rho(t_i) \leftarrow _{pre}$;
             else $\rho(t_i) \leftarrow \max(\rho_{re}, \rho_{aft})$;
                 if a= π then $pos(t_i) \neg (?(t_i)*\cos a, ?(t_i)*\sin a)$ else
        $pos(t_i) \neg (?(t_i)*\sin a, ?(t_i)*\cos a)$
b(t)←max(pos(t_i)-r(t_i)-min(pos(t_i)-r(t_i)))'^2
      r(t)←||b(t)||/2;
            for each child $t_i \in$child(t)' do
            dis(t_i)←pos(t_i)-p(t);
      (B(t), r(t), dis(t_i), pos(t_i))←Divide(t, child(t))
            return (b(t), r(t), dis(t_i), pos(t_i))
```

Modified energy based layout for N-node: Energy model are popular for creating straight-line drawings of general undirected graphs. For a given layout p and two vertices u and v, the term $||p_u-p_v||$ denotes the Euclidean distance of the two vertices. Generic energy model including three parameters to strengthen its adaptive capacity for different applications :

$$\xi(p) = \begin{cases} \sum\limits_{\{u,v\}\hat{I}E} w_{\{u,v\}} \frac{\|p_u - p_v\|^a}{a} - \sum\limits_{\{u,v\}\hat{I}V^{(2)}} (w_u w_v)^e \ln\|p_u - p_v\| & r = 0 \\ \sum\limits_{\{u,v\}\hat{I}E} w_{\{u,v\}} \frac{\|p_u - p_v\|^a}{a} - \sum\limits_{\{u,v\}\hat{I}V^{(2)}} (w_u w_v)^e \|p_u - p_v\|^r & r^{1}0 \end{cases}$$

where, α is exponent to the distance in the attraction term; r applies value r as exponent to the distance in the repulsion term if $r \neq 0$ and applies the logarithm to the distance in the repulsion term if r = 0; e eliminates the edge-weight factor in the repulsion term by e setting to value 0. When $\alpha = 1$, r = 0 and e = 1, i.e., the energy model is the weighted edge-repulsion energy model. The (3,0,0)-energy model has been proposed by Fruchterman Reingold, (1,0,0)-energy is known as Vertex-repulsion LinLog and (1,0,1)-energy is known as Edge-repulsion LinLog.

In the experiments on generic model, overlapping phenomenon always occurs especially when the radius of nodes are very nonuniform. One of the important reasons lies in that center distance without regulating the impact of radius is applied. In some literature, center distance is replaced with boundary distance to prevent node overlapping. In the implements, cotree structure are introduced to accelerate computation and don't use the distance between nodes directly, so boundary distance isn't improper here. To address this problem, a parameter λ is introduced to regulate the center distance, i.e. regulated distance $d' = (d/\lambda)$. Thus the model can be modified as:

$$\xi(p) = \begin{cases} \sum\limits_{\{u,v\}\hat{I}E} w_{\{u,v\}} \frac{\left(\frac{\|p_u - p_v\|}{\lambda}\right)^a}{a} - \sum\limits_{\{u,v\}\hat{I}V^{(2)}} (w_u w_v)^e \ln\frac{\|p_u - p_v\|}{\lambda} & r = 0 \\ \sum\limits_{\{u,v\}\hat{I}E} w_{\{u,v\}} \frac{\left(\frac{\|p_u - p_v\|}{\lambda}\right)^a}{a} - \sum\limits_{\{u,v\}\hat{I}V^{(2)}} (w_u w_v)^e \left(\frac{\|p_u - p_v\|}{\lambda}\right)^r & r^{1}0 \end{cases}$$

The strengths of the forces are often chosen to be proportional to some power of the distance. Formally, for a layout p and two vertices u, v with $u \neq v$, the attractive force exerted on u by v is:

$$w_{\{u,v\}} \|p_u - p_v\|^a \overrightarrow{p_u p_v}$$

and the repulsive force exerted on u by v is:

$$w_u w_v \|p_u - p_v\|^r \overrightarrow{p_v p_u}$$

where, $||p_u-p_v||$ is the distance between u and v, $\overrightarrow{p_v p_u}$ is the unit-length vector pointing from u to v and a and r are real constants with a>r.

The condition a>r ensures that the attractive force between connected vertices grows faster than the repulsive force. For most practical force models holds $a \geq 0$ and $r \leq 0$, i.e., the attractive force is non-decreasing and the repulsive is non-increasing with growing distance.

As shown in Fig. 6, the system will stop at equilibrium point (d'_0) after iterative learning. Because $d' = (d/\lambda)$, the center distance between two nodes is $d_0 = \lambda\, d'_0$. Through adjusting λ, proper distance that prevents overlapping can be obtained. When $\lambda > 1$, the system will reach equilibrium status in advance and prevent overlapping.

For energy minimization, the hierarchical energy minimization algorithm of Barnes and Hut (Barnes and Hut 1986)is introduced to graph drawing. Its runtime is in O(|E|+|v|log|v|) per iteration. The overall runtime grows somewhat faster because the number of iterations needed for convergence tends to grow with. Some other efficient minimization algorithms are not expected to find good energy minima for clustering energy models like LinLog

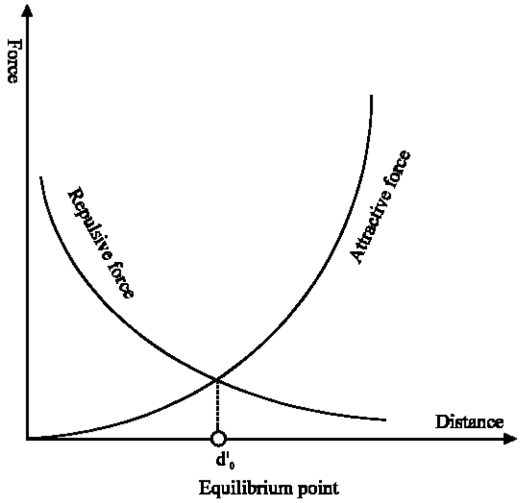

Fig. 6: Force equilibrium

and for graphs with small diameter (Hachul and Junger, 2005).

```
Algorithm 5:
Input: internal node t∈T
Output: (b(t),r(t), dis(t_i), pos(t_i)) where t_i∈ child(t)
1. while iterations iterations ≤ maxiter do
        Create Octree tree←Create Octree tree();
        for each vertex t_i∈child(t) do
              dirt(t_i)←(p(t_i)-r(tree))*r*r(t_i)*r(tree)*||p(t_i)-p(tree)||^{r-2};
 p(t_i)← p(t_i)+η*dir(t_i);
2. b(t)←max(pos(t_i)-r(t_i)-min(pos(t_i)-r(t_i))
3. b(t)←max(pos(t_i)-r(t_i)-min(pos(t_i)-r(t_i)))^{/2}
4. R(t)←||b(t)||/2;
5. for each child t_i∈child(t)' do
        dis(t_i)←pos(t_i)-p(t);
6. (B(t), r(t), dis(t_i), pos(t_i))←Divide(t, child(t))
              return (b(t), r(t), dis(t_i), pos(t_i))
```

Cluster structure in graph: A clusteringof a graph partitions the vertex set into disjoint subsets called clusters and thereby maps each vertex v to a cluster c(v). Proposals of quality measures for clustering are numerous and scattered over the literature of diverse research fields. One of the most widely used was introduced by Newman and Girvan and is called modularity. The modularity of a clustering is:

$$\sum_{c\in p(V)}\left(\frac{w_{\{c,c\}}}{w_{\{V,V\}}}-\frac{\frac{1}{2}w_c^2}{\frac{1}{2}w_V^2}\right)$$

where, V is the set of all vertices in the graph and c(V) is the set of clusters, $w_{(c,c)}$ is the total edge weight within the cluster c and $w_{(c)}$ is the total weight of the vertices in c. Intuitively, the first term of the modularity measure is the actual fraction of intra-cluster edge weight. In itself, it is not a good measure of clustering quality, because it takes the maximum value for the trivial clustering where one cluster contains all vertices. This is corrected by subtracting the second term, which specifies the expected fraction of intra-cluster edge weight in a network with uniform density.

EXPERIMENTAL RESULTS

MDLayout, IMDLayout and Linloglayout are tested on Karate, Hitech and DBLP dataset. Hitech data describes the 147 friends= relationship data between 33 employees. Karate data describes the multiple links between karate club members (for example, relations between the university students to participate in the same stage game, etc.), the data set includes 34 members and 156 relational data. DBLP indexes more than 1.8 million articles and contains many coauthor sip relations of computer scientists. It is impossible visual all of them and 3 cases are considered: one author with his coauthors, a research community and multi research community.

The results on Karate are shown in Fig. 7. The optimized parameter values for IMDLayout are (λ = 0.55, a = -0.25, r = 3.0, e = 0.15), (λ = 0.6, a = -0.25, r = 3.0, e = 0.15) for MDLayout and (λ = 1, a = -0.25, r = 3.0, e = 0.15) for LinlogLayout. It can be found that the three algorithms can reveal clusters and prevent overlapping effectively through adjusting parameter values.

The result on Hitech is demonstrated on Fig. 8. The optimized parameter values for IMDLayout are (λ = 0.55,

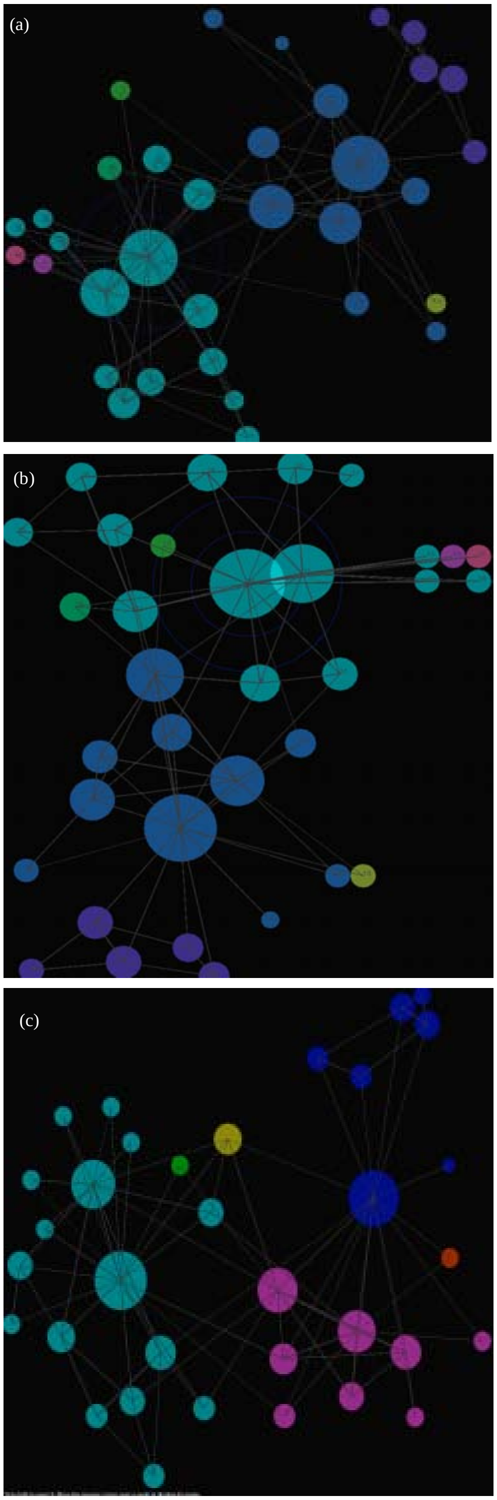

Fig. 7(a-c): (a)Improved modular decomposition-based layout(0.55, -0.25, 3.0, 0.15), (b) Modular decomposition-based layout(0. 6, -0.25, 3.0, 0.15) and (c) Linlog Layout (λ,-0.25, 3.0, 0.15)

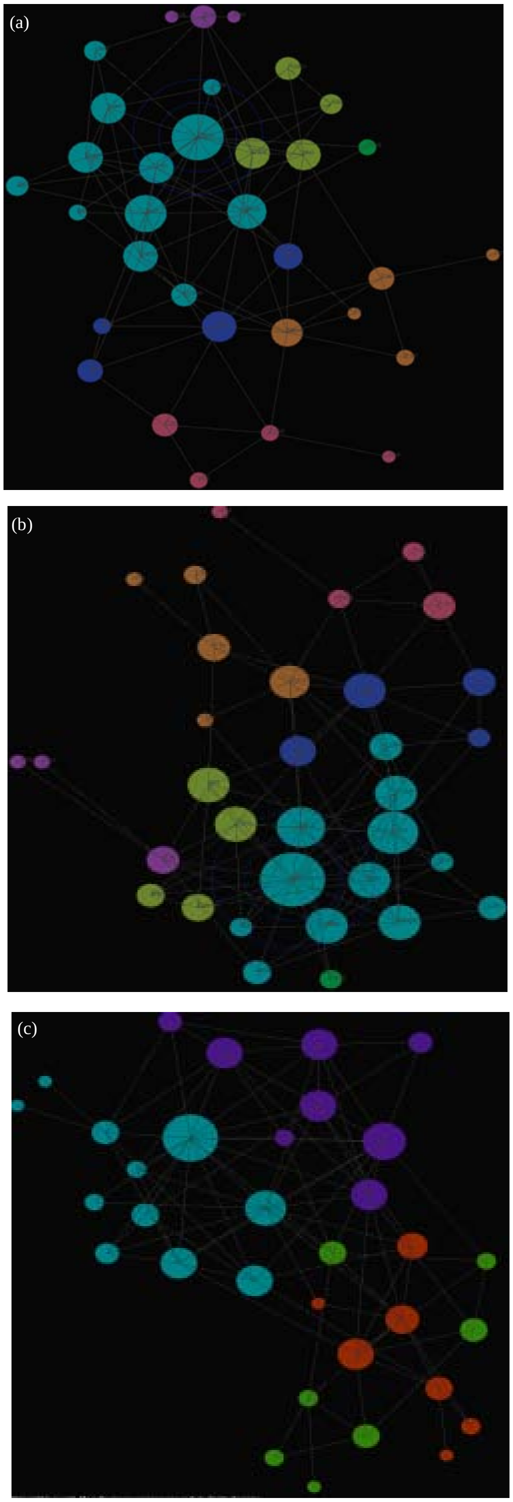

Fig. 8(a-c): (a) IMD layout for Hitech dataset(0.55, -0.25, 3.0, 0.15) (b) MD layout for Hitech dataset(0..6, -0.25, 3.0, 0.15) (c) Linlog layout for Hitech dataset (λ,-0.25, 3.0, 0.15)

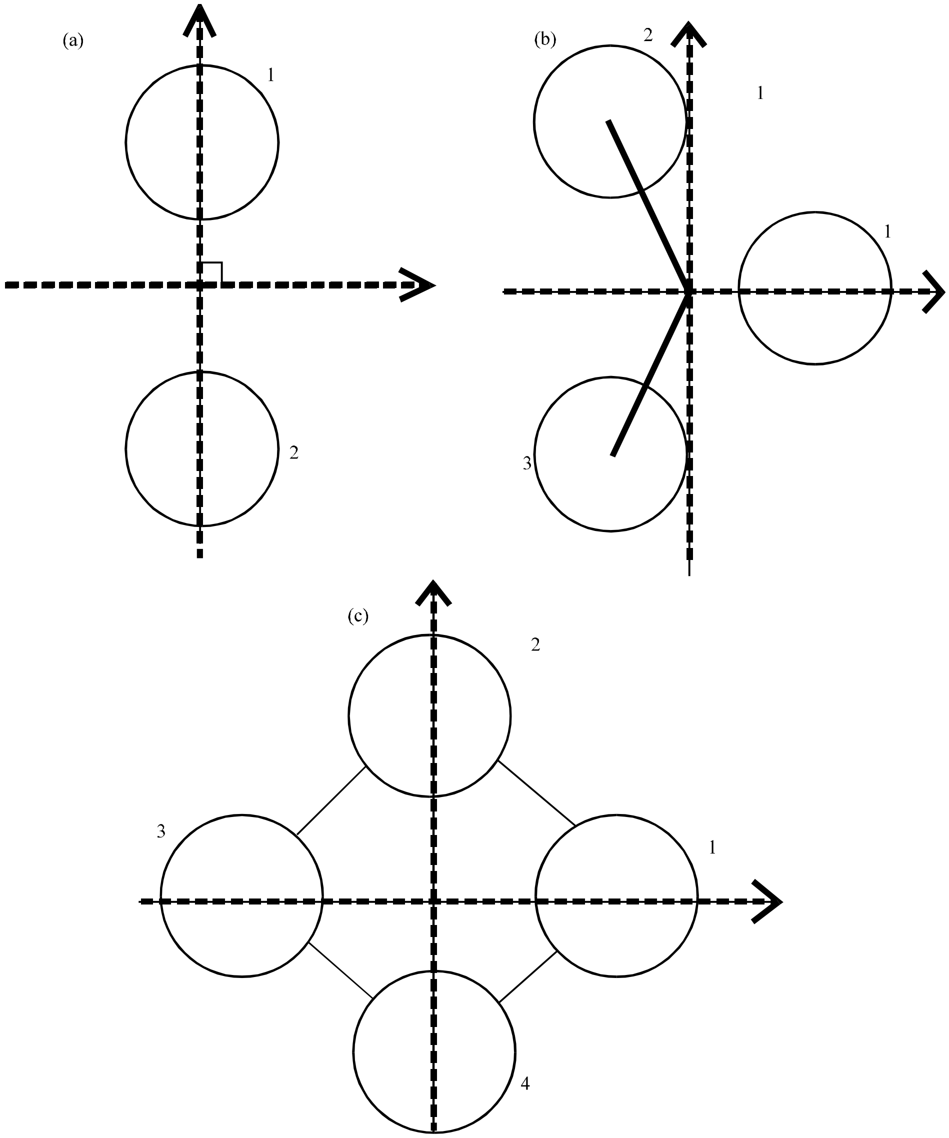

Fig. 9(a-c): (a) Improved modular decomposition-based layout on coauthor ship data(λ, -0.25, 3.0, 0.15) three nodes, (b) Modular decomposition-based layout on coauthor ship data (λ,-0.25, 3.0, 0.15) and (c) Linlog Layout on coauthor ship data(λ, -0.25, 3.0, 0.15)

a = -0.25, r = 3.0, e = 0.15), (λ = 0.6, a = -0.25, r = 3.0, e = 0.15) for MDLayout and (λ = 1, a = -0.25, r = 3.0, e = 0.15) for LinLogLayout, respectively. It can be found that IMDLayout can reveal clusters clearly.

The results on DBLP are demonstrated in Fig. 9, It can be found that the proposed algorithm can put the acroteric nodes around the center. This screw style can make the layout more aesthetic and prevent overlapping. Additionally, in order to make the content tag have clearly view for users, horizontal nodes are far away as possible and thus for two nodes, they are arranged from $\pi/2$ as shown in Fig. 10a, while for three nodes or more nodes, will be arranged from 0 as shown in Fig. 10b and c. From Fig. 9. It can be found that IMDLayout can show modular structure more artistic than other two algorithms.

In order to make it more clearly one author and his coauthors are displayed since they are typical circle data as shown in Fig. 11. It can be found that IMDLayout is like "Atrumpet shell" and more pretty than other algorithms. The nodes are sorted ascendant firstly such that the layout is more compact and regularly.

From Fig. 11a, it can be observed the radius are not sorted like mentioned before, the reason lies that the sorted radius considering the content tag's length. The

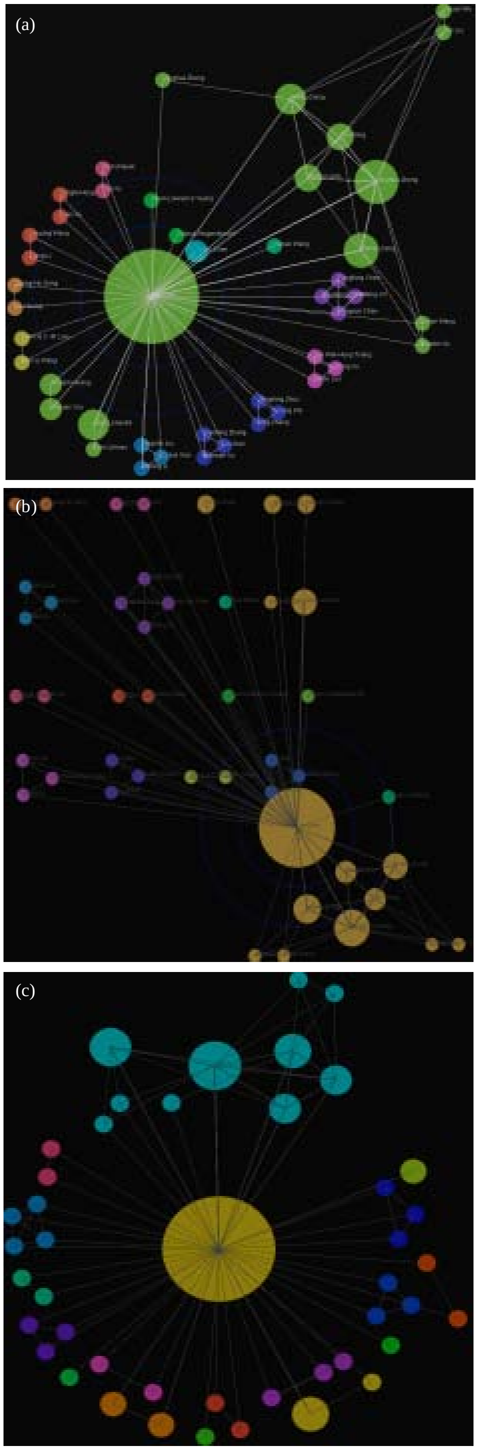

Fig.10(a-c): (a) IMD Layout on coauthor ship data (λ, -0.25, 3.0, 0.15) (b) MD Layout on coauthor ship data (λ,-0.25, 3.0, 0.15) and (c) Linlog Layout on coauthor ship data (λ, -0.25, 3.0, 0.15)

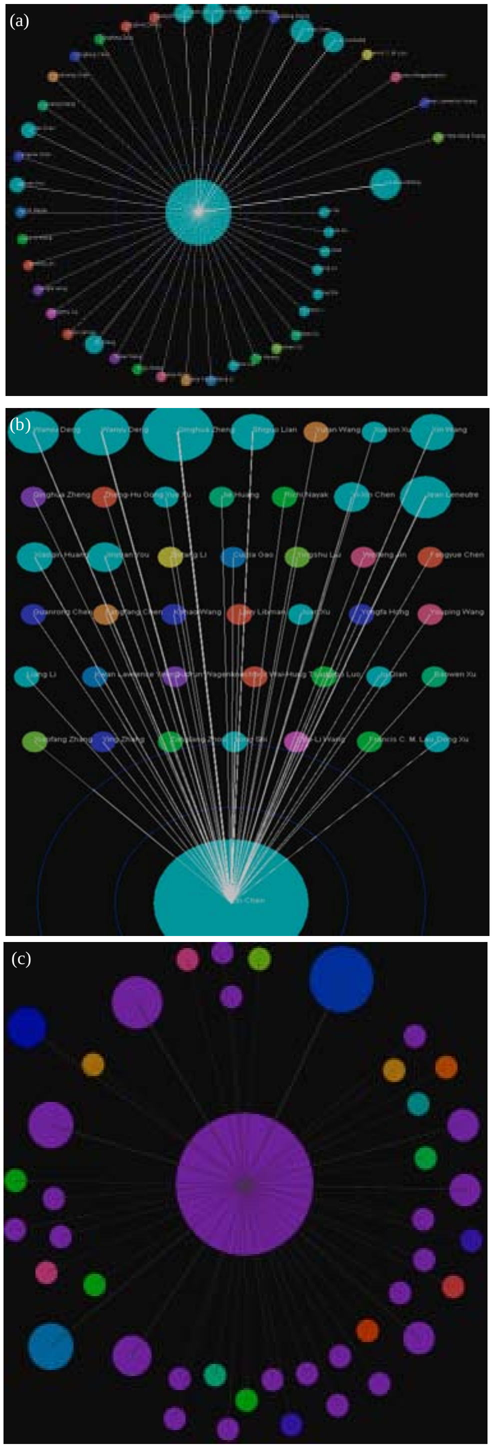

Fig. 11(a-c): (a) Improved modular decomposition-based layout on coauthor data (b) Modular decomposition-based layout on coauthor data and (c) Linlog Layout on coauthor data (λ, -0.25, 3.0, 0.15)

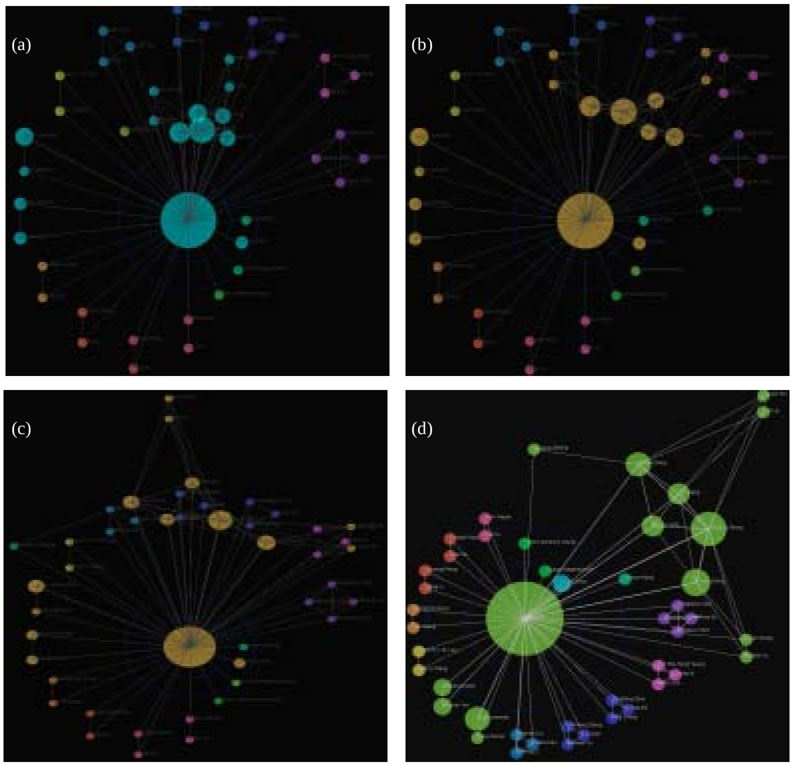

Fig. 12(a-d): Improved modular decomposition-based layout for 4 different distance parameter values (a) λ = 1, (b) λ = 0.7, (c) λ = 0.4 and (d) λ = 0.3

content tag are longer than in the front of serial. For example "AIvor Wai-Hung Tsang", "AKwan Lawrence Yeung" these long content node are put large radius circle to make it can show without overlapping.

In order to observe distance parameter's affect on the layout, four results are listed for λ = 1, λ = 0.7, λ = 0.4 and λ = 0.3 as shown in Fig. 12. When λ = 1, the distance is unadjusted center distance which is used popularly, one module is attracted strongly by another such that it enter inside the another module. But with the change of λ, i.e., from 0.7 to 0.3, the module gradually away and when λ = 0.3 there isn't overlapping.

CONCLUSION AND FUTURE WORK

Through introducing distance parameter λ, the energy model is generalized to make it has more control ability for obtain non-overlapping result. Through modular decomposition, the S-type and P-type nodes (or subgraphs) are recognized which can be layouted by regular circle are participate in energy model layout and avoid complexity. Because the energy model can obtain real optimized value, the more nodes can increase the probability of overlapping. Although, through parameter λ, node with large radius overlapping can be prevent, nodes with small radius will be very far away from each other. Additionally, Because different graph need different parameter value, thus our future work is to propose one adaptive method to decide parameter value. Local Zoom like Google map to provide multi granularities displacement is our another future work.

ACKNOWLEDGMENTS

The study was partially supported by the National Science Foundation of China under Grant Nos. 61100166 and 61202184, the Science Research Plan of Shaanxi under Grant No. 11JK1035 and No.2013JK1139, and the Key Projects in the National Science under Grant No. 2009BAH51B00, China CNGI project under Grant No. CNGI-09-01-13, doctoral fund of ministry of education of china under Grant No. 20090201110060, the Fundamental Research Funds for the Central Universities under Grant No. xjj20100057.

REFERENCES

Barnes, J. and P. Hut, 1986. A hierarchical O (N log N) force-calculation algorithm. Nature, 324: 446-449.

Battista, G.D., P. Eades, R. Tamassia, and I.G. Tollis, 1994. Algorithms for drawing graphs: An annotated bibliography. Comput. Geo.Theor. Appli., 4: 235-282.

Battista, G.D., P. Eades, R. Tamassia and I.G. Tollis, 1999. Graph Drawing: Algorithms for the Visualization of Graphs. vol. 3. Prentice Hall New Jersey USA Pages: 397.

Charis, P. and V. Constantinos, 2006. Drawing graphs using modular decomposition. Proceedings of the Graph Drawing 13th International Symposium, September 12-14, 2005, Limerick, Ireland, pp: 343-354.

Clemencon, S., H. De Arazoza, F. Rossi and V.C. Tran, 2011. Hierarchical clustering for graph visualization. Proceedings of the 19th European Symposium on Artificial Neural Networks, April 27-29, 2011, Bruges, Belgium, pp: 1-6.

Cournier, A. and M. Habib, 1994. A new linear algorithm for modular decomposition. Proceedings of the 19th International Trees in Algebra and Programming, Volume 787, April 11-13, 1994, Colloquium Edinburgh, U.K., pp: 68-84.

Dahlhaus, E., J. Gustedt and R.M. McConnell, 2001. Efficient and practical algorithms for sequential modular decomposition. J. Algorithms, 41: 360-387.

Gallai, T., 1967. Transitiv orientierbare graphen. Acta Math. Acad. Sci. Hung., 18: 25-66.

Ghoniem, M., J.D. Fekete and P. Castagliola, 2005. On the readability of graphs using node-link and matrix-based representations: A cont. exper. Stat. Anal. Infor. Visual., 4: 114-135.

Habib, M., and M.C. Maurer, 1979. On the X-join decomposition for undirected graphs. Disc. Applied Math., 1: 201-207.

Hachul, S. and M. Junger, 2005. Drawing large graphs with a potential-field-based multilevel algorithm. Graph Drawing, 3383: 285-295.

McConnell, R.M. and J.P. Spinrad, 2000. Ordered vertex partitioning. Dis. Math. Theo. Comp. Sci., 4: 45-60.

Noack, A., 2004. An energy model for visual graph clustering. Graph Drawing, 2912: 425-436.

Tedder, M., D. Corneil, M. Habib and C. Paul, 2008. Simpler linear-time modular decomposition via recursive factorizing permutations. Proceedings of the 35th International Colloquium on Automata, Languages and Programming, July 7-11, 2008, Reykjavik,Iceland, pp: 634-645.

Wei, W. and B. Zhou, 2012. A p-laplace equation model for image denoising. Inform. Technol. J., 11: 632-636.

Wei, W., A. Gao, B. Zhou and Y. Mei, 2010a. Scheduling adjustment of mac protocols on cross layer for sensornets. Inform. Technol. J., 9: 1196-1201.

Wei, W., B. Zhou, A. Gao and Y. Mei, 2010b. A new approximation to information fields in sensor nets. Inform. Technol. J., 9: 1415-1420.