Skeletal Images as Visual Cues in Graph Visualization

I. Herman[1], M.S. Marshall[1], G. Melançon[1], D.J. Duke[2], M. Delest[3], and J.-P. Domenger[3]

[1] CWI
P.O. Box 94079
1090 GB Amsterdam, The Netherlands
{Ivan.Herman, Scott.Marshall, Guy.Melancon}@cwi.nl
[2] Department of Computer Science
The University of York
Heslington, York, YO10 5DD
duke@cs.york.ac.uk
[3] LaBRI, UMR 5800
351, Cours de la Libération
33405 Talence Cedex, France
{maylis,domenger}@labri.u-bordeaux.fr

Abstract. The problem of graph layout and drawing is fundamental to many approaches to the visualization of relational information structures. As the data set grows, the visualization problem is compounded by the need to reconcile the user's need for orientation cues with the danger of information overload. Put simply: How can we limit the number of visual elements on the screen so as not to overwhelm the user yet retain enough information that the user is able to navigate and explore the data set confidently? How can we provide orientational cues so that a user can understand the location of the current viewpoint in a large data set? These are problems inherent not only to graph drawing but information visualization in general. We propose a method which extracts the significant features of a directed acyclic graph as the basis for navigation [1].

1 Introduction

A fundamental challenge for information visualization applications that use graph visualization techniques for relational data sets is the scale and structural complexity of the data. Beyond the well known and researched problems of graph layout, large-scale data sets call for new approaches to navigation, and the provision of visual cues to support the user's awareness of the context or location within the data set. There is a large body of published research results in this area, which involve the use of zoom [8], pan, visual cues [3], and focus+context techniques using non-linear filters such as, for example, fish-eye views [7], hyperbolic geometry [5], and distortion-oriented presentations [9]. This paper contributes a method which can be used to produce a schematic view of a directed acyclic graph or DAG to the tools and techniques available for viewing graph structures.

[1] Note: The color figures and a demonstration applet are available at the web site: http://www.cwi.nl/InfoVisu.

The *skeleton* of a graph is the set of nodes and edges that are determined to be significant by a given metric. The skeleton can give the impression of a structural backbone. Because it is a selection of a small subset of important nodes, the skeleton eliminates the problem of information overload while still providing information essential for further exploration. The skeleton also allows the user to characterize a particular graph by providing a simple image which contains the most important features or 'landmarks' of a graph. In this way, the skeleton provides the user with a map for orientation and navigation. The features chosen by the metric may be structurally important or reflect some other measure. By changing the metrics used to extract the skeleton, we may produce different maps for different purposes.

The highlighting of trees according to the underlying Strahler values was proposed as an aid to navigation in [3]. In this paper, we will explain how to apply Strahler and other metrics to trees and DAGs to obtain a skeleton. Obviously, the metric is crucial in the determination of the skeleton. We have looked for metrics which result in a skeleton that is a good indicator of the underlying structure of the graph.

Our current methods require that the graph be acyclic. Although it is possible to extract a DAG from an arbitrary graph, for simplicity we have chosen in this phase of work to assume that the graph is already directed and acyclic. As with other types of graphs, DAGs can be quite overwhelming when the number of nodes is large and are found in many applications. This makes the DAG an excellent candidate for skeleton extraction. Of course, any result derived for DAGs is also applicable for trees.

2 General methodology

The general methodology for extracting skeletons is as follows:

- Choose a metric function for a graph and a threshold value.
- Traverse the graph and extract the nodes whose metric values are equal to or above the threshold value.
- Display the final skeleton. This consists of the nodes which have been extracted and the edges connecting them.
- Display the leftover nodes and edges, possibly merged or simplified.

The metric should reflect the relative significance of each node of the graph. The metric and the resulting skeleton should correspond to a clear mental model which aids the user during navigation. The threshold value determines the level of detail which is represented by the skeleton.

The last step of our method involves representing the nodes and edges not selected by the extraction process. For trees, the subtrees not belonging to the skeleton may be simply replaced by triangles or other shapes, resulting in a schematic view of the tree. Representing the non-skeletal nodes and edges from a DAG calls for more sophisticated techniques using different colours and intensities to distinguish between skeletal and non-skeletal parts of the DAG (See Section 2.3 for details).

Introductory example: Figure 1 shows the original structure of a tree. Figure 2 shows the skeleton which results from selecting the nodes which have metric values equal to or above a threshold value. Our program has replaced the excluded nodes by triangles

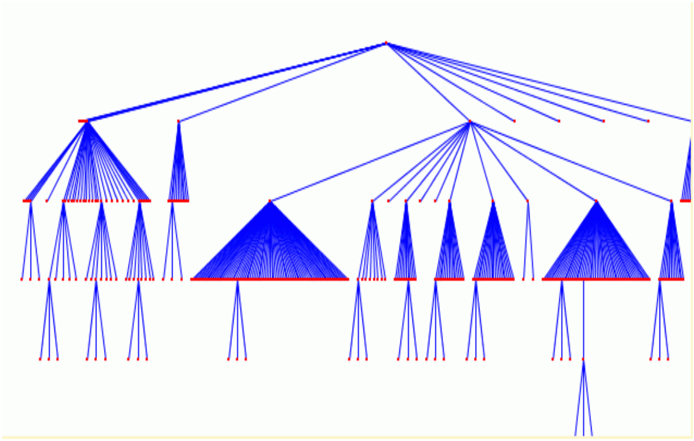

Fig. 1. A fully displayed tree

to create a *schematic view* of the tree. A schematic view is a simplified representation of a graph which makes use of the skeleton and replaces non-skeletal parts of the graph with lines or shapes.

2.1 Metrics

The extraction of a skeleton from a DAG requires the computation of a value for each node of a DAG in the same fashion as for trees. In this section, we will talk about two different metrics and give an impression of the skeletons which they give as a result. These metrics were chosen for two reasons. First, each can be explained in terms of a simple metaphor, which we believe will help users develop an intuition about the effect of the metric, without needing to understand the underlying mathematics. Second, experimental results have shown that these metrics provide an impression of the overall structure of the DAG. Later, we will also indicate how different metrics can be composed. The composition of metrics can produce quite useful results and can be applied much the same way as one might apply several optical filters to a camera lens.

The Strahler metric. We used Strahler numbers for trees as a metric to extract the skeleton in Figure 2. This metric was already presented in [3] and we will recall it here; for a full account on Strahler numbers see [2].

The Strahler value of a leaf is set to 1. For any other node v, a value is computed using the formula:

$$S(v) = \max(S(k_1), \ldots, S(k_p)) + \begin{cases} p-1 \text{ if all values } S(k_i) \text{ are equal} \\ p-2 \text{ otherwise} \end{cases} \tag{1}$$

where $k_1, \ldots, k_p$ are the successors of v.

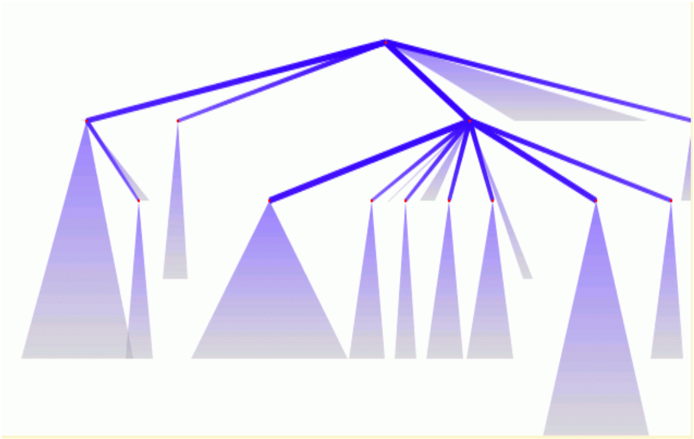

Fig. 2. A schematic view of the tree in Figure 1

Strahler numbers have proven to be a good measure of the branching structure of hierarchical networks (trees). They were also used as the basis of a method for producing realistic images of 2D trees in a paper from Viennot et al. [10]. The results presented there certainly confirm the potential that Strahler numbers bear as a means for describing graphical effects on trees.

In the case of trees, it is easy to check that greater values are attained by balanced trees. Numbers such as the Strahler number for trees are often referred to as *synthetic values*, because of their links with attribute grammars [4] and their use in combinatorial mathematics [6]. Other values can be computed using the same recursive scheme. For example, giving a value of 1 to every leaf of a tree and setting the value of a node to be the sum of the values of its children leads to a synthetic computation of the numbers of leaves for each subtree. This metric can be given more application-specific values, through the use of weights (see [3]). The weight of a node could represent, for example, the number of visits in the case of a web page or the size of a file in case of directory trees.

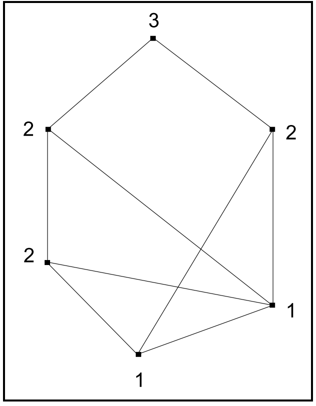

Fig. 3: Strahler for DAGs

The same computation scheme can be applied to any graph without cycles. Indeed, it is the absence of cycles in the structure that makes it possible to define a function depending on the set of *successors* of a node. A DAG has no cycles and provides an explicit direction to traverse its nodes. In a DAG, the subset of nodes having incoming degree zero are called *source nodes*. Similarly, the nodes having outgoing degree zero are called *sink nodes*. Obviously, a full traversal of a DAG may start from the source nodes and end in the sink nodes. The traversal of a DAG may also start from the sink nodes, depending on the desired results.

The Strahler metric can be easily generalized to DAGs by setting $S(b) = 1$ for every sink node, and by applying Eq. (1) to the set of *successors* of a node v. Figure 3 gives an example. Note: There is an implicit downward direction in Figures 3 and 4.

The Flow metric. The second metric we present is based on a water flow metaphor. A downward scan of the DAG emphasizes the distribution or flow of information from a node to its successors.

The DAG could be compared to a set of connected pipes through which water flows from top to bottom. We will call this metric the *Flow metric* and denote it by M.

Let $M(t) = 1$ for every source node t. Then compute values for every other node the following way: Divide the value at a node by the number of its successors to find its contribution to each of them. A node receiving a set of values from its ancestors sums them up. More precisely, the value $M(v)$ for a node v is obtained by summing contributions over the set of all its ancestors $a_1, \ldots, a_q$ ($q \geq 1$). That is,

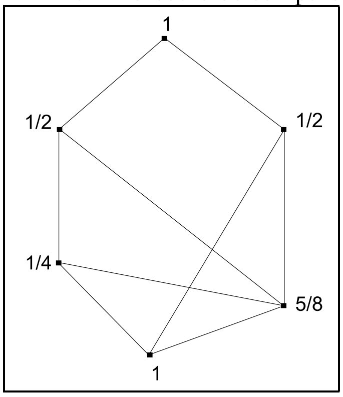

Fig. 4: Flow metric for DAGs

$$M(v) = \sum_j M(a_j)/\text{number of successors of } a_j \qquad (2)$$

The DAG in Figure 4 provides an example. Observe that values produced by this metric do not necessarily decrease (or increase) along a path from source to sink nodes. The value $M(v)$ at a node v evaluates the flow going through that node.

General framework. Of course, both the Strahler and Flow metrics produce values which are proportional to the node's importance. This should be kept in mind when designing any other metric to be applied to a DAG. The pattern used for Strahler, as well as the Flow metric, can easily be extended to a general computation scheme as follows: Suppose arbitrary values $K(b)$ are given to the sink nodes of a DAG. Set

$$K(v) = F(\,(K(k_1), \ldots, K(k_p)\,) \qquad (3)$$

for any other node v, where $k_1, \ldots, k_p$ are the successors of v and F is a function (or formula) depending on the values $K(k_1), \ldots, K(k_p)$. Hence, values are assigned to nodes of the DAG through an upward search, as with the Strahler metric. We could also define a function computing values through a downward search. In that case, we use a recurrence:

$$K'(v) = F(\,(K'(a_1), \ldots, K'(a_q)\,) \qquad (4)$$

where $a_1, \ldots, a_q$ are the ancestors of v, and assign starting values $K'(t)$ to source nodes of the DAG. This computation scheme was used to define the Flow metric. Observe that a dual Strahler metric could be defined by applying the opposite computation scheme using the same formula (Eq. (1)), but applying it to ancestors instead of successors. The same observation applies to the Flow metric, yielding a measure for an ”upward” flow of information or data.

The actual function to compute is in some sense application dependent. However, the choice or design of a metric should be strongly linked to a clear interpretation of its effect on the extraction process. From this point of view, metrics corresponding to well understood metaphors might have a wider range of uses and applications. This is the case for the Flow metric since it is supported by the water flow metaphor: the nodes in the skeleton are those through which much of the water flows. Also, weights can be used the same way they are used with Strahler to influence values of the nodes. Another possibility could be to give distinct starting values to source nodes of a DAG.

2.2 Skeleton extraction

Given a metric, the simplest approach to extract a skeleton is to collect the nodes with a value greater than or equal to a lower bound. We can compute a lower bound that extracts a specific percentage of the nodes, so we will express it in terms of the percentage.

Figure 5 shows the skeleton which results from selecting the nodes with Flow values in the top 30 %. The square bold-faced nodes are those belonging to the skeleton. The thicker arcs are those joining nodes in the skeleton. This example actually has no need for a skeletal view because it is not very complex but the smaller number of nodes allows us to more easily illustrate the essential concepts. For examples using a larger number of nodes, please see the color plates in the Appendix.

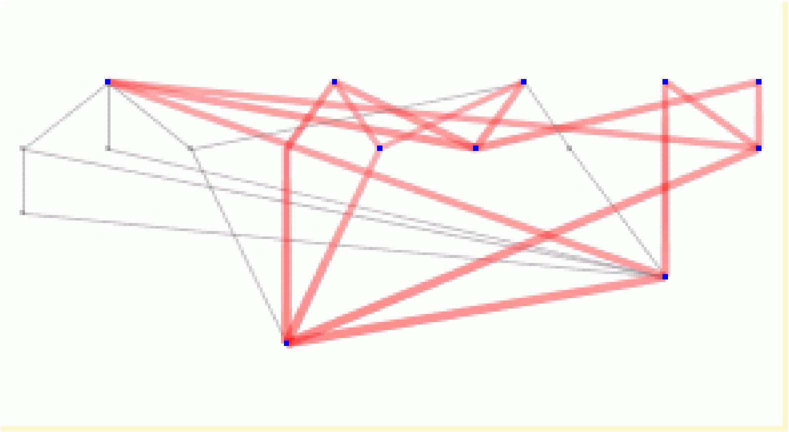

Fig. 5. The nodes selected based on Flow values in the top 30 %

All source nodes are part of the skeleton since they have an assigned values of 1. The water flow metaphor aids in understanding why a given node is present in the skeleton. This is obvious, for example, for the node with incoming degree 4 and outgoing degree 0 in the upper part of the skeleton; the contributions it collects from all sources nodes except one sum up to a value of 1.25, hence its presence in the skeleton. The node at the bottom of the right part on the skeleton bears the same value. Observe that it collects values from six different nodes, only two of which are part of the skeleton. The fact that the value 1.25 makes those nodes part of the skeleton depends on the set of values reached by all nodes in the graph and our choice to display the 30% top nodes.

2.3 Implementing a schematic view

The goal of the schematic view is to emphasize the ”backbone”, as produced by the skeleton extraction. The schematic view consists of two displayed parts: the skeleton itself, and the leftover nodes and edges.

Instead of lines, very long and thin trapezoids are used to display the skeleton edges. Trapezoids were chosen because they can have different widths at each end as an extra visual cue. The width of the trapezoids at the nodes are proportional to the metric values of the incident nodes. In other words, the sizes of the edges give an indication of the magnitute of the metrics at the incident nodes. Similarly, a continuous visual indication is provided by colour. Skeleton edges and nodes are drawn using a different hue than the leftover nodes (e.g. red on the colour plates in the Appendix). As a further visual cue, the saturation component of the colour along each edge is interpolated from the metric values at the source and destination nodes.

For DAGs, the leftover nodes and edges are simply drawn using a low-contrast hue (light gray on the Appendix colour plates). For trees, the monotonicity of the metrics, as well as the simpler structure of trees, allows for an alternative representation: triangles are used to replace the leftover nodes and edges. The size of the triangle image is proportional to the subtree being represented (see Figure 2). A continuous colour transition similar to the scheme for the skeleton edges is also used on the triangles. The top of the triangles have a saturation proportional to the node’s metric value, and the triangle gradually changes colour and saturation toward a shade of the background color. Of course, more complicated representations than triangles could be used for the subtrees such as the type of images used in the Aggregate TreeMaps of Chuah [1]).

In the cases of both trees and DAGs, the use of Alpha blending has also been an effective aid for both trees and DAGs. The transparency provided by alpha blending ensures that the intersections of edges and triangles do not interfere with the clarity of the figure.

3 Metric combination

Any good metric should emphasize a specific aspect of the DAG. *Combining* different metrics into new ones is a way to capture multiple aspects of the graph; some examples will be presented in this section.

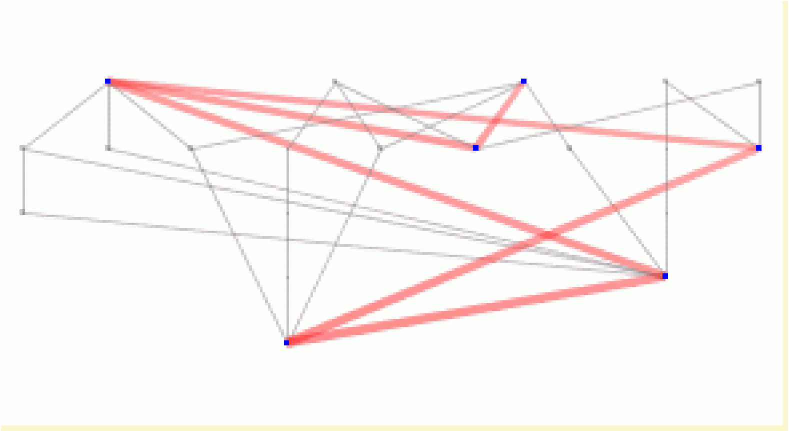

Fig. 6. Skeleton of a DAG based on a combination of Strahler and Flow metrics

Combining Strahler and Flow metrics. We use an example to illustrate how a combination of metrics can be achieved. When looking at the skeleton in Figure 5, one might object to the fact that *all* the source nodes were selected, i.e. that the Flow metric does not allow us to distinguish among the source nodes. Indeed, one may want to use a metric reflecting the fact that a subgraph, starting at a specific source, is more complex than another. Using our water flow metaphor, the metric should provide pipes with different diameters, depending on the complexity of the corresponding subgraph. The Strahler value of a node, which measures the structural complexity of its subgraph, is then a good candidate for measuring this complexity. This, in combination with the Flow metrics, may be used to define the desired new metric. The detailed definition of the new metric is as follows.

The Flow metric is modified so that the node receives a value from its ancestor proportional to its Strahler value. Denote by $\mu(v)$ the sum of the Strahler values of the children of a given node v. That is, $\mu(v) = \sum_j S(k_j)$, where the sum extends over the set of successors $k_1, \dots, k_p$ of v. The new metric is then defined by $P(t) = 1$ for all source nodes t and by the equation:

$$P(v) = \sum_j P(a_j) \cdot \frac{S(v)}{\mu(a_j)}$$

for all other nodes, where the sum extends over the set of ancestors $a_1, \dots, a_q$ of v. That is, the value $P(v)$ is obtained by summing contributions obtained from ancestors of v. A specific ancestor will give its children a part of its own value proportional to their Strahler values.

The skeleton based on this modified Flow scheme is shown in Figure 6. It is extracted from the same DAG as in Figure 5 using identical threshold values. Notice how this new computation scheme sorts the source nodes to extract only those playing a more important role in the whole graph (or network of pipes).

Combining directions. A further combination of metrics can be achieved if their directions are also taken into consideration. Indeed, the choice between Eq. (3) or Eq. (4) for the computation scheme priviledges a direction. If both a metric and its ”dual” are used on the same graph, each node is assigned two different values, reflecting directional measures. These values can then be combined (for example, by taking their average value), thereby yielding a new metric again. This ”average” metric reflects both the ”upward” and the ”downward” characteristics of the DAG relative to the metric.

As a specific example, the modified Flow metric of the preceding section has its dual metric, too. This dual metric uses the dual Flow metric and the *dual Strahler values*. Finally, the two directional Flow metric can be combined into an average Flow metric. This metric has been used to obtain the color plates in the Appendix.

4 Conclusions and further research

Our skeleton extraction methodology can be applied to any tree or DAG without using domain-specific knowledge, i.e. the semantic information usually associated with nodes or edges in a graph visualization application. However, it is possible to add domain-specific weights to the metric in order to sift the nodes for features of interest. In this way, it is possible to tailor a metric in order to implement a search. We have discussed the first type of metric which extracts interesting features from the graph relations inherent to the data, resulting in a structural view.

In our java application, skeletal views play an important role as navigational aids, complementing techniques such as zoom, pan, and fish-eye views. Although this is the only application of skeletal views which we have discussed, there are others. For example, we have created thumbnail images with skeletal views of a DAG. These thumbnails can then be used as the representation of a folded subtree. Another well known application of thumbnails is as a bird’s-eye view of the graph with an indication of the current viewing location.

The primary goal of our future research is to extend the skeleton idea to more general graphs. The definition of metrics for such graphs may be a significant problem because convergence must be ensured. Other metrics for DAGs, as well as other techniques to display the skeleton and the leftover nodes should be explored. A more elaborate usability study on the utility of skeletal views is also a possible future activity.

5 Acknowledgements

Part of this work has been funded by the Franco-Dutch research cooperation programme ”van Gogh”. We are also grateful to Bèhr de Ruiter (CWI), who developed an interactive interface for tree visualization, including zoom, pan, and fish-eye. Extending this framework to DAGs, instead of developing a completely new framework, allowed us to concentrate on the research issues in a more timely manner.

References

1. Chuah M.C. Dynamic Aggregation with Circular Visual Designs. In Wills G. and Dill J., editors, *Proceedings of the IEEE Symposium on Information Visualization (InfoViz '98), Los Alamitos*, pages 35 – 43. IEEE CS Press, 1998.
2. Flajolet P. and Prodinger H. Register allocation for unary-binary trees. *SIAM Journal. of Computing*, 15:629 – 640, 1986.
3. Herman I., Delest M., and Melançon G. Tree Visualization and Navigation Clues for Information Visualization. *Computer Graphics Forum*, 17(2):153 – 166, 1998.
4. Knuth D.E. Semantics of context-free languages. *Math. Systems Theory*, 2:127–145, 1968.
5. Lamping J., Rao R., and Pirolli P. A focus+context technique based on hyperbolic geometry for viewing large hierarchies. In *ACM CHI'95*. ACM Press, 1995.
6. Delest M. Algebraic languages: a bridge between combinatorics and computer science. In *Actes SFCA '94*, volume 24 of *DIMACS*, pages 71–87. American Mathematical Society, 1994.
7. Sarkar M and Brown M.H. Graphical fisheye views. *Communication of the ACM*, 37(12):73–84, 1994.
8. Schaffer D., Zuo Z., Greenberg S., Bartram L., Dill J., Dubs S., and Roseman M. Navigating hierarchically clustered networks through fisheye and full–zoom methods. *ACM Transactions on Computer-Human Interaction*, 3(2), 1996.
9. Keahey T.A. The Generalized Detail-In-Context Problem. In Wills G. and Dill J., editors, *Proceedings of the IEEE Symposium on Information Visualization (InfoViz '98), Los Alamitos*. IEEE CS Press, 1998.
10. Viennot X.G., Eyrolles G., Janey N., and Arques D. Combinatorial analysis of ramified patterns and computer imagery of trees. *Computer Graphics (SIGGRAPH '89)*, 23:31 – 40, 1989.

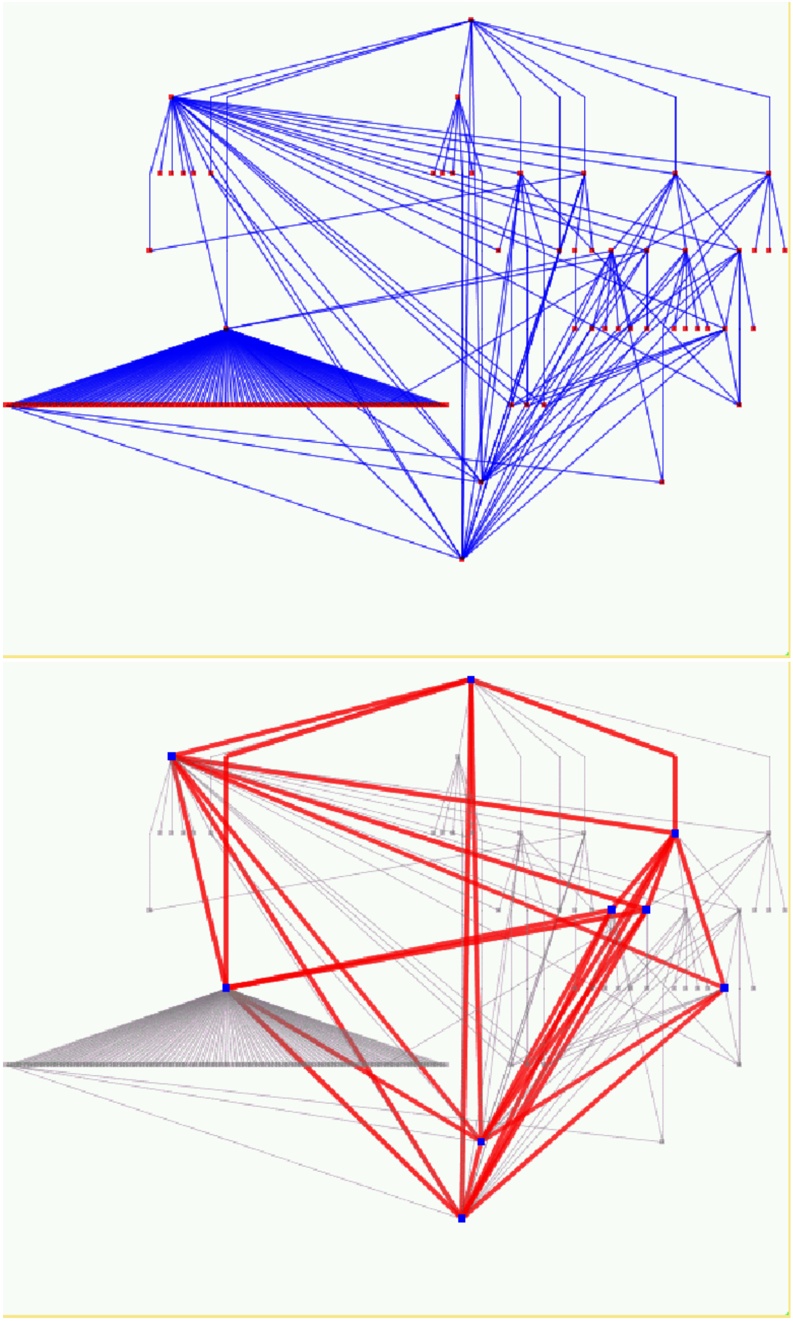

Fig. 7. Underlying DAG extracted from a web site (approx. 200 nodes) and skeleton based on average of modified Flow metric and its dual.