Labeling Schemes for Flow and Connectivity

Distributed Computing Seminar, WS 2003/04
ETH Zurich

Julio Pérez
<jperez@student.ethz.ch>

First, I will give a short summary of the topics and algorithms covered in the paper "Labeling Schemes for Flow and Connectivity" written by Michal Katz, Nir A. Katz, Amos Korman and David Peleg. Afterwards I'd like to critically evaluate the achievements of the paper.

1 Summary

The goal of a network representation is to cheaply store useful information about the network. The studied paper deals with network representation methods based on assigning informative labels to the vertices of the network, such that it is possible to infer information about any two labels directly from their labels.

A *labeling scheme* is composed of two components: A *marker* and a *decoder* algorithm. Given a graph G, the marker selects a *label* assignment L for G, where L is a function which assigns a label to each node of G. The decoder on the other hand takes as input a set of labels (assigned by the marker) and returns a value.

Given a family $\hat{G}$ of weighted graphs, an *f-labeling* scheme for $\hat{G}$ is a marker-decoder pair (M_f, D_f) with following properties: Consider $G \in \hat{G}$ and let $L = M_f(G)$ be the vertex labeling assigned by the marker M_f to G. Then for any set of vertices $W = \{ v_1, ..., v_k \}$ in G, the value returned by the decoder D_f on the set of labels $L'(W) = \{ L(v) \mid v \in W \}$ satisfies $D(L'(W)) = f(W)$. For our purposes f will be the flow or the vertex-connectivity function.

The first part of the paper deals with flow labeling schemes (i.e. f is the flow function) for general graphs. The basic idea is that, for a graph $G \in \hat{G}$, where $\hat{G}$ denotes the family of undirected capacitated connected n-vertex graphs with maximum integral capacity $\hat{w}$, the relation $R_k = \{ (x,y) \mid x,y \in V, flow(x,y) \geq k \}$ is an equivalence relation. Each R_k induces a collection of equivalence classes. That's why it's possible to construct a tree T_G out of G with the k^{th} level of T_G corresponding to the relation R_k. Without going to much into detail, the flow labeling scheme makes use of a *separation level labeling scheme*[1], which can be applied to the tree T_G corresponding to a graph $G \in \hat{G}$. It is stated that the resulting labels have size $O(\log n * \log\hat{w} + \log^2 n)$.

[1] See [2] for more details

The second part of the paper concentrates on *vertex-connectivity labeling schemes*. Concretely four such labeling schemes are introduced. The 1-connectivity labeling scheme is very simple and doesn't make use of any special trick. The other labeling schemes (2-, 3- and k-connectivity) are all based on the same idea, which consists of decomposing a graph into two simpler ones (actually in the case of 2-connectivity a graph is composed into a forest and graph) and then finding a labeling scheme which takes advantage of that decomposition.

2 Critical evaluation

Obviously it's quite irritating to understand a paper if there are some mistakes in it. Especially the definition of flow given in the paper would lead to a disaster. The definition is as follows:

- The maximum flow in a path $p = (e_1, ..., e_m)$ is the maximum value that does not exceed the capacity of any edge e_i in p, i.e. $flow(p) = \min_{1 \leq i \leq m}\{ w(e_i) \}$, where w is the capacity function
- A set of paths P in G is edge-disjoint if each edge $e \in E$ appears in no more than one path $p \in P$
- The maximum flow in a set P of edge-disjoint paths is $flow(P) = \sum_{p \in P} flow(p)$
- *flow(u,v)* = $\max_{P \in P_{u,v}}\{ flow(P) \}$, where $P_{u,v}$ is the collection of all sets P of edge-disjoint paths between u and v

Unfortunately this definition does not satisfy that R_k (see summary) is an equivalence relation, since the transitivity property is not met. It should not be required that the paths in P have to be edge-disjoint. Instead the flow definition found in literature should be taken, which demands that for each edge $e \in p_i$, and $p_i \in P$, the ingressing flow aggregated over all $p_i \in P$ mustn't exceed w(e).

Furthermore the authors state that, assuming any edge of a graph G has a capacity lower or equal to $\hat{w}$, the depth d of the tree T_G satisfies $d \leq \hat{w}$. This is not true, since for a graph G having a node u which is connected to all the other nodes in G via edge-disjoint paths, each of them having a flow equal to $\hat{w}$, the depth d of T_G would actually exceed $\hat{w}$. There are some other minor mistakes throughout the paper which are not mentioned here.

Another point is that for flow, for example, there is not really presented a labeling scheme. The authors just mention that there exists a separation level labeling scheme which can be tuned to a flow labeling scheme. So the interested reader has to look up the separation level labeling scheme in a referenced paper ([2]). My personal opinion is that in such a case, it's almost better to leave the labeling scheme away and to not present it in the paper.

Clearly there is another thing that seemed quite annoying to me. Throughout the whole paper there are only three or four figures used to illustrate theorems or a definitions of terms. But particularly for graph algorithms, a good figure sometimes can work wonders. That's why I really would suggest to include more figures, which results in having to use fewer words to explain a particular theorem, for example.

An aspect I didn't think about, but which was pointed out by an attentive listener during my presentation of the paper, was that using the labeling schemes presented in the paper, it is possible to retrieve the maximum flow between two nodes, but you don't know which path actually provides this flow. So if node u wants to send data to node v, u can compute the maximum flow between itself and v, but u won't receive the information, in which direction to send the data, i.e. it will not know which path to send the data along. I wouldn't be surprised if the authors or some other researchers in this area presented a labeling scheme for that, basing it on the flow labeling scheme explained in this paper. It seems to be standard practice to develop one labeling scheme out of another. A good example is the just mentioned flow labeling scheme that is based on a separation level labeling scheme, which in turn is based on a distance labeling scheme.

References

[1] M. Katz, N. Katz, A. Korman, and D. Peleg,
Labeling schemes for flow and connectivity (extended abstract), SODA 2002

[2] David Peleg, Informative labeling schemes for graphs, in Proc. 25th Symp. on Mathematical Foundations of Computer Science, vol. LNCS-1893, Springer-Verlag, Aug. 2000, pp. 579-588