Identifying Websites with Flow Simulation

Pierre Senellart[1,2]

[1] École normale supérieure, 45 rue d'Ulm, F-75230 Paris Cedex 05, France
[2] INRIA Futurs, 10 rue Jacques Monod, F-91893 Orsay Cedex, France
pierre@senellart.com

Abstract. We present in this paper a method to discover the set of webpages contained in a logical website, based on the link structure of the Web graph. Such a method is useful in the context of Web archiving and website importance computation. To identify the boundaries of a website, we combine the use of an online version of the preflow-push algorithm, an algorithm for the maximum flow problem in traffic networks, and of the Markov CLuster (MCL) algorithm. The latter is used on a crawled portion of the Web graph in order to build a seed of initial webpages, a seed which is extended using the former. An experiment on a subsite of the INRIA Website is described.

1 Introduction

Though the notion of *website* is commonly understood, there is no simple formal definition of it. The most obvious idea would be to define a *logical website* as the set of pages hosted by a given server. Though correct for many websites, it does not reflect their intuitive notion since some may span over several servers, and servers may host various sites.

The problem of discovering the boundaries between logical websites occurs in the topic of automatic Web archiving, that some instituions (e.g. national libraries) want to perform [1]: once webpages are selected to be archived, what is the boundary of the corresponding websites? To be able to define websites could also lead to *website importance* computation: to devise a *SiteRank* for websites, as *PageRank* [2] is defined for webpages.

The method for website identification we present in this paper heavily relies on the (directed) graph structure of the Web, with webpages as nodes and hyperlinks as edges. The fundamental assumption is that webpages in the same website are much more connected between them than webpages from different websites. We use an adaptation and combination of two algorithms related to flow simulation in traffic networks: a *preflow-push* algorithm by Goldberg [3] which solves the maximum flow problem and the *Markov CLuster algorithm* (abbreviated as *MCL*) by van Dongen [4], a graph clustering algorithm. MCL is used to cluster a part of the Web in order to build *seeds* of websites which are extended to complete logical websites with the preflow-push algorithm. The techniques used are not based on the concept of webservers, domain names or other heuristics, like traditional website recognition methods, but on the link structure of the Web graph and, secondarily, on the global form of the URLs.

D. Lowe and M. Gaedke (Eds.): ICWE 2005, LNCS 3579, pp. 124–129, 2005.

We show in Section 2 how flow simulation and the maximum flow problem may be used to identify websites in the Web graph. We notice that in many cases, the seed of webpages the simulation starts from needs to be extended; we present thus a way to use MCL for that purpose. An experiment is presented in Section 3. Finally, we discuss related works in Section 4. **An extended version of this paper is available in** [5].

2 Website Identification Process

In this section, we present an algorithm to solve the *maximum flow/ minimum cut* problem, namely the preflow-push algorithm, and its online adaptation to the Web. This algorithm is applied to extend a seed of webpages into a complete logical website. Then we briefly introduce MCL, a graph clustering algorithm, which is used to grow the initial seed of webpages. We finally describe our complete website identification process.

Preflow-Push Algorithm. We assume that the reader is familiar with the *maximum flow/minimum cut* problem. An introduction to this topic can be found in [6]. Let $\mathcal{T} = (S, c, s, t)$ be a traffic network, where S is the set of nodes, $c : S^2 \to \mathbb{R}_+$ the capacity function, s the source node and t the sink node. The preflow-push algorithm is based on the notion of *preflow*: a *preflow* in $\mathcal{T}$ is a function $f : S^2 \to \mathbb{R}$ which satisfies:

(i) (*Symmetry*) $\forall (u, v) \in S^2, f(u, v) = -f(v, u)$
(ii) (*Capacity constraint*) $\forall (u, v) \in S^2, f(u, v) \leq c(u, v)$
(iii) (*Relaxed flow conservation*) $\forall u \in S \setminus \{s, t\}, \sum_{v \in S} f(u, v) \leq 0$

This definition means that, in a preflow, a node u can have some *overflow* $o(u) = -\sum_{v \in S} f(u, v)$: it can receive more from the nodes it is pointed by than it sends to the nodes it points to. The preflow-push algorithm, as well as other algorithms working with preflows, maintains at each step a preflow in $\mathcal{T}$, converging finally toward a flow in $\mathcal{T}$ which is maximal, thus solving the *maximum flow/minimum cut* problem.

All nodes are assigned a height (0 in the beginning for all nodes except the source). At each iteration, the preflow is *pushed* from a node with overflow to a lower node. If there are no lower nodes to unload a node with overflow, this node is *raised*. The algorithm ends when there are no nodes with overflow any longer. This algorithm can be demonstrated to converge toward a maximum flow in $\mathcal{T}$ (cf [3]), with a complexity of $O(|S|^2|A|)$ ($|A|$ is the number of edges with non-zero capacity), whatever the strategy for selecting nodes with overflow.

Adaptation to the Web. The fundamental assumption of website identification based on the graph structure of the Web is that webpages in the same website are much more connected between them than webpages from different websites. If the Web is seen as a traffic network in which some fluid flows from a set of source nodes in the same website, the bottleneck of the flow should not be inside the

website, where there are many internal connections (and thus, a large capacity), but between the website and the rest of the Web, where the connections are much more sparse.

The idea behind using flow simulation to identify websites is that a clear cut should be visible between a "source of seed webpages" of a site and a "sink for the remaining part of the Web", a cut which would match the borders of the website. This cut is computed as a minimum cut in a traffic network whose underlying graph is the Web graph. More formally, let *Seed* be a set of seed webpages, characteristic of the website we would like to compute the borders of, and *sim* a similarity function over webpages. We consider the traffic network $\mathcal{T}_{Seed} = (S, c, s, t)$ where:

- S is the set of webpages in the World Wide Web, along with two virtual nodes s (a virtual source) and t (a virtual sink).
- (i) For all $(u, v) \in (S \setminus \{s, t\})^2$, $c(u, v) = sim(u, v)$ if there is a link from u to v, $c(u, v) = 0$ otherwise.
  (ii) For all $u \in Seed$, $c(s, u) = +\infty$
  (iii) For all $u \in S \setminus \{s, t\}$, $c(u, t) = \varepsilon$ ($\varepsilon << 1$)

The choice of the similarity function is important. The most simple choice would be to use a constant function. In this case, however, a cut separating the seed webpages from the rest would be most likely minimal. We chose to use a function of the edit distance between the URLs of the webpages: even if a website span over several webservers, the URLs of the pages tend to look similar.

On-line Preflow-Push. Classical graph and network algorithms are *off-line*: they require that the entire matrix is stored, so that computations can be made on it. In the context of the Web, *on-line* algorithms are more interesting. An on-line algorithm on the Web graph is an algorithm which does not require the storage of the entire matrix of the graph, and in which computations are made progressively, at the same time webpages are crawled. The preflow-push algorithm can be made on-line in a straightforward way: webpages with overflow are progressively crawled and dealt with (pushed or raised). Several strategies adapted to crawling can be chosen. We decided to use a greedy one: the node with maximum overflow is selected at each step.

Seed Extension with Graph Clustering. Applying this version of the preflow-push algorithm on a seed of characteristic webpages of a website gives quite good results on small or medium-sized, well-organized, websites. When the website is very large or not well organized, however, the algorithm only retrieves a small proportion of the webpages of the real website (cf Table 1). The problem is that a small seed is not sufficient to discover the entire website.

MCL is a graph clustering algorithm by van Dongen [4]: given an undirected graph as input, MCL will output a set of densely connected *clusters* of nodes. Based on an alternation of *expansion* and *inflation* of the graph matrix, MCL does not require any strong condition on the graph. The reader is advised to

refer to [4] or [5] for a more detailed description. MCL is used to extend the seed the on-line preflow-push starts from. The entired process is shown below.

Process for Identifying a Website W

1. Find a superset S of a large part of W (S may be, for instance, the set of webpages hosted by the main webserver for W); crawl and build the corresponding subgraph G_S of the Web graph.
2. Cluster G_S using MCL on the underlying undirected graph G'_S (there is an edge (u, v) in G'_S if and only if either (u, v) or (v, u) is in G_S).
3. Find the obtained cluster K which is the most relevant to W. It may be identified by finding the cluster which contains the largest number of URLs containing some given keyword.
4. Use the on-line preflow-push algorithm with K as a seed. The resulting set of pages is the logical website for W found by the process.

3 Experiment

Our main experiment was on the website of our research team, GEMO (other experiments are described in [5]). Its entry point is `http://www-rocq.inria.fr/verso/` (VERSO is the former name for GEMO) and it spans over several webservers, of the `inria.fr` domain and of other domains.

A large part of the webpages hosted on webservers of the `inria.fr` domain was crawled. Following the process described in Section 2, a MCL clustering was performed on the underlying undirected graph. The most relevant cluster was identified as the one with the largest number of URLs containing "verso". Finally, flow simulation with preflow-push gave the resulting logical GEMO website.

Table 1 shows some data about the website found by the algorithm, along with results of other simpler methods[1]:

- **Flow Simulation**: direct on-line preflow-push, starting from the website entry page.
- **MCL**: clusters from MCL, without flow simulation
- **MCL + Flow Simulation**: process described above
- `http://www-rocq.inria.fr/verso/*`: "naive" recursive crawl of the hierarchy of URLs starting from the website entry page.

As noted in Section 2, flow simulation alone retrieves a very small portion of the relevant webpages, whereas MCL effectively retrieves many more webpages, which are nearly all relevant (that is, the precision is very high). The recall for MCL clusters is still low, however; this is why the online preflow-push is applied afterwards. The complete process gives a good precision (over 90%) and especially gives a high recall, much higher than all the other methods. This shows the

[1] Precision and recall are defined as follows: the precision of a set of webpages W is the ratio of relevant webpages in W over the size of W; the recall of a set of webpages W is the ratio of relevant webpages in W over the total number of relevant webpages

Table 1. GEMO website, according to different methods

	Number of Pages	Precision	Recall
Flow Simulation	8	87.5%	1.3%
MCL	320	99.7%	33.0%
MCL + Flow Simulation	788	90.4%	86.4%
`http://www-rocq.inria.fr/verso/*`	221	100%	44.4%

interest of the combination of flow simulation and graph clustering techniques, over each technique alone. The naive technique naturally has a perfect precision but a rather low recall: there is indeed a need for more elaborate website identification methods, such as the one we used.

The results of our experiment on the GEMO cluster are thus very satisfactory. It is to be noted, though, that this does not represent the relative performance of the different techniques on every website. On smaller or more organized ones, the online preflow-push algorithm alone may be sufficient. On many websites, the naive recursive crawl of the hierarchy of URLs may even be perfect. Still, a large part of the Web is composed of not-so-well organized websites, spanning over several webservers, like the GEMO website.

4 Related Work

In most works where the notion of website appears, it is taken to be the pages hosted by a given webserver, or a lexical hierarchy of URLs (e.g. the set of URLs that share a common prefix), in addition to heuristics such as the recognition of `/~user/` part in an URL. Links between pages are usually only taken into account in an elementary way, such as in [7] where *clan graphs* are introduced to find closely connected pages. In that paper, websites are still assumed to be on a single webserver and much importance is given in the form of the URLs. In [8], Mathieu proposes a way to partition the Web by using the fact that the matrix of the Web graph in which URLs are lexically ordered is nearly block diagonal. Each block seems to correspond to a logical website, heavily connected inside and sparsely connected with other webpages.

The maximum flow problem in traffic networks is a classical and much studied algorithmic problem. Karzanov developed the notion of *preflow* [9] and Goldberg invented the preflow-push algorithm [3]. Flake et al [10] use a modified version of the preflow-push algorithm on the Web graph in a similar way as we do, for identifying Web communities. Beside the purpose, our approach differs in the on-line adaptation of the preflow-push algorithm, in the choice of the capacity of the edges and in the use of an extended seed.

5 Conclusion

We presented in this paper a website identification process, based on a combination of flow simulation and graph clustering. The preflow-push algorithm,

which solves the maximum flow problem in a traffic network, was adapted to the case of the World Wide Web. Logical websites are discovered by computing the minimum cut between a set of seed webpages and the rest of the Web, a seed which is computed using the Markov CLuster algorithm.

The first obvious perspective on this topic would be to improve the performance of the process, both in its execution time and in the quality of its results. Currently, the graph clustering needs to be computed on an off-line, crawled, subgraph of the Web, which can take a few days for a large graph, in order not to overload the corresponding webservers. It would thus be very useful to be able to realize an on-line computation of MCL. The adaptation is not obvious, especially because of the behavior of the inflation operator, which cannot be easily expressed in terms of classical linear algebra operators. Other improvements could be made on the online preflow-push algorithm, in particular with the choice of an efficient crawling strategy. Finally, a semi-automatic method, with the possibility of splitting and merging selected MCL clusters, would allow a more precise selection of the website to identify.

We would like to acknowledge Serge Abiteboul and Grégory Cobéna for their advice on this research topic.

References

1. Abiteboul, S., Cobéna, G., Massanes, J., Sadrati, G.: A first experience in archiving the French Web. In: Proceedings of the European Conference on Digital Libraries. (2002)
2. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
3. Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum-flow problem. Journal of the ACM **35** (1988) 921–940
4. van Dongen, S.M.: Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht (2000)
5. Senellart, P.: Identifying websites with flow simulation. Technical Report 387, Gemo, INRIA Futurs, Orsay, France (2005) `ftp://ftp.inria.fr/INRIA/Projects/gemo/gemo/GemoReport-387.ps.gz`.
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. The MIT Electrical Engineering and Computer Science Series. The MIT Press / McGraw-Hill Book Company (1990)
7. Terveen, L., Hill, W., Amento, B.: Constructing, organizing, and visualizing collections of topically related Web resources. ACM Transactions on Computer-Human Interaction **6** (1999) 67–94
8. Mathieu, F.: Mesures d'Importance à la PageRank. PhD thesis, Université Montpellier II (2004)
9. Karzanov, A.V.: Determing the maximal flow in a network by the method of preflows. Soviet Mathematics Doklady **15** (1974) 434–437
10. Flake, G., Lawrence, S., Giles, C.L.: Efficient identification of Web communities. In: Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA (2000) 150–160