G-DI: a Graph Decontamination Iterator for the Web

[Extended Abstract]

Daniel S. F. Alves
Department of Computer Science, CCMN
Universidade Federal do Rio de Janerio
Rio de Janeiro, Brasil
daniel.fsalves@gmail.com

Vanessa C. F. Gonçalves
Systems Engineering and Computer Science Program, COPPE
Universidade Federal do Rio de Janerio
Rio de Janeiro, Brasil
vcarla@gmail.com

Priscila M. V. Lima
Department of Mathematics, ICE
Universidade Federal Rural do Rio de Janeiro
Seropédica, Brasil
priscilamvl@ufrrj.br

Nelson Maculan
Systems Engineering and Computer Science Program, COPPE
Universidade Federal do Rio de Janerio
Rio de Janeiro, Brasil
maculan@cos.ufrj.br

Felipe M. G. França
Systems Engineering and Computer Science Program, COPPE
Universidade Federal do Rio de Janerio
Rio de Janeiro, Brasil
felipe@cos.ufrj.br

ABSTRACT

The Web, although of great importance to contemporary life, is also object of considerable misuse, such as cybercrime and unwanted advertising. Among non ethical activities on the Web, there is the non authorized insertion of links in webpages, performed by Web spammers, in other to increase the visibility of a target webpage T, via the creation of *Web bubbles*, also called *link farms*. As this is usually done via the use of self-replicating agents, this problem can be seen as a contamination process and this work introduces an evolution of the *scheduling by edge reversal* -based distributed iterator, in which varying criteria are considered for the following parameters: (i) the number of contaminated neighboring webpages of a webpage, and; (ii) a *refractory period*, i.e., the amount of time a recently decontaminated webpage, still having contaminated webpages as neighbors, remains decontaminated. Both criteria are associated, respectively, to the resistance to infection, and to the time factor of the spreading of the contamination. Experimental results showing qualitative and quantitative results concerning the new distributed decontamination mechanism are presented.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

webgraph decontamination, link farms, scheduling by edge reversal, spreading of influence

1. INTRODUCTION

The massive scale of the Web tends to facilitate cybercrime and malicious alterations of its topology, such as the creation of *Web bubbles*, also called *link farms* [7], targeting the artificial increase a given webpage T. As such typically happens through an invasive insertion of links, performed by self-replicating agents, a contamination like process can be identified and the focus of our work is to tackle this problem. Applications that manipulate the Web structure must consider its massive scale. Besides being able to deal with a huge amount of data, it could be pondered that a centralized approach to the problem of looking for contaminated webpages (and decontaminating them) would not allow for a scalable solution. Therefore, distributed mechanisms performing Web decontamination, having in mind different and concurrent contamination scenarios, are considered here.

The notion of Web neighborhood herein assumed means that a webpage has a neighbor if the later possesses a link to the former. In fact, the Web can be modelled as a set of the so-called *webgraphs* [2][5][6]. Webpages of a webgraph correspond to its nodes and the set of hyperlinks between them to the arc set. The original graph decontamination problem proposed in [8] assumed a single contaminated neighbor as contamination criterium, and put focus on optimal solutions corresponding to the minimum number of agents needed for decontaminating any given graph of arbitrary topology. Luccio & Pagli's *webmarshals* were concerned with the distributed decontamination of circulant graphs, a typical *link* *farm* structure [7], while considering a minimum percentage of 50% of contaminated neighbors as contamination criterion.

A *scheduling by edge reversal* (SER) [1] based approach to the decontamination of arbitrarily connected graphs, having the maximum number of webmarshals instances (agents) as a major concern, was recently proposed in [3]. It was shown that it could achieve better results than the ones presented in [7], while being able to deal with arbitrary graph topologies. In this work we introduce G-DI, Graph Decontamination Iterator, an evolution of the SER-based distributed decontamination algorithm, in which varying criteria are considered for the following parameters: (i) the number of contaminated neighboring webpages of a webpage, and; (ii) a *refractory period*, i.e., the amount of time a recently decontaminated webpage, still having contaminated webpages as neighbors, remains decontaminated. Both criteria are associated, respectively, to the resistance to infection, and to the time factor of the spreading of the contamination. Experimental results showing qualitative and quantitative results concerning the new distributed decontamination mechanism are presented together with correlation to novel interesting applications in the understanding of the spreading of social influence in social networks [4].

2. G-DI: SELF-COORDINATED WEBMARSHALLS

2.1 SER-based webgraph decontamination

The following is how SER — *scheduling by edge reversal* — works. By creating an acyclic orientation over the edges of a non oriented graph G, we have a directed graph with a non-empty set of sink nodes. By reversing all edges of sink nodes, we have a new directed graph with a new non-empty set of sink nodes. Three important properties have been proved [1]:

- edge reversals result in a finite set of acyclic oriented graphs, called *period*;
- given an acyclic directed graph ω' that is the result of the reversal of directed graph ω, any node that is a sink in ω' have at least one neighbor that was a sink in ω;
- all G's nodes become sink nodes the same number of times inside a period.

SER-based decontamination starts by associating decontamination agents, i.e., webmarshalls, at all (and just at) sink nodes. After cleaning a node, a webmarshall reverse all edges and copies itself just into the new sink nodes belonging to its immediate neighborhood. While a node remains with a number of infected neighbors higher than a maximum tolerable, a webmarshall stays to protect the node. Webmarshall copies proceed in a similar fashion, until the entire graph is decontaminated.Given the above mentioned properties, we know that all nodes will eventually be cleaned in finite time.

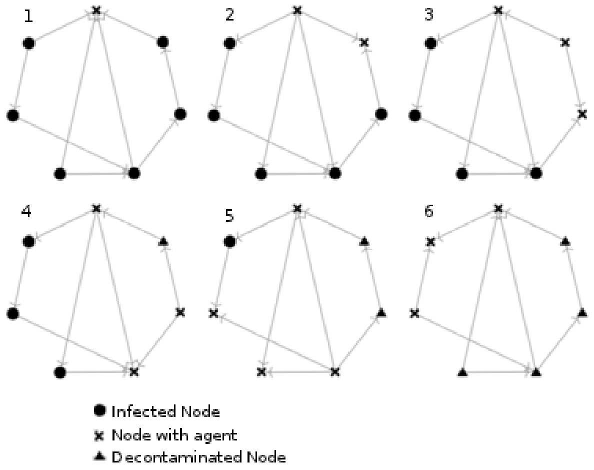

Figure 1: Example of decontamination by the SER approach on a non-circulant graph.

2.2 Generalized web decontamination

Up to this point, it was considered that the typical situation would be the one in which a node could be instantly recontaminated if the agent left it after decontamination. We will now consider that a delay associated with the spread of the infection exists.This way, it would be possible then to leave a node without agents for a certain period of time after which it would again be visited by another agent, if necessary, avoiding recontamination. We will also consider that a threshold (percentage) of infected neighbors a node can resist infection isn't the same in all situations. We study then the period a node can resist infection without the help of webmarshalls, which we call the *refractory period*, as well as the percentage of infected neighbors a node can resist infection.

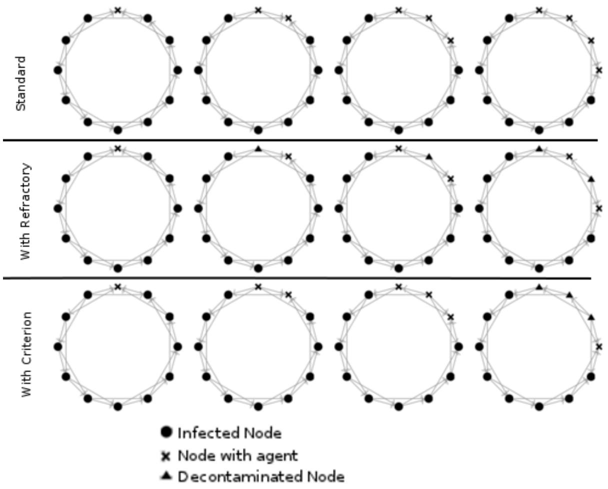

Figure 2: Comparison of the standard SER-based algorithm, the decontamination considering refractory and considering the infection criterion.

We explore then those two dimensions: (i) the refractory period and (ii) the infection criterion. These dimensions are related to the "speed" and "strength" of the infection, respectively. The refractory period is the measure of how long can a uninfected node withstand infection before being contaminated. It is related to the speed the infection propagates, so that agents can leave a node for some interval of time before the infection can reach the node they left.

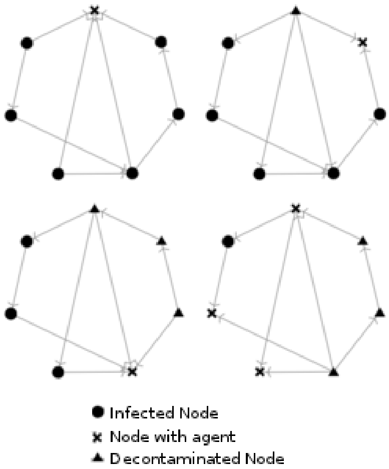

Figure 3: Example of decontamination by the SER approach on a non-circulant graph with a different refractory.

The infection criterion is the tolerance of a node to infected neighbors. It is related to the resistance of a node to infection, so that the more resistant a node is, the more immediate neighbors must be infected before it is at risk.

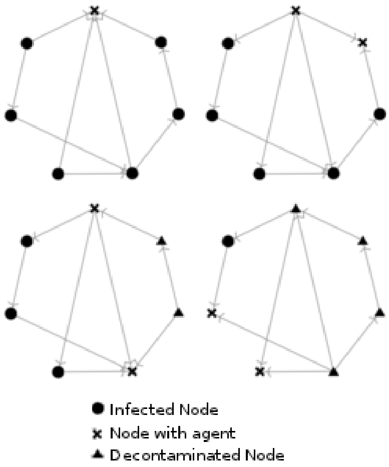

Figure 4: Example of decontamination by the SER approach on a non-circulant graph with a different infection criterion.

2.3 Experimental results

Although topology independent, the SER-based decontamination algorithm [3] presented better or equal figures than Luccio and Pagli's one [7] when just circulant graphs were considered. The generalized model considered here allows for the existence of conditions in which the number of agents needed for the target webgraph decontamination is much lower than the numbers presented in previous woks. The experimental results presented in Figure 5 shows an exploration of the two dimensions defined earlier, i.e., (i) infection criterion and (ii) refractory period, over a $C_{i,12}(1,2,3)$ circulant graph.

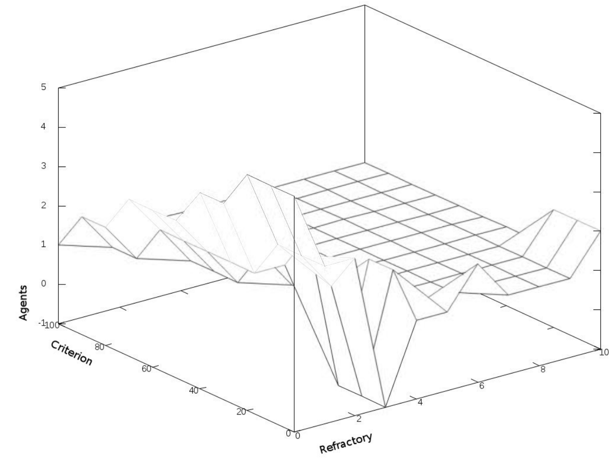

Figure 5: Number of agents needed to decontaminate a $C_{i,12}(1,2,3)$ circulant graph considering different values of infection criteria and refractory periods. A negative value in the number agents points out a configuration in which decontamination cannot occur.

Higher values in the refractory period dimension presents an impact on the number of agents needed, but also increase the number of steps needed to decontaminate the graph, possibly leading to long periodic attractors in which overall decontamination never occurs. The conditions that allows for such dynamics are not yet known and are subject of future research. It is also worth noticing that a non linear effect may exist on having higher refractory periods; such does not necessarily provides a decrease in the number of agents needed, since it might cause a "synchronization" between nodes in need for another webmarshall visit. On the other hand, the infection criterion behaves linearly with the number of agents needed.

3. CONCLUSIONS

The study of the dimensions of the webgraph decontamination problem allow for a better conceptual framework in terms of the number of webmarshalls needed. Is was demonstrated that a compromise between the length of the refractory period and the total decontamination time exists when looking for solutions involving small numbers of webmarshalls. It must also be noticed that although a particular but meaningful webgraph topology was chosen for the experimental exploration presented here, the proposed decontamination framework remains quite general. Among future work, the possibility of tackling the web decontamination problem via a different view, in which to keep the web safe of link farms by a certain decontamination percentage, would motivate the interest for having a relatively small number of webmarshalls running around the web. The exploration of the two decontamination dimensions showed the existence

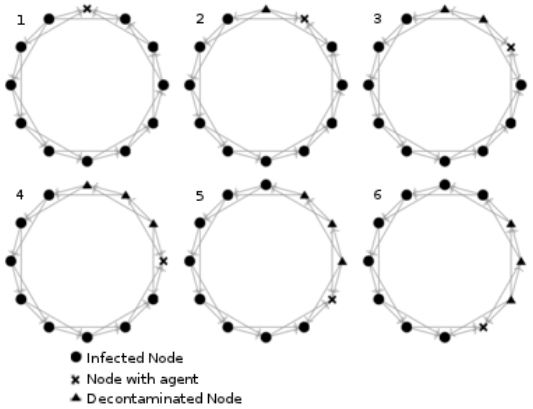

Figure 6: Formation of infinite cycle caused by poor refractory.

of long attractors, in which 100% decontamination never occurs. If a manageable percentage of contaminated web sites would be considered acceptable, the use of smaller numbers of agents would be plausible in the web.

4. ACKNOWLEDGMENTS

This work was partially supported by CNPq and FAPERJ, Brazil.

5. REFERENCES

[1] V. C. Barbosa and E. Gafni. Concurrency in heavily loaded neighborhood-constrained systems. *ACM Trans. Program. Lang. Syst.*, 11:562–584, October 1989.

[2] A. C. Gilbert. Compressing network graphs. In *LinkKDD*, 2004.

[3] V. C. F. Gonçalves, P. M. V. Lima, N. Maculan, and F. M. G. França. A distributed dynamics for webgraph decontamination. In *4th international conference on Leveraging applications of formal methods, verification, and validation*, ISoLA 2010, pages 462–472, Berlin, Heidelberg, 2010. Springer-Verlag.

[4] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[5] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: measurements, models, and methods. In *Proceedings of the 5th annual international conference on Computing and combinatorics*, COCOON'99, pages 1–17, Berlin, Heidelberg, 1999. Springer-Verlag.

[6] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, and E. Upfal. The web as a graph. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '00, pages 1–10, New York, NY, USA, 2000. ACM.

[7] F. Luccio and L. Pagli. Web marshals fighting curly link farms. In P. Crescenzi, G. Prencipe, and G. Pucci, editors, *Fun with Algorithms*, volume 4475 of *Lecture Notes in Computer Science*, pages 240–248. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-72914-3_21.

[8] M. Moscarini, R. Petreschi, and J. L. Szwarcfiter. On node searching and starlike graphs. In *Congressus Numerantium*, volume 131, pages 5–84, 1998.