

Self Regulating Mechanisms for Network Immunization

Luca Mancini, Alfredo Milani, Valentina Poggioni, and Andrea Chiancone

University of Perugia, Italy
{poggioni, milani, chiancone@dmi.unipg.it}
manci88@hotmail.it

Abstract. In this paper, we propose an immunization strategy based on distributed autonomous entities which self-regulate their diffusion in the network where they are deployed with a target coverage to be achieved. Entities move from node to node over the network, while collectively searching the nodes with the highest degree and they achieve the target coverage by reproducing themselves according to a self regulating reproduction strategy. Experiments show that the proposed approach produces a population which is able to self regulate in many widely accepted benchmarks while achieving different target coverage rates.

1 Introduction and Related Work

Effective and efficient immunization strategies have been proposed to contrast virus propagation and to prevent a virus from becoming a rapid epidemic in the network. Since the immunization of a node can be a very expensive process in many network environments, the goal of immunization strategies is usually reduced to immunize a small number of nodes i.e., those nodes which have the highest degree [4, 7, 10].

Several immunization strategies have been proposed in the last years, but most of them are based on centralized processing of static networks and cannot adapt well to large-scale, decentralized scale-free artificial networks [1, 3, 4, 8, 10], while the distributed strategies, including the acquaintance immunization [2] and the D-steps immunization [7], do not adequately address the issue of dynamic network.

In the recent, also the autonomy-oriented computing (AOC) approach has been efficiently applied to network immunization [6]. Although the AOC approach does not need to know the network topology, it should be given as input the number of important nodes which must be vaccinated, i.e. it should know the network size in order to reach a given immunization rate.

Based on the above motivation, we propose an AOC immunization strategy where the autonomous entities have the ability to reproduce and distributively

Proceedings of the 21st RCRA workshop on *Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion* (RCRA 2014).
Vienna, Austria, July 17–18, 2014.

self-regulate their population in order to reach a target network coverage rate, while optimizing the number of high degree nodes occupied. In the proposed approach each distributed autonomous entity estimates the percentage of occupation of the graph, on the base the history of entity's observation and from information found on its own local environment; the estimate is then used to decide whether or not, and how much, to reproduce itself.

2 AOC approach to Network Immunization

In the AOC approach to network immunization, a group of computational autonomous entities are dispatched into a decentralized network [5, 6]. These entities work together, to achieve a common goal, by communicating through their local environment: they decide autonomously to perform one of the available actions based on the information in their local environment.

The network is traditionally represented as a graph $G = (V, L)$ with N nodes (vertices) $V = \{v_1, v_2, \dots, v_N\}$ and links (edges) represented as pairs of nodes $L = \{< v_i, v_j > | 1 \leq i, j \leq N, i \neq j\}$.

In [4, 7, 10] it has been shown that the *targeted immunization strategy* i.e the strategy according to the problem of network immunization is translated into that of finding the k largest-degree nodes in the network, is one of the most efficient strategies. They introduce the concept of maximizing the *Sum of the Important node Degrees*, $SID = \sum_{v_i \in V_e} deg(v_i)$ where V_e is the set of vaccinated nodes and $deg(v_i)$ is the degree of the node v_i .

In [6] the authors formalize the concept of immunization strategy in a AOC perspective, and show how the AOC-based immunization is more efficient than other methods, in particular with respect the D-steps strategy [7].

Nevertheless a major assumption of the AOC-based strategy is that it has to take as input the (fixed) number of nodes to immunize, i.e. the number of entities to deploy on the network in order to obtain a suitable immunization rate.

Since immunization is a costly task, overestimating the number of needed entities can result in a waste of resources, which could be saved or, for instance, deployed in different networks. On the other hand, underestimating the number of nodes to immunize can lead to inefficient immunization and can make the network vulnerable and prone to virus attack. The problem of under/over estimation is particular relevant for dynamic networks where the size can grow or shrink over the time, and in the case of large online networks of unknown size where no central entity is available to provide the necessary data.

3 Self Regulating AOC

Our goal is to overcome this limitation by proposing an approach which extends the AOC-based algorithm [6] in which the entities can self-regulate their population in order to reach a given coverage rate of the network.

The basic idea is to extend the entity behaviours by a new action, called *reproduction*, that allows an existing entity to generate one or more new entities that will be deployed on the network. This behaviour, driven by a suitable reproduction strategy, can avoid the need of knowing the network structure and size which can be unknown and can dynamically change.

Definition 1 (Entity with reproduction). *An entity with reproduction e_R is a tuple $\langle id, node, target, lifecycle, observations, actions \rangle$ where*

- *id*: is the entity identifier
- *node*: stores information about the node where the entity resides (*nodeId*, *degree*). The degree is called entity utility $u_t(e)$,
- *target*: is the target coverage to reach; it represents the percentage of nodes in the network we want to be resided by an entity.
- *lifecycle*: is the maximum time steps for an entity to reside on a node
- *observations*: is a pair defined by *nObs*, that is the number of nodes that the entity have observed during the exploration, and *eObs* that is the number of other entities that the entity have met during the exploration, i.e. is the number of occupied nodes it observed.
- *actions*: is the set of possible actions each entity can perform.
- **RationalMove**: An entity moves to the highest-degree (not already occupied) position in its local environment.
- **RandomJump**: An entity moves randomly to a distant or long-range node.
- **Wait**: An entity stay in its position whenever it cannot find any better position.
- **Reproduce**: Under given conditions an entity can reproduce itself by generating one or more other entities.

The entities can communicate with each other through their local environment. The *local environment* of an entity is the set of direct and indirect neighbor nodes as specified in the following definition.

Definition 2 (Local Environment). *Let e an entity and v_i the node in which it resides. The local environment of e is defined as $E_l(e) = \{v_j, friend(v_j) \mid \langle v_i, v_j \rangle \in L\}$, where $friend(v_j)$ is a node such that:*

- *it is the second highest-degree node, not being resided by any other entities,*
- *it is known to the entity which has ever visited and/or encountered v_j during its search process.*

Local environment of an entity is not a static set of nodes because it can be modified by other entities. Due to this possibility the information of the important nodes on the network quickly reach every entities.

Definition 3 (Profit). *The profit $\delta_t(e)$ of the entity local environment $E_l(e)$ is defined as the degree of the highest degree node in the local environment of e at the time step t .*

Each entity chooses one of four possible actions *RationalMove*, *RandomJump*, *Wait* and *Reproduce* on the basis of the feedback received from its local environment. The choice is based on the estimation of the graph coverage rate and the comparison of the current entity utility with its *profit*.

If the estimated network coverage is less than the target coverage, the entity will reproduce itself. Then, one of the basic actions *RationalMove*, *RandomJump* or *Wait* is performed: if the profit is greater than the *entity utility*, then the entity performs a rational-move (to increase its utility), otherwise if its lifecycle is zero and it cannot stay more in the same node, it performs a random-jump to a distant or long-range node in order to avoid getting stuck in local optima.

Due to the decentralized nature of the algorithm, entities do not know the exact value of the coverage rate at a given time step, and they try to estimate it in order to regulate the reproduction. The more the coverage is approaching the target, the less the entities will reproduce themselves.

3.1 Estimation Phase

The estimation phase is the core issue of the self regulating mechanism and it plays a key role on the success of the strategy. Several techniques have been developed and tested and has been noted that too rough estimates, like the simple estimate by means the percentage of resided neighbors, do not work.

The technique proposed in this work is affected by three factors: the *local coverage*, the entity's *exploration history* and the other entities' *stamps* found in the *local environment*.

The *local coverage* takes into account the direct neighbors actually occupied by some entity, i.e. the network coverage in the area surrounding the entity. This value is combined with the *exploration history* in order to maintain statistics of the entity's past observations, in such a way that the impact of old observations is progressively fading off and replaced by the new ones. These observations are stored in the entity properties $nObs_t$, number of observed nodes, and $eObs_t$, number of entities met at the time step t , that are updated at every step according to the formulas:

$$nObs_t(e) = (1 - \alpha) \cdot u_t(e) + \alpha \cdot nObs_{t-1}(e) \quad (1)$$

$$eObs_t(e) = (1 - \beta) \cdot RN_t(e) + \beta \cdot eObs_{t-1}(e) \quad (2)$$

where $RN_t(e)$ is the number of direct neighbors of the entity e that are resided at the time step t , and $\alpha, \beta \in [0, 1]$ are parameters that regulate how much the values, computed in the precedent steps, affect the current estimate. Some preliminary tests have been run to determine which values is convenient to assign to α and β . The experimental results shown in this paper derive from runs with $\alpha = \beta = 90/190$.

$nObs_t(e)$ and $eObs_t(e)$ are also stored as *local stamps* s_nObs and s_eObs in the entity's node and its direct neighbours, in order to leave a trace to be used by other entities to implement a form of entity to entity communication. s_nObs and

s_eObs will overwrite the preexisting stamps in a node only if $nObs_t(e) > s_nObs$. The rationale behind this strategy is that more experienced entities, i.e. entities with a greater $nObs_t(e)$ are more trustful.

These *local stamps* are combined with the entity's actual and past observations to compute the graph coverage estimate according to the formula

$$localEst_t(e) = \frac{eObs_t(e) + \sum_{n \in N(e)} s_eObs_t(n)}{nObs_t(e) + \sum_{n \in N(e)} s_nObs_t(n)} \quad (3)$$

The entities compare this ratio with the *target* coverage value and if it is lower, then the entities decide to reproduce themselves according to the reproduction rules.

3.2 Reproduction rules

When an entity decides to reproduce itself, it has to determine: the number of entities to be generated, the location where a new entity will be placed and its initial state.

About the number of new generated entities, different strategies have been implemented and tested. We tested a strategy that fixes the branching factor as an input parameter of the program, but the experimental results did not appear promising in order to control the entities offspring.

So, it was necessary to design a self regulating mechanism that dynamically and distributively determine a local branching factor. The number of new entities to be generated, called *bf*, is defined as

$$bf_t(e) = k|localEst_t(e) - target| \quad (4)$$

where $k \in \mathbb{R}$ is called *reproduction factor*.

This means that in the first steps the number of entities will quickly increase, while, as the entities detect a more populated network, this growth will slow down, until it stops.

Moreover, several strategies for determining the *initial location* and *initial state* of a new entity have been implemented and tested. Results about these strategies have been shown in [9].

4 Experiments

In this section we present and discuss the results obtained with Self Regulating (SR) AOC immunization strategy on widely known and publicly available experimental data sets (<http://snap.stanford.edu/data/>). The eight considered networks cover different classes of real world and synthetic graphs and are described in Table 1.

The table presents an overview of the network details that we have used in our studies, where: $\langle K \rangle$ is the average degree of nodes, *diameter* is the greatest distance between any pair of vertices, *ACC* (*Average Cluster Coefficient*) is the

average of the local clustering coefficients of all the nodes and *APL* (*Average Path Length*) is the average number of steps along the shortest path for all possible pairs of network nodes.

Name	#Nodes	#Edges	$\langle K \rangle$	Diameter	ACC	Triangles	APL
email-Enron	33696	180811	10.732	13	0.605	725311	4.025
ego-Facebook	4039	88234	43.691	8	0.605	1612010	3.693
NET7	56276	315923	11.228	17	0.687	1472256	8.557
NET6	12741	26888	4.221	11	0.455	24732	3.631
wiki-Vote	7066	100736	28.513	7	0.209	100736	3.247
p2p-Gnutella	10876	39994	7.355	10	0.008	934	4.5
p2p-Gnutella08	6299	20776	6.597	9	0.0015	2383	4.64
NET9	4000	16733	8.366	11	0.079	17248	5.463

Table 1: Networks used in the experiments

All the experiment results are obtained as an average of 40 runs and are executed on a Macbook pro 2.3 GHz Intel Core i5 and 16 GB of ram. The experiments will be analyzed according to different metrics and dimensions:

- **Reproduction Rate**, defined as the percentage of entities that are reproducing themselves at a given time step t . It measures the ability of the SR-AOC strategy to progressively stop the reproduction. We expect that the number of reproducing entities will converge to zero as the coverage rate gets closer to the input target coverage.
- **Coverage Rate**, defined as the ratio between the number of entities in the graph at the step t and the number N of nodes in the graph. The ideal situation is a convergence to the *target coverage*. This measure strongly depends on the capacity of the system to give a good estimate of the global coverage using the information collected by the entities during their exploration.
- **Immunization rate**, defined as the ratio between the SID reached by the entities at a given time step t and the maximum SID reachable in the same step by that number of entities. It measures the ability of the algorithm to place the entities in the nodes with the highest degree thus maximizing the immunization effect. In the following we consider satisfactory results with a SID rate over 90%.

Fig. 1 presents, for two of the eight tested networks, four series of tests each on a different *target coverage*: for each test the reproduction rate and the immunization rate are plotted. It can be noted that, for these two tests, the reproduction rate quickly decrease below one percent, i.e. the entities reproduction eventually stops. The immunization rate is plotted together with the reproduction rate in order to show how, after the convergence to zero of the reproduction rate, in a very few steps the immunization rate reaches and exceeds the 90% threshold. Results for the other networks are very similar [9], that also point out the scalability of the proposed Self Regulating strategy.

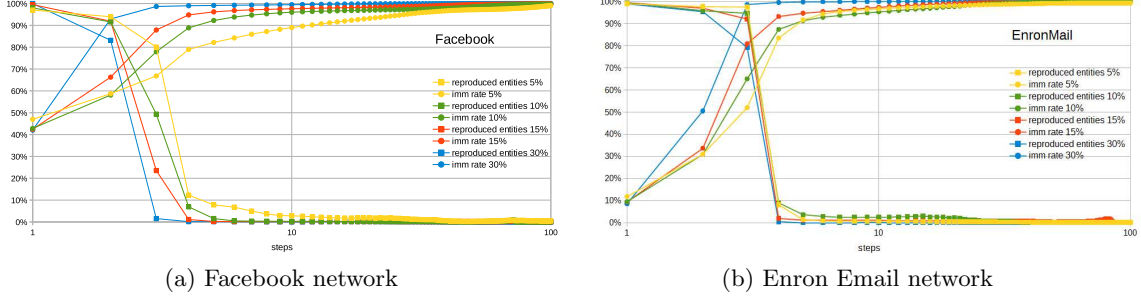


Fig. 1: Experiment results for entity reproduction rate compared with the immunization rate. Each graph shows results obtained for different targets (5%, 10%, 15% and 30%).

The results show that the main goal of building a mechanism in which the entities autonomously regulate their population, while moving on the network and maximizing at the same time the immunization rate, has been reached.

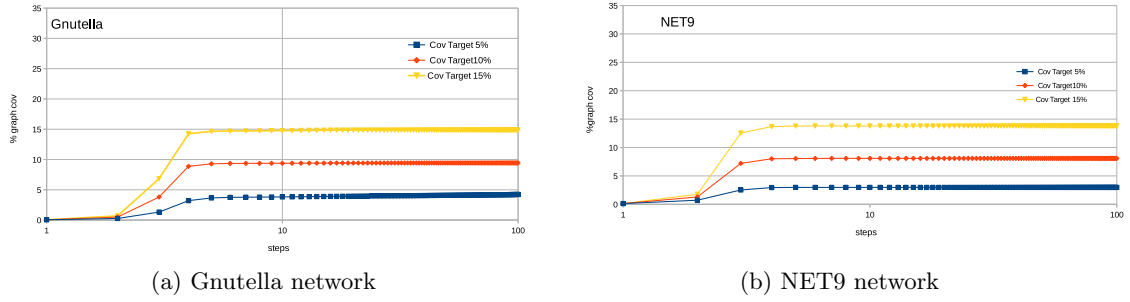


Fig. 2: Experiment results for coverage rate: Fig.2a shows how the coverage rate reaches the target coverage in a very few steps, while Fig.2b shows a case where the target is not reached due to an over-estimation of $localEst_t(e)$ by the entities.

Further analysis has been conducted about the precision of the proposed method, i.e. the ability to reach the target coverage and minimize the coverage over/under estimation error. Fig.2 focuses on the results of coverage rate for two of the eight tested networks with three different targets (5%, 10% and 15%). In these plots different behaviours can be observed: in the case of the Gnutella network (Fig.2a) the target is reached in a very few steps and the proposed strategy shows a high stability, while in the case of NET9 network (Fig.2b) the tests show convergence and high stability but the target coverage has not been reached: the actual reached coverage is respectively 3.5%, 8% and 13.7% for

the targets 5%, 10% and 15%. This behaviour has been analyzed and the main cause can be attributed to an over-estimation of the coverage $localEst_t(e)$ by the entities. The formula (3) is object of further investigation and we are currently studying different versions.

5 Conclusions and Future Work

A Self Regulating Autonomy Oriented (SR-AOC) immunization algorithm has been presented. The entities self regulate their population while exploring the graph and maximizing the immunization rate. Differently from other distributed immunization schemes previously proposed in the literature, the presented method allows to immunize networks of unknown size.

Experiments on widely accepted network datasets show that, when the entities population stabilizes, the immunization rate converges to the optimum. Tests on different networks confirm the scalability.

Further work is needed in order to improve the precision of the method. Ongoing work is investigating tuning methods for the reproduction phase in order to avoid the sharp gap observed on the population growth among two consecutive steps of the algorithm. In order to improve the reproduction rate convergence to zero it will be also considered the introduction of entity elimination action.

References

1. Y. Chen, G. Paul, S. Havlin, F. Liljeros, and H. E. Stanley. Finding a better immunization strategy. *Phys. Rev. Lett.*, 101:058701, Jul 2008.
2. R. Cohen, S. Havlin, and D. ben Avraham. Efficient immunization strategies for computer networks and populations. *Phys. Rev. Lett.*, 91:247901, Dec 2003.
3. Z. Dezső and A.-L. Barabási. Halting viruses in scale-free networks. *Phys. Rev. E*, 65:055103, May 2002.
4. L. K. Gallos, F. Liljeros, P. Argyrakis, A. Bunde, and S. Havlin. Improving immunization strategies. *Phys. Rev. E*, 75:045104, Apr 2007.
5. C. Gao, J. Liu, and N. Zhong. Network immunization and virus propagation in email networks: experimental evaluation and analysis. *Knowl. Inf. Syst.*, 27(2):253–279, 2011.
6. C. Gao, J. Liu, and N. Zhong. Network immunization with distributed autonomy-oriented entities. *IEEE Trans. Parallel Distrib. Syst.*, 22(7):1222–1229, 2011.
7. J. Gmez-Gardees, P. Echenique, and Y. Moreno. Immunization of real complex communication networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 49(2):259–264, 2006.
8. P. Holme and B. J. Kim. Vertex overload breakdown in evolving networks. *Phys. Rev. E*, 65:066109, Jun 2002.
9. L. Mancini. Self regulating mechanisms for network immunization. *MSc Thesis, University of Perugia*, 2014.
10. R. Pastor-Satorras and A. Vespignani. Immunization of complex networks. *Phys. Rev. E*, 65:036104, Feb 2002.
11. D. Watts and S. Strogatz. Collective dynamics of small-world networks,. *Nature*, 393:440–442, 1998.