Network Immunization with Distributed Autonomy-Oriented Entities

Chao Gao, Jiming Liu, Ning Zhong

Abstract—Many communication systems, e.g., Internet, can be modeled as complex networks. For such networks, immunization strategies are necessary for preventing malicious attacks or viruses being percolated from a node to its neighboring nodes following their connectivities. In recent years, various immunization strategies have been proposed and demonstrated, most of which rest on the assumptions that the strategies can be executed in a centralized manner and/or that the complex network at hand is reasonably stable (its topology will not change overtime). In other words, it would be difficult to apply them in a decentralized network environment, as often found in the real world. In this paper, we propose a decentralized and scalable immunization strategy based on a self-organized computing approach called autonomy-oriented computing (AOC) [1], [2]. In this strategy, autonomous behavior-based entities are deployed in a decentralized network, and are capable of collectively finding those nodes with high degrees of conductivities (i.e., those that can readily spread viruses). Through experiments involving both synthetic and real-world networks, we demonstrate that this strategy can effectively and efficiently locate highly-connected nodes in decentralized complex network environments of various topologies, and it is also scalable in handling large-scale decentralized networks. We have compared our strategy with some of the well-known strategies, including acquaintance and covering strategies on both synthetic and real-world networks.

Index Terms—Immunization strategy, complex networks, distributed search, autonomy-oriented computing (AOC), self-organization, positive feedback, scalable computing

1 Introduction

In real-world networks [3], [4], nodes represent individuals (e.g., computers, web pages, email-boxes) and edges represent their connecting or interaction relationships (e.g., network links, hyperlinks, and the relationships between two people in an email network). To some extent, the degree of a node reflects its importance in the network. This is because a node with a larger degree has more connections with others. In our present work, we call those highly-connected nodes *important* nodes in the network. With the growth of computer network-based applications, computer viruses have extended their propagations from physical networks to logical networks. For instance, the threats of Internet worms against network security have become increasingly serious. Thus, it is extremely important for us to design an effective and efficient mechanism for restraining virus propagation. Presently, one of the more popular methods is network immunization. Here, by immunization it is meant that some nodes in the network are immunized and hence will not be infected by any virus or worm. In view of the costs associated with immunization doses/operations in many network environments, the core of immunization becomes how to immunize a small or minimum number of important nodes in order to prevent a virus from becoming a rapid epidemic in the network.

In the past, several immunization strategies have been proposed, ranging from global strategies to local strategies, as described in [5], [6], [7], [8], [9], [10], [11], [12], [13]. These strategies are designed for centralized processing and dealing with static networks. Most of them cannot adapt well to large-scale, decentralized scale-free networks [3], [14], [15]. In order to restrain a virus epidemic in scale-free networks, the traditional *random immunization* strategy [16], [17] needs to protect 80% of the nodes [5], whereas the *targeted immunization* strategy [6], [7] requires to know the global topology of the networks. In this regard, two limitations can be noted: (1) it would be impossible to sort them in a large-scale network, simply because the computational complexity ($O(N^2)$) is too large; (2) it would be unrealistic to obtain the global topology for a decentralized and/or dynamically evolving network.

Large-scale decentralized and/or dynamically-evolving networks, as often found in the real world (such as Ad Hoc or WAN networks), have the characteristics of being extremely robust to random failures but highly vulnerable to malicious attacks. For instance, they may be efficient for file-sharing, but prone to computer virus or worm propagations [18]. In view of such threats, it would be critical to develop and deploy effective distributed protection strategies. In recent years, some efforts have been made in developing distributed immunization strategies, as an attempt to achieve the efficiency of the targeted immunization in a decentralized and/or dynamic network. Some

Chao Gao is a PhD student at the International WIC Institute, and Beijing Key Laboratory of Multimedia and Intelligent Software, Beijing University of Technology, 100124, Beijing.
Jiming Liu (corresponding author) is with the Department of Computer Science, Hong Kong Baptist University, and affiliated with the International WIC Institute, Beijing University of Technology, 100124, Beijing; jiming@comp.hkbu.edu.hk
Ning Zhong is with the Department of Life Science and Informatics, Maebashi Institute of Technology, Japan, and affiliated with the International WIC Institute, Beijing University of Technology, 100124, Beijing.

typical distributed strategies include the *acquaintance immunization* [9] and the *D-steps immunization* (or covering immunization) [12], [13]. Their aims are to vaccinate the highly-connected important nodes, and thus cut the epidemic path without the global topological knowledge. So far, it remains a challenge for them to deal with some real-world networks (e.g., P2P networks, or networks containing cliques or communities) with decentralized and/or dynamic structures, since they randomly select the initial seed nodes and thereafter only vaccinate their neighbors with the maximum degree within a certain range. If the initial seed nodes are located inside one community, this strategy will not be able to effectively protect nodes inside other communities. In other words, due to their fixed selection rules, the traditional distributed immunization strategies are sensitive to their initial settings or parameters, i.e., the initial positions of seed nodes and the length of steps. Thus, it cannot adapt to dynamic and/or community networks. In summary, the followings are some unavoidable limitations:

- They ignore the structure of a network, e.g., the network with different communities;
- They ignore those nodes with the second highest degree;
- Although the D-steps immunization is suited to distributed networks, it is computationally too costly. At the same time, it is difficult or impractical to know other indirect neighbors, i.e., one's neighbor's neighbors, in the real-world situations.

Based on the above motivations, in our present work, we propose an autonomy-oriented computing (AOC) approach [1], [2], in which a group of computational entities are dispatched into a decentralized network. They autonomously work with each other and update their local environment based on their own autonomous behaviors. In this paper, we present the formulations, algorithms, and experimental validations using both synthetic and empirical datasets. As shown in our experiments, our distributed strategy can effectively find most of the highest-degree nodes in large-scale decentralized networks, including those with community structures. This strategy is computationally efficient in that it can find the immunized nodes within a few cycles which is particularly useful for finding an approximate solution given a hard deadline. It is also demonstrated that the strategy is scalable in dealing with an open and/or growing network based on the self-organization of entities.

The remainder of this paper is organized as follows: Section 2 states some important research questions. Section 3 presents the basic ideas and the detailed formulations of our proposed strategy. Section 4 provides several experiments for systematically validating our strategy. Finally, Section 5 highlights our major contributions.

2 Problem Statements

Technically speaking, the problem of decentralized network immunization can be translated into the question of how to most effectively and efficiently prevent a virus from turning into an epidemic, given a minimum number of vaccinated nodes. In a decentralized environment, this problem is mathematically equivalent to a distributed combinatorial optimization problem (DCOP) [19], for which we are interested in finding the largest sum of vaccinated degrees if given a fixed number of vaccinated nodes.

***Definition** 1: **Graph** $G = \langle V, L\rangle$ is a network where $V=\{v_1, v_2, ..., v_N\}$ is the set of nodes (vertices), and $L = \{\langle v_i, v_j\rangle \mid 1 \le i, j \le N, i \ne j\}$ is the set of links (edges). $N = |V|$ represents the total number of nodes in the network, and $\langle v_i, v_j\rangle$ means there is a link between v_i and v_j.*

In the immunization network, we denote vaccinated nodes as $V_e \subseteq V$, where $|V_e|=N_e$ and $N_e \le N$. Existing work has shown that the *targeted immunization* strategy is the most efficient among others, due to the fact that it is a *centralized* strategy which searches all the nodes and finds those *important* nodes with the largest degrees and hence efficiently cuts a possible epidemic path [5], [11], [12]. Let the sum of the important node degrees be defined as $SID = \Sigma_{v_i \in V_e} v_i.degree$.

The immunization problem can be translated into that of finding those largest-degree nodes in the given network, i.e., maximizing *SID*. Generally speaking, it is an NP-hard problem to maximize *SID*, if N_e is constant in the network.

***Definition** 2: An **immunization strategy** refers to a scheme for achieving network immunization by means of selecting and immunizing a set of nodes V_e, which is denoted as $V_e = IS(V_0, G)$, where $V_0 \subseteq V$ is the initial set of* seed *nodes. In an autonomy-oriented computing (AOC) approach, these seed nodes correspond to the initial positions of the deployed autonomous entities. The set of selected nodes, V_e, corresponds to the final positions of the entities, i.e., a set of* important *nodes to be immunized.*

Given the same V_0, the criterion used to evaluate the efficiency of different strategies is the value of *SID*. As we know, this value will be globally optimized if the targeted immunization strategy is used. Therefore, in our current study, we can take the *SID* of the targeted immunization as our benchmark, and define our evaluation measure as follows:

***Definition** 3: The **coverage rate** of an immunization strategy, S, is defined as the proportion of the SID of a certain strategy to that of the targeted immunization, i.e.,*

$$coverage\_rate = \frac{SID_S}{SID_{Targeted}} \tag{1}$$

Based on the previous immunization strategy, it would be difficult to find the highest-degree nodes in a decentralized and/or dynamic network without a centralized control. In this paper, we aim to propose an autonomy-oriented strategy, called AOC-based strategy [1], [2], for tackling this problem. This strategy allocates N_e number of autonomous entities (*e*) in a large-scale decentralized network to search the highest-degree nodes. It takes into account the law of self-organization as the advocated computational means for immunizing the network. In order to achieve the optimal performance, i.e., the coverage rate as defined above is close to 1, each entity aims to move to the highest-degree node within its neighborhood and ensure $SID_{t+1} > SID_t$ at each step. Specific research questions to be answered are as follows:

1) **Coverage efficiency:** Can the AOC-based strategy rapidly find most of the highest-degree nodes in a decentralized network? In order words, is it possible for the AOC-based strategy to quickly and stably converge to a set of good solutions with the largest *SID*?
2) **Computational cost:** Is the cost of the AOC-based strategy lower than that of other existing strategies, e.g., the D-steps strategy? Two aspects of this question need to be addressed: One is concerned with the distributed requirement, i.e., how many steps (or cycles) in principle does the AOC-based strategy need in order to find the set of immunized nodes (V_e)? Another aspect is the sequential consideration, i.e., how would it compare with others in its total CPU runtime, if we use a sequential simulation (implementation) of the strategy?
3) **Scalability:** When the scale of a network is growing, can the performance of the AOC-based strategy remain computationally effective and efficient? In other words, does the AOC-based strategy have the ability to scale up its computation?

3 THE AOC-BASED STRATEGY

In this section, we will present the detailed formulations of the AOC-based strategy, introducing the notions of autonomous entities, environments, local behaviors, and evaluation functions, following the basic framework and guidelines given in [1], [2]. Also, we will discuss the conditions of self-organized computing in our strategy.

3.1 Entities and Their Local Environments

In the AOC-based strategy, a group of autonomous entities are deployed into the network (Graph *G* based on Definition 1). These entities can reside and move through the network. When the dynamically-moving entities converge to a static state, the nodes that are selected by the entities will be seen as immunized ones.

***Definition** 4: Let e be an **entity** in a network G. Entity e is represented by a tuple $\langle id, nodeId, utility, lifecycle, rules\rangle$, where id denotes the identifier of an entity, i.e., $e.id = i, i \in [1, N_e]$. N_e is the total number of entities. nodeId and utility represent the identifier and the degree of the node resided by e, respectively. lifecycle is the maximum time steps for an entity to reside on a node. The rule set stores local behaviors, which include rational-move, random-jump, and wait.*

In our strategy, the interaction between two entities will be based only on their local environments. The local environment of entity *e* includes both its direct and indirect neighboring nodes. The direct neighbors consist of the nodes that have edges connecting to the resided node. And, the indirect neighbors are the nodes that are the highest-degree nodes encountered by entities. Fig. 1a shows some cases of local environments that are constituted by direct neighbors only. On the other hand, Fig. 1b illustrates the local environment of e_2 that is composed of both direct and indirect neighbors.

***Definition** 5: The **local environment** of an entity, as denoted by E_l, contains a set of direct and indirect neighbors of e. If an entity e resides on node v_i, its local environment is defined as $E_l(e) = \{v_j \vee v_j.friendId \mid \langle v_i, v_j\rangle \in L\}$, where friendId stores the nodeId of a node that is (1) the second highest-degree one, not being resided by any other entities, and (2) known to the entity which has ever visited and/or encountered v_j during its search process. If none of the entities visits v_j and its direct neighbors, the friendId of v_j is null.*

Specifically, when an entity e moves to v_i, its $v_j.friendId$, $\langle v_i, v_j\rangle \subseteq L$, will be marked to the $nodeId$ of a node that has the maximum degree among (1) the direct neighbors and (2) the indirect neighbors (if $v_j.friendId$ is not null before the arrival of e) of v_i, as well as (3) the second highest-degree node in the search path of e.

For example, as shown in Fig. 1a, the local environment of e_1 is composed of its direct neighbors, i.e, $\{v_8, v_9, v_{11}\}$. This is due to the fact that the $friendId$ of these nodes are null before e_1 encounters them. In Fig. 1b, when e_1 moves from v_{10} to v_{11}, its local environment $E_l(e_1)$ becomes $\{v_6, v_{10}, v_{12}, v_{13}, v_{14}, v_{15}\}$. That is because no entity has updated the *friendId* of these nodes. However, after e_1 updates its local environment based on Eq.2, $v_6.friendId$ changes from null to 15; that is, $v_6.friendId$=$argmax\{E_l(e_1), v_{10}\}$=$v_{15}.nodeId = 15$, which is the highest-degree node among the direct and indirect neighbors of v_{11}, and the second highest-degree node in the search path of e_1 (i.e., v_{10}). Therefore, when e_2 moves from v_5 to v_4, its local environment is composed of both direct and indirect neighbors, i.e., $\{v_3, v_5, v_6, v_7\}\vee\{v_{15}\}$.

$$v_j.friendId(t+1) = argmax\{\ E_l(e_i(t+1)),\ e_i.nodeId(t)\ \},\quad n_j \in E_l(e_i(t+1)) \tag{2}$$

where *argmax*$\{\cdot\}$ returns a *nodeId* corresponding to the node that has the maximum degree and not resided by any entities in the sets of $E_l(e_i(t+1))$ and $e_i.nodeId(t)$.

***Definition** 6: The **profit** from an entity's local environment is defined as $\delta(e) = M(E_l(t))$, where $M(\cdot)$ returns the degree of the highest-degree node in the local environment of* e *at time step t.*

In Fig. 1a, $\delta(e_1)$=6 and $\delta(e_2)$=4, the highest-degree nodes are composed of direct neighbors. However, in Fig. 1b, v_{15} is an indirect neighbor of e_2, which is the highest-degree node in the local environment of e_2. According to this definition, we can note that the local

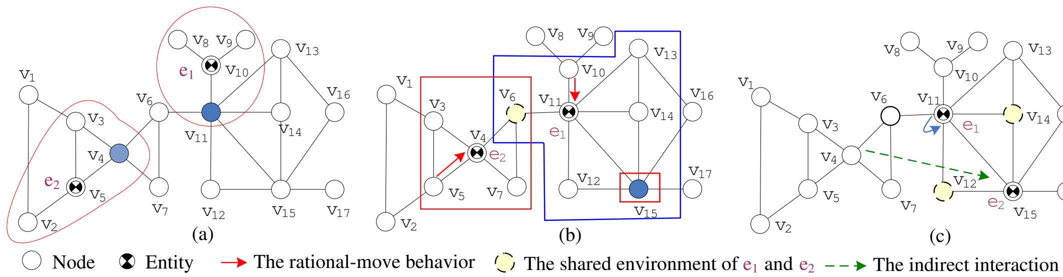


Fig. 1: (a) Illustrations of two autonomous entities and their local environments. (b) e_2 enlarges its local environment through its interaction with e_1. (c) The positive feedback mechanism helps entity e_2 find a new position.

environment E_l is not simply a static set of nodes. It can be modified by other entities. By this, it is meant that entities can undertake *indirect interactions*, and thus achieve the *positive feedback* from their local environments.

As explained in [1], the sufficient condition for self-organized computability is that entities should implicitly incorporate certain global influences. In our present strategy, the coupling relationships among entities and their local environments have implicitly reflect the global influences, that is, all entities attempt to reside on the highest-degree nodes in a network.

In what follows, we will further define the local behaviors of an entity, which further embody the necessary condition for self-organized computability, i.e., the short and long-range stochastic or exploratory actions, and the positive feedback-based accelerated aggregations [1].

3.2 Local Reactive Behaviors of Entities

Our central issue in designing a distributed strategy is how to enable entities to rapidly find those highest-degree nodes without centrally ranking all the nodes (since there is no centralized control involved). As we have mentioned, entities will indirectly interact among themselves based on the indirect neighbors of nodes that have been visited by others. Each of them attempts to move to a node whose degree will be higher than that of its currently resided node, based on the local information and the feedback from the environment. The local attraction will keep the entities with higher utilities residing in present places and those with lower utilities moving toward other positions. Meanwhile, a *long-range* move will enable an entity to escape out of local optima.

1) **Rational-Move**. An entity moves to a highest-degree position in its local environment by means of rational-move. If there exist more than one highest-degree positions, the entity will choose the first one from its friend list. Fig. 1 provides an illustrative example. The corresponding algorithm is given in Alg. 1 in [23].
2) **Random-Jump**. An entity moves randomly to a distant or long-range node in order to avoid getting stuck in local optima. This would become especially critical if the network concerned contains community structures, as it is difficult for entities to move out of them. At the same time, the random-jump is necessary for realizing the diversity and emergence of computable configurations in the AOC system [1].
3) **Wait**. Besides the two moving behaviors, an entity can also stay in the old position, whenever it cannot find any better positions than the current one. In Fig. 1c, e_1 resides in the old position, and e_2 moves to a new position.

3.3 Evaluation Functions

In order to effectively apply its behavioral rules as mentioned above, an entity needs to evaluate its current state, utility, as well as the current states of its local environment, by using several evaluation functions [1].

Each entity can only sense its local environment and apply the behavioral rules to govern its degree-search moves. The utility (degree) of an entity can be estimated iteratively based on the following equation:

$$e_i.utility(t+1) = \begin{cases} \delta_l(e_i(t)), \ if \ e_i.utility \leq \delta_l(e_i(t)) \\ n_j.degree, \ if \ e_i.lifecycle < 1 \\ \qquad \text{and } e_i.utility > \delta_l(e_i(t)) \\ e_i.utility(t), \ if \ e_i.lifecycle \geq 1 \end{cases} \tag{3}$$

Eq. 3 provides the basis for establishing a positive feedback between an entity and its environment. The equation contains three parts, namely, three different results of behaviors.

Further, when an entity moves to a new position, it will update its lifecycle as follows:

$$e.lifecycle(t+1) = e.lifecycle(t) + \triangle t \tag{4}$$

where $\triangle t$ is a reward value (which is the degree increment at each step) for keeping an entity resided on the node with a higher degree. Meanwhile, its lifecycle will decay over time based on Alg. 3 in [23].

At the end of each cycle, the collective outcomes of the system will be evaluated in order to accelerate the whole system evolving toward a new level based on Definition 6 and Eq. 5:

$$\delta(e_i(t+1)) = Max\{M(E_l(e_i(t+1))), e_i.utility(t)\} \tag{5}$$

Meanwhile, an entity can improve the profit of its local environment by updating the *friendId* of the neighbors based on Eq. 2 as given in Section 3.1. Thus, the behavior *probability* of *rational-move* can be indirectly affected by the increased profit from the local environment.

For example, after the rational-move behavior in Fig. 1b, entity e_1 improves its utility from 3 to 6. Then, the entity updates its respective local environments. In Fig. 1b, e_1 will evaluate the profit of the local environment ($\delta(e_1) = 5$) and update its local environment, e.g., $v_6.friendId = argmax\{\delta(e_1)\} = 15$. Specially, v_6 is a shared environment between e_1 and e_2. Therefore, the local environment of e_2 includes direct neighbors $\{v_3, v_5, v_6, v_7\}$ and an indirect neighbor $\{v_{15}\}$. Meanwhile, the local environment profit of e_2 becomes $\delta(e_2) = 5$. This corresponds to an indirect interaction between e_1 and e_2 through their shared environment.

The positive-feedback control mechanism, as mentioned earlier, plays an essential role in effectively triggering and speeding up the global emergence of finding important nodes. It accelerates entities in converging toward to their optimal positions through enlarging the scope of their local environments. For example, if there were no indirect interaction, e_2 would miss the second highest-degree node v_{15} in Fig. 1b. With the indirect interaction, the local environment profit of e_2 is improved from 4 to 5. Thus, e_2 will move from v_4 to v_{15} in Fig. 1c in the next time. More evaluations are shown in Fig.4.

4 Experimentation and Validation

The main algorithm of the AOC-based strategy and subfunctions are provided in [23]. In this section, some experiments are used to evaluate the search performances and some special features of the AOC-based strategy in order to answer the research questions introduced in section 2.

For our experimental validation, we have generated some synthetic networks of different scales, even with community-based structures, by using the GLP algorithm [20] in which the exponent can be tuned. Meanwhile, we also use five publicly available benchmark networks, including two email networks, two autonomous systems networks (i.e., AS networks), and one typical community-based coauthorship network. The structures of different networks and some symbols mentioned in our experiments are listed in Table I and II in [23], respectively.

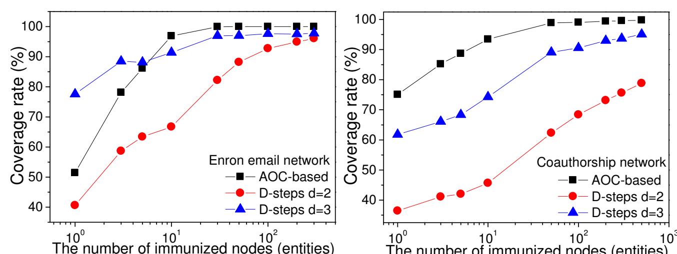
4.1 Coverage Efficiency

In this section, we will evaluate the coverage efficiency of the AOC-based strategy. All our experimental results will be given in average values obtained over 100 simulation runs. Several parameters will be used in evaluating the efficiency of different strategies, as follows:

1) **Entity steps:** the average moving steps of an entity;
2) **Coverage rate (%):** the ratio of the total degree obtained by an immunization strategy to be evaluated to the total degree by the targeted strategy;
3) **Environment steps ($\overline{p}$):** the updating times (or cycles) for the AOC-based strategy to converge to a stable state;
4) **Time (ms):** the average CPU runtime based on a sequential simulation (implementation) of the strategy (the unit is in milliseconds).

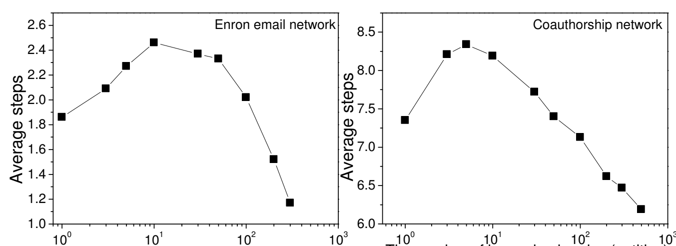
4.1.1 Simulation Results

Fig. 2 and Fig. 3 present the results of coverage rates and average entity steps for different numbers of immunized nodes, N_e, as obtained from comparative experiments using the Enron email network and coauthorship network (i.e., NET3 and NET7 in [23]). It can be noted that the AOC-based strategy demonstrates an overall higher coverage rate than the D-steps strategy. At the same time, the average entity moving steps in the AOC-based strategy decreases with respect to the increase in the number of immunized nodes; this nonlinear decrease is due to the effects of local interactions among entities.



(a) Enron email network (b) Coauthorship network

Fig. 2: The coverage rate on different networks (semilog charts). The AOC-based strategy leads to better results than the D-steps strategy.



(a) Enron email network (b) Coauthorship network

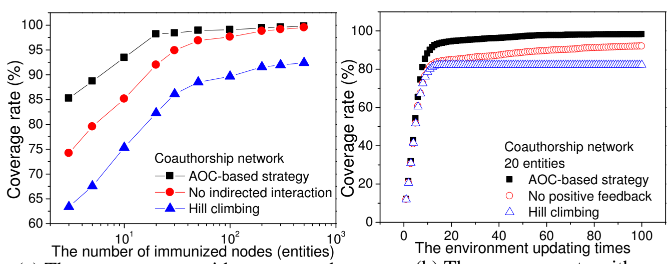
Fig. 3: The average moving steps on different networks (semilog charts). All entities need only a few steps to find the targeted positions.

The Enron email network contains many isolated communities each of which contains only two nodes. If there are only a few entities in a network, the entities need more *move* steps to find their destination positions. That is because there are no indirect interactions among entities nor emergent computing. Only when an adequate number of entities is reached (at least five entities are necessary as shown in Figs. 2 and 3), the efficiency of the AOC-based strategy can nonlinearly grow based on the indirect interactions among entities, i.e., their indirect interaction-based positive feedback mechanism resulting in the nonlinear performance improvement. Thus, there can be seen an *inflection point* in Fig. 3. The same situation has also been observed in other networks (shown in Fig. 9 in [23]).

As shown in these figures, the coverage rates of the AOC-based strategy are much higher, and the average moving steps are fewer, than the D-steps strategy in most of the cases. This indicates that our strategy can effectively find most of the highly-connected nodes in various networks with fewer moving steps.

4.1.2 Performance Analysis

As we know, both the acquaintance and D-steps strategies can find n highest-degree neighbors around seed nodes. If the seed nodes are randomly disposed on the outliers of the network or centralized inside one community, this would affect the coverage efficiency of the D-steps strategy. But, the autonomous entity, in the AOC-based strategy, chooses adaptive behaviors according to the feedback from its environment and its own state at each time step. The *rational-move* enables an entity to find the highest-degree node from its local neighbors, whereas the *random-jump* enlarges its search scope that ensures the diversity of the potential solutions.



(a) The coverage rate with respect to the number of entities

(b) The coverage rate with respect to times

Fig. 4: The influence of positive feedback in the coauthorship network. The triangles correspond to the case of applying a hill climbing method, i.e., there are no jump behaviors in the strategy. The circles correspond to the case where the *move* behaviors of entities are based only on their directed neighbors with no indirect interactions. This is a typical local search method that combines hill climbing and random walks together.

The influences of positive feedback on the coauthorship network are shown in Fig. 4. Based on the indirect interactions among autonomous entities, the global utility has shown a nonlinear growth. The AOC-based strategy embodies a positive feedback mechanism to speed up the search for desired positions, i.e., all entities can quickly converge to a stable position that has a high degree. That is because the higher the global utility is, the larger the search scope for entities will become. The entities will collectively find more highly connected nodes in the next step and the strategy quickly converges to a stable state. Conversely, the entities without the indirect feedback from other entities have to spend more time and steps to find the destination positions. And, the entities without the random-jump behaviors will stop exploring if they fall into local optima. We have selected twenty entities as examples to observe the coverage rate with respect to environment updating times. The results are shown in Fig. 4b, which again validate the above analysis. Some experiments in [21] have shown that various collective effects can be reached and observed over either different temporal and/or spatial scales.

Fig. 2 further provides the results of comparisons between the AOC-based and D-steps strategies. When d and the number of immunized nodes are large enough, the D-steps strategy can search almost the entire network. In other words, the coverage rate of the D-steps strategy relies on the size of steps and the number of immunized nodes. On the other hand, the coverage efficiency of the AOC-based strategy relies on the emergent computing and self-organization among entities. When there are fewer entities on the network, the efficiency may be lower than the D-steps strategy, especially on the community-based networks, e.g., the Enron and coauthorship networks. When the number of entities reaches a certain level, the efficiency of the AOC-based strategy will nonlinearly grow based on the positive feedback and self-organization among the entities.

In order to examine the changes of coverage rates with respect to environment updating times (steps), varying numbers of entities are allocated into different networks. The result is shown in Fig.5. More results are provided in Fig. 11 in [23].

From the above experiments, we can observe that:

- the AOC-based strategy can achieve the maximum coverage set within the finite steps;
- the coverage rate of the AOC-based strategy is higher than that of the D-steps strategy because the long jump ensures the diversity of potential solutions and enlarges the searching range of entities;
- the AOC-based strategy adapts well to networks with community structures, whereas the D-steps strategy will fall into local optima.

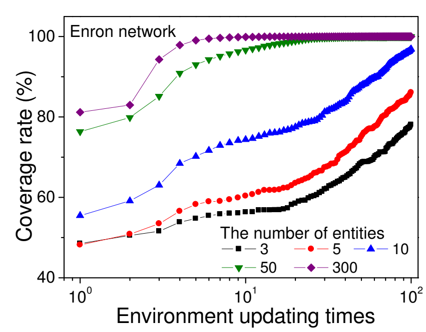


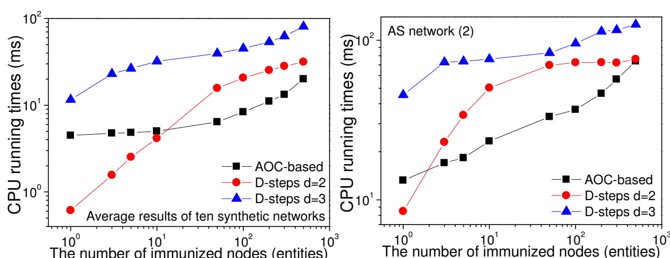
Fig. 5: The coverage rates with respect to the updating steps on the Enron email network (semilog charts). The coverage rate with a larger number of entities reaches higher than those with fewer entities.

TABLE 1: The complexities of different strategies.

Strategy	Complexity
Targeted immunization	$O(N^2)$
Betweenness immunization	$O(N * E)$
Acquaintance (random neighbor)	$O(n * k)$
Acquaintance (max neighbor)	$O(n * (\langle K \rangle^2 + k))$
D-steps immunization	$O(n * (\langle K \rangle^{d-1}) * (\langle K \rangle^2 + 1))$
AOC-based immunization	$O(n * \overline{p} * (\langle K \rangle^2 + 1))$

4.2 Computational Cost

Table 1 lists the worse-case computational complexities of the existing immunization strategies. The descriptions of parameters and more analyses about computational complexities are provided in Section III-C in [23].



(a) Average results of ten networks (b) AS network

Fig. 6: The sequential cost with respect to the number of immunized nodes on the synthetic and benchmark networks (logarithmic charts). (a) is averaged over NET1 and nine networks, as listed in Table III in [23].

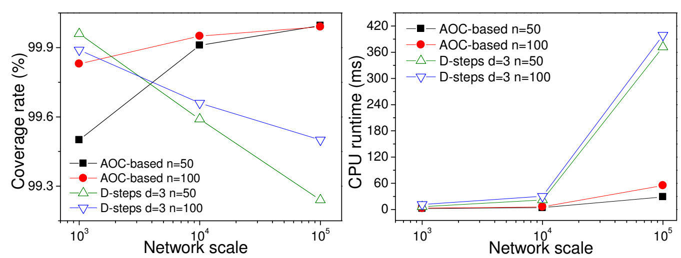
Some large-scale networks, ten synthetic networks (i.e., NET1 and nine networks with 10^4 nodes as listed in Table III in [23]) and AS network (i.e., NET6 in [23]), are selected for comparing both the sequential and distributed time of the D-steps and AOC-based strategies. Here, the sequential time corresponds to the runtime as required by the sequential implementations (simulations) of the distributed algorithms. The distributed time, however, measures the number of synchronized cycles or time steps as needed for the distributed computations. The results are shown in Fig. 6. From this result, we can see that the runtime of the AOC-based strategy is shorter than that of the D-steps strategy in most of the situations.

Based on the above analysis of computational complexity, we know that $\langle K \rangle^{d-1}$ and $\overline{p}$ will both determine the cost of the AOC-based strategy. For the AS network, $\langle K \rangle$ is 4.22, and system updating time $\overline{p}$ is equal to 50. In this case, the runtime of the AOC-based strategy is more than that of the D-steps strategy. However, the above analysis is based on the worst case. In the actual operations, the computing time of each entity is less than $\overline{p}$. Since most of the entities reside in their old positions, the sequential time of the AOC-based strategy is in fact less than the D-steps strategy, especially in the case of larger-scale networks. That is why the experimentally-measured cost of the AOC-based strategy is lower than the D-steps strategy in Fig. 6, even if $\overline{p} > \langle K \rangle^{d-1}$. More explanations are provided in Fig. 10 in [23].

4.3 Scalability

In this section, we examine the scalability of the AOC-based strategy. Specifically, we compare the coverage rates and cost of both the AOC-based and D-steps strategies with respect to different network scales. The comparison results are shown in Fig. 7. From the figure, we note that the coverage rate of the AOC-based strategy is much closer to that of the targeted immunization with the growth of network scales. However, the coverage rate of the D-steps strategy decreases with the increase of network scales. This is because the core of the AOC-based strategy lies in entities' local autonomy and their self-organization. The larger the network scale, the more effective and efficient the behaviors of entities will become. The entities self-organize their own behaviors according to their states and positive feedback from the environments. On the other hand, the behaviors in the D-steps strategy are unchangeable; the highest-degree node will be chosen only around the seed nodes. Therefore, the efficiency of the D-steps strategy will be restricted to the number and initial positions of seed nodes, as well as the length of steps d.

Fig. 8 presents a comparison of their distributed steps in different network scales. The result also confirms the scalability of the AOC-based strategy in that each entity only linearly increases one step when the network scale increases by one order of magnitude.



(a) Coverage rate (b) Comparing cost

Fig. 7: Scalability evaluations of the AOC-based and D-steps strategies (semilog charts). The coverage rate of the D-steps decreases and the cost of the D-steps drastically increases with respect to the increments of network scales. However, the network scale will not affect the coverage rate of the AOC-based strategy, which, as expected, gets an even higher coverage rate with a lower cost in the larger network.

5 Conclusions

In this paper, we have presented a distributed and feasible immunization strategy that is based on the ideas of autonomy-oriented computing (AOC). In order to investigate the efficiency of our strategy, we have compared our strategy with some existing strategies using both

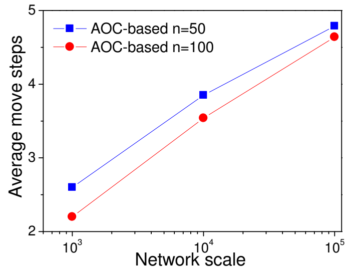


Fig. 8: A comparison of average moving steps per each entity in different scales (semilog charts).

synthetic and real networks, including those with community structures. Our experimental results have shown that the AOC-based strategy is effective for large-scale decentralized and community-based networks, and the robustness of the AOC-based strategy in dynamically-evolving networks has been discussed in [22]. Specially, the AOC-based strategy has the ability to scale up its computation based on the design of self-organization. The efficiency of our strategy will improve with the increase of network scale. The results have shown that the cost of the AOC-based strategy is lower than others.

In summary, our work has offered a means for network immunization. It enables us find the highest-degree nodes quickly on a large-scale decentralized and/or community-based network with very low cost in order to cut epidemic paths.

Acknowledgments

The authors are grateful to the anonymous reviewers for their comments and suggestions. This work was supported by National Natural Science Foundation of China (60673015), Beijing Natural Science Foundation (4102007), and Hong Kong Research Grants Council grant (210508/32-08-105).

References

[1] J. Liu, "Autonomy-Oriented Computing (AOC): The nature and implications of a paradigm for self-organized computing," *Proceedings of the 4th International Conference on Natural Computation (ICNC'08)*, pp. 3–11, Oct. 2008.

[2] J. Liu, X. Jin, and K.C. Tsui, *Autonomy Oriented Computing (AOC): From Problem Solving to Complex Systems Modeling*, Kluwer, 2005.

[3] S.H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, May 2001.

[4] A.-L. Barabasi, and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[5] R. Pastor-Satorras and A. Vespignani, "Immunization of complex networks," *Physical Review E*, vol. 65, no. 3, p. 036104, Mar. 2002.

[6] Z. Dezso and A.-L. Barabasi, "Halting viruses in scale-free networks," *Physical Review E*, vol. 65, no. 5, p. 055103, May 2002.

[7] P. Holme and B.J. Kim, "Vertex overload breakdown in evolving networks," *Physical Review E*, vol. 65, no. 6, p. 066109, June 2002.

[8] Y. Chen, G. Paul, S. Havlin, F. Liljeros, and H.E. Stanley, "Finding a better immunization strategy," *Physical Review Letters*, vol. 101, no. 5, p. 058701, Aug. 2008.

[9] R. Cohen, S. Havlin, and D. Ben-Averaham, "Efficient immunization strategies for computer networks and populations," *Physical Review Letters*, vol. 91, no. 24, p. 247901, Dec. 2003.

[10] P. Holme, "Efficient local strategies for vaccination and network attack," *Europhysics Letters*, vol. 68, no. 6, pp. 908–914, Nov. 2004.

[11] L.K. Gallos, F. Liljeros, P. Argyrakis, A. Bunde, and S. Havlin, "Improving immunization strategies," *Physical Review E*, vol. 75, no. 4, p. 045104, Apr. 2007.

[12] J. Gomez-Gardenes, P. Echenique, and Y. Moreno, "Immunization of real complex communication networks," *European Physical Journal B*, vol. 49, no. 2, pp. 259–264, Jan. 2002.

[13] P. Echenique, J. Gomez-Gardenes, Y. Moreno, and A. Vazquez, "Distance-d covering problem in scale-free networks with degree correlation," *Physical Review E*, vol. 71, no. 3, p. 035102, Mar. 2005.

[14] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 251–262, Oct. 1999.

[15] M.E.J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, Mar. 2003.

[16] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, "Resilience of the internet to random breakdowns," *Physical Review Letters*, vol. 85, no. 21, pp. 4626–4628, Nov. 2000.

[17] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Physical Review Letters*, vol. 86, no. 14, pp. 3200–3203, Apr. 2001.

[18] R. Albert, H. Jeong, and A.-L. Barabas, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, Jul. 2000.

[19] T.W. Sandholm and V.R. Lesser, "Coalitions among computationally bounded agents," *Artificial Intelligence*, vol. 94, no. 1-2, pp. 99–137, Jul. 1997.

[20] T. Bu and D. Towsley, "On distinguishing between internet power law topology generators," *Proceedings of the Twenty First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, pp. 638–647, Jun. 2002.

[21] J. Liu, C. Gao, and N. Zhong, "An autonomy-oriented paradigm for self-organized computing," *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'09)*, pp. 100-103, Sept. 2009.

[22] J. Liu, C. Gao, and N. Zhong, "Autonomy-oriented search in dynamic community neworks: A case study in decentralized network immunization," *Fundamenta Informaticae*, vol. 99, no. 2, pp. 207-226, 2010.

[23] C. Gao, J. Liu, and N. Zhong, "Network immunization with distributed autonomy-oriented entities (Supplementary file)," *Digital Library of IEEE Transactions on Parallel and Distributed Systems*.

Chao Gao is currently a PhD student in the International WIC Institute, College of Computer Science and Technology, Beijing University of Technology. He has been an exchange student in the Department of Computer Science, Hong Kong Baptist University, for four months. His main research interests include Web Intelligence (WI), Autonomy-Oriented Computing (AOC), complex networks analysis, and network security.

Jiming Liu is the Chair Professor and Head of Computer Science Department at Hong Kong Baptist University. His current research interests include: Autonomy-Oriented Computing (AOC), Web Intelligence (WI), and self-organizing systems and complex networks. Prof. Liu has contributed to the scientific literature in those areas, including over 250 journal and conference papers, and 5 authored research monographs. He has served as the Editor-in-Chief of *Web Intelligence and Agent Systems* and an Associate Editor of *IEEE TKDE* and *IEEE TSMC-B*.

Ning Zhong is currently Head of the Knowledge Information Systems Laboratory and is a Professor in the Department of Systems and Information Engineering at Maebashi Institute of Technology, Japan. He has conducted research in the areas of knowledge discovery and data mining, rough sets and granular-soft computing, Web Intelligence, intelligent agents, brain informatics, and knowledge information systems, with more than 250 papers and 10 books.