Network Immunization with Distributed Autonomy-Oriented Entities (Supplementary File)

Chao Gao, Jiming Liu, Ning Zhong

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.

I. Network Immunization In Complex Networks

The notion of complex networks has been used in different domains for modeling and characterizing the structures and dynamics of certain physical and/or conceptual networks. For instance, the node of a complex network may denote a computer in a physical computer network or an individual in a conceptual/logical network of emails or instant messaging. The link between two nodes in a complex network may denote a certain physical connectivity or relationship/interaction between them, such as a physical connection between two computers, frequent email communication (e.g., sending/receiving) between two individuals, or a situation of two people appearing on each other's contact lists.

With the growth of computer network-based applications, viruses have extended their propagation from physical networks to logical networks, such as Internet worms in computer networks, rumors in social networks or mobile phone networks, SARS and H1N1 in biology networks. Currently, network immunization is one of the most effective techniques to restrain virus propagation in complex networks. In what follows, we provide a survey of some typical distributed immunization strategies.

A. *Acquaintance Immunization*

The acquaintance immunization [2] is an effective local strategy that makes a tradeoff between the efficiency and the feasibility of distributed immunization without knowing the global information about a network. First, it randomly selects a *seed* node and then randomly immunizes one of its acquaintances. The probability for selecting an acquaintance node is proportional to its corresponding degree of connectivity; an acquaintance with a higher degree will have more links in the scale-free network.

Figs. 1(a) and 1(b) show this process. Suppose that v_1 is a seed node, v_3 is selected as an immunized node from the set of $\{v_3,\ v_6,\ v_8\}$ based on the acquaintance immunization. This method is more efficient than the random immunization, but less efficient than the targeted immunization. Later, some variants of the acquaintance immunization strategies[3], [4] have been proposed to improve the efficiency. However, since they do not make distinctions among different neighbors and only randomly select seed nodes and their direct neighbors, their improvement in efficiency is still quite limited [5].

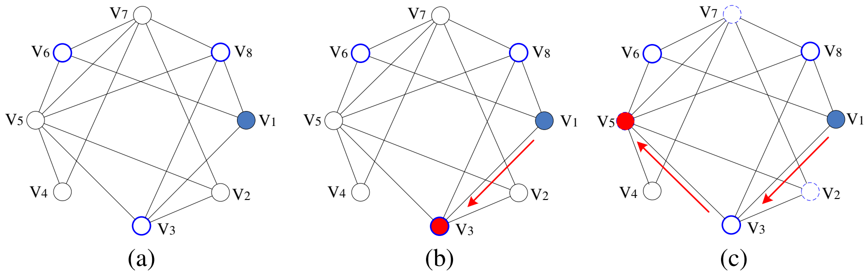


Fig. 1. Illustrations of different strategies. Suppose that v_1 is a seed node. v_3 will be immunized based on the *acquaintance immunization* strategy and v_5 will be indirectly selected as an immunized node based on the *D-steps immunization* strategy, where d=2 .

B. D-Steps Immunization

Another effective distributed strategy is the D-steps immunization (or covering immunization) [5], [6]. This strategy formulates distributed immunization as a graph covering problem.

For a node v_i, this strategy immunizes the highest-degree neighbor within the d number of steps. Suppose that v_1 is the seed node and the length of steps d=2, as shown in Fig. 1(a). The direct and indirect neighbors are denoted as solid and dashed nodes in Fig. 1(c), respectively. v_5 is selected as an immunized node based on the D-steps immunization. Note that v_5 has the highest-degree in this network. This strategy only utilizes the local topological knowledge within a certain range. Thus, it is more effective than the acquaintance immunization.

C. Remarks

Many networks in the real world contain clique or community structures [7]. The nodes within the same community will have more connections than those from different communities. At the same time, in some networks (e.g., P2P network), peers form self-organizing overlays that go beyond services offered by a client-server system. In this kind of network, a client may also be a server where only a loose control is possible. The current distributed immunization strategies, as mentioned above, is unable to find those important nodes and protect community networks well, because the strategy is static and the topology of the network is changing all the time.

In order to search highly-connected nodes in large-scale decentralized networks, we present a framework for formalizing the distributed immunization strategy by extending distributed constrained heuristic search (DCHS) [8] into an AOC-based approach [9], [14]. Both DCOP and

DCHS are typical distributed artificial intelligence problems whose solutions rely on cooperative entities searching while satisfying some constraints. In the immunization strategy, the optimality is that the total degree of these entities should be the largest in decentralized networks.

It is well known that local search is an effective approach to solving computationally hard search problems [15], e.g., DCOP and DCHS [16]. As a widely-used dynamic process [17], random walk has been applied in data mining [18], as well as in complex network studies [19], [20]. To some extent, the distributed AOC-based approach [9], [14] that utilizes a self-organization method for locating highly-connected nodes in a decentralized and/or dynamic network incorporates the ideas of local search and random walk.

The remainder of this supplementary file is organized as follows: Section II presents the basic ideas and the algorithms of our proposed strategy. Section III provides several experiments for systematically validating the efficiency and special performances of the AOC-based strategy. Section IV concludes our contributions.

II. Formulations of the AOC-based Strategy

Different from traditional agent-based systems. AOC-based systems emphasize autonomy-oriented self-organization based on certain nature and/or real-world inspired rules [9], and can find broader applications in: (1) characterizing working mechanisms that lead to emergent behavior in natural and artificial complex systems (e.g., phenomena in Web Science, and the dynamics of social networks and neural systems) [10], [11], and (2) developing solutions to large-scale, distributed computational problems (e.g., distributed scalable scientific or social computing, and collective intelligence) [12], [13].

An AOC-based system is an open and non-equilibrium system that generally contains four basic elements:

1) A **global environment** (E_g) serves not only as the domain for autonomous entities, but also as an indirect communication medium among entities.
2) **Autonomous entities** (e) react to each other as well as their **local environments** (E_l), as shown in Section II-A, based on local behaviors and behavioral rules.
3) **Local behaviors and behavioral rules** (B) that govern the entities react to exterior information, as to be introduced in Section II-B.

4) An entity evaluates its current state and utility, as well as the state space of its local environment by means of **local utility function** (u) **and system evaluation functions** (F), to be described in Section II-C, in order to effectively select autonomous behaviors b from B.

Finally, certain behaviors of the entities and/or their effects are non-linearly aggregated and amplified as opposed to other behaviors and/or effects, that is, a desired solution and/or structure for complex tasks is emergent on the whole through the local interactions among entities [9], [29]. In this section, we present the detailed formulations of the AOC-based strategy following the basic framework and guidelines given in [14].

A. Autonomous Entities and Their Local Environments

In our strategy, we use a graph G to denote the global environment. The graph G serves not only as the domain for autonomous entities, but also as an indirect communication medium among entities, i.e., the *positive feedback* from their local environments. Autonomous entities (denoted by e) only react to their local environments composed of some direct and indirect neighbors. The goal of an entity is to find the node that has the highest degree within its local environment. Fig. 2 provides some examples of autonomous entities and their local environments.

The coupling relationships among entities and their local environments include selection range and selection condition. The selection condition is that the highest-degree node will be chosen as the immunized node within the selection range of e_i. The local behaviors (*rational-move*, *random-jump* and *wait*) and rules that govern how entities would act or react to the information received constitute the core of autonomous entities. And, the selection range of an entity covers both the direct and indirect neighbors. For entity e_i, its selection range will be $\{v_j \vee v_j.friendId \ \mid \langle v_i, v_j \rangle \in L\}$. Fig. 2 shows some cases of local environments that are constituted by direct neighbors only. On the other hand, Fig. 4 illustrates the local environment of e_2 that is composed of both direct and indirect neighbors.

B. Local Reactive Behaviors

Each entity has three behaviors, i.e., **rational-move**, **random-jump** and **wait**. The rational-move is the best strategy to induce the entity to move to a nearby node with the maximum degree. Fig. 4 provides an illustrative example, where we can see how an entity moves from its

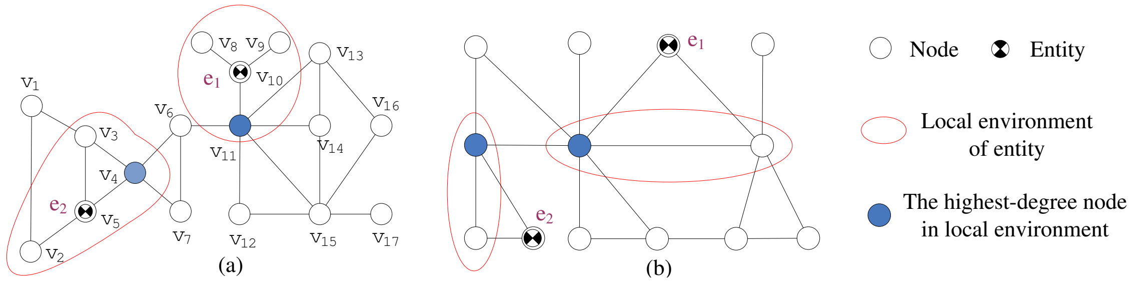


Fig. 2. Illustrations of two autonomous entities and their local environments.

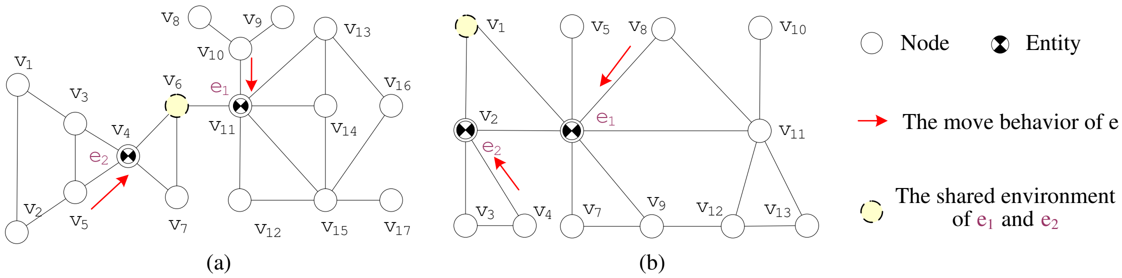


Fig. 3. Autonomous entities move to the highest-degree nodes based on the rational-move behavior.

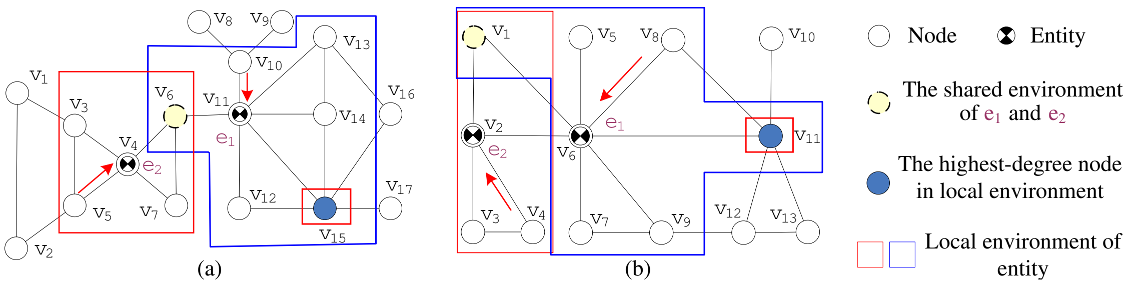


Fig. 4. The evaluation of a local environment. Entity e_2 enlarges its local environment through its interaction with e_1. (a) is the indirect interaction based on a shared environment v_6. (b) is the direct interaction between e_1 and e_2. The $\delta(e_2)$ is improved from 4 to 5 based on the positive feedback.

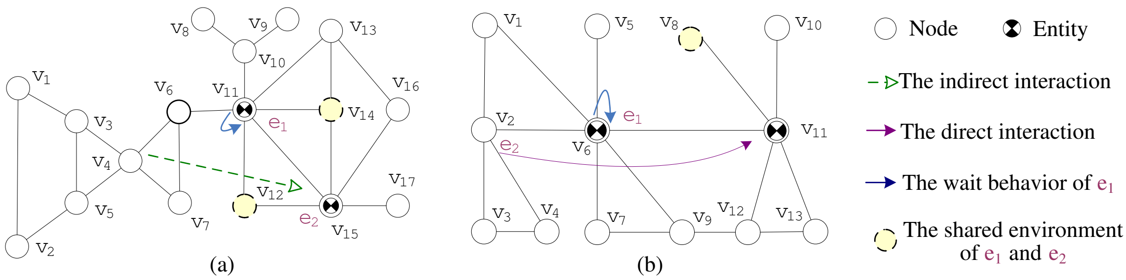


Fig. 5. The positive feedback mechanism helps entity e_2 find a new position in the network.

old position as shown in Fig. 2 to a new one following the arrow in Fig. 3. The corresponding algorithm is given in Alg. 1 following Rule 1.

Rule 1: If $e.utility < \delta(e)$, then $e.MoveTo()$.

Algorithm 1 Rational-Move: $MoveTo()$

Input: Entity e_i and node v_j

Output: Entity moving from $v_{e_i.nodeId}$ to v_j

//Record environment information

1: E_d=E_d+($v_j.degree - e_i.utility$);

2: $v_{e_i.nodeId}.resided$=0; //Update original node

//Implicitly record neighbor information

3: $v_{e_i.ID}.friendId = j$;

4: $v_j.resided$ = 1; //Update new node

//Update entity information

5: $e_i.nodeId = j$; $e_i.utility = v_j.degree$;

//Update lifecycle based on Eq.(4)

6: $e_i.lifecycle = e_i.lifecycle + \triangle t$

An entity moves randomly to a distant or long-range node. If there is only the rational move in our strategy, all entities may eventually stay in their local highest-degree positions. We call them local optima. In the case of local optima, no entity will be able move to a new position, and there will be no further feedback. Thus, the entities may lose their chances to find better destinations if there are no other behaviors to help them avoid getting stuck in local optima. In order to avoid the local optima, we employ some random long-range moves similar to the random walks in local search. At the same time, the random-jump is necessary for realizing the diversity and emergence of computable configurations in the AOC system [14]. The algorithm of random-jump is shown in Alg. 2 following Rule 2.

Rule 2: If $e.utility > \delta(e)$ and $e.lifecycle < 1$, then $e.JumpToNode()$.

Besides the two moving behaviors, an entity can also stay in the old position (i.e., a Wait behavior), whenever it cannot find any better positions than the current one. In Fig. 5, e_1 resides in the old position, and e_2 moves to a new position.

Algorithm 2 Random-Jump: $JumpToNode()$

Input: Entity e_i

Output: Entity that randomly jumps to a new position or resides in an old position

1: Randomly select one node v_j with no entity residing on it;

2: **If** $e_i.utility < v_j.degree$ **then**

3: $e_i.MoveTo(v_j)$; //Find a new position

4: **Else if** $e_i.utility < v_{v_j.friendId}.degree$ **then**

5: $e_i.MoveTo(v_{v_j.friendId})$; //Find an implicit position

6: **Else** $e_i.wait(\ )$;

7: **End**

C. Evaluation Functions

Some functions evaluate the local environments and update the profiles of entities. Local utility function (u) is used to improve entities' profiles and states. The algorithm u is shown in Alg. 3. At the same time, environment evaluation function (F) is used to update the local environment, especially the $friendId$ in the local environment of e and the local utility of e at time t, i.e., $\delta(e_i(t))$ based on Alg. 4.

For example, after the rational-move behavior in Fig. 4, entity e_1 improves its utility from 3 to 6 and from 2 to 7, respectively. Then, the entity updates its respective local environments. In Fig. 4(a), 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. On the other hand, different from Fig. 4(a), e_1 and e_2 have a direct interaction in Fig. 4(b). And, e_2 enlarges its local environment from $\{v_1, v_3, v_4\}$ to $\{v_1, v_3, v_4, v_{11}\}$ through the direct interaction with e_1. After that, entity e_2 will move from v_4 to v_{15} in Fig. 5(a), and from v_2 to v_{11} in Fig. 5(b) in the next time.

Algorithm 3 UtilityUpdate: u

Description: Updating the utility of each entity

1: **For** each entity e_i

2: $e_i.utility = \delta(e_i)$; //Based on Eq.(3)

3: $e_i.lifecycle$=$e_i.lifecycle$+ $\triangle t$ -1; //Based on Eq.(4)

4: **End**

Algorithm 4 LocalEvaluation: F

Description: Assessing environment information

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

2: **For** each v_j in $E_l(e_i(t+1))$

3: $v_j.friendId = argmax\{E_l(e_i(t+1)), e_i.nodeId(t)\}$

4: **End**

D. The Main Algorithm

Alg. 5 presents the main algorithm of the AOC-based strategy, which contains two phases. In Phase I, some nodes are randomly selected as initial *seed* nodes for autonomous entities to start their search. Next, in Phase II, the self-organized computing of entities occurs. This phase further involves two stages, namely, inactive and action stages. The task of the inactive stage is to assess and update the states of an entity and its environment utilizing function *LocalEvaluation()* (Alg. 4) and *UtilityUpdate()* (Alg. 3).

It is worth mentioning that the self-organized interactions will mainly occur in the action stage. In this stage, entities will choose different behaviors based on their own utility values and local environments. Each entity moves to the highest-degree neighbor based on function *MoveTo()* (Alg. 1), and escapes out of local optima by function *JumpToNode()* (Alg. 2).

From the above descriptions, we note that the entities that have found highly-connected nodes will not move further. Considering the worst case, we may use the network degree $\langle K \rangle$ to replace the average number of neighbors, and assume that the AOC-based strategy needs to update $\overline{p}$ times in order to converge to a stable state. In such a case, the computational complexity of our strategy would be $O(n * \overline{p} * (\langle K \rangle^2 + 1))$, where *n* is the number of entities. More detailed analysis will be given in Section III-C.

Algorithm 5 V_e=$IS_{AOC-based}(V_0, G)$

Input: A network G and a set of autonomous entities

Output: A final set of immunized nodes V_e composed of entities e

//Phase I: Initialize *e* with N_e random *seed* nodes

For each entity e_i

    Select one non-resided node from V for e_i

End

//Phase II: Self-organized computing

While ($\triangle SID > \varepsilon$)

////Inactive stage for environment evaluation and utility updating

    LocalEvaluation ( ); //Alg.4

    UtilityUpdate (e_i); //Alg.3

////Action stage based on Eq.(3)

  For each entity e_i

    If $e_i.utility < \delta(e_i)$ **then**

      $e_i.MoveTo(v_j)$; //Alg.1

    Else if $e_i.lifecycle < 1$ **then**

      $e_i.JumpToNode(\ )$; //Alg.2

        Else $e_i.wait(\ )$;

    End

  End

End

III. Experimentation and Validation

In this section, we first present some synthetic and benchmark networks used in our experiments in Section III-A. Then, we use some experiments to evaluate the performances and some special features of the AOC-based strategy:

1) The coverage efficiency and emergent computing by positive feedback and self-organization in Section III-B.
2) The computational cost, in Section III-C, which includes the analyses of computational

complexities of various strategies and the runtime performances of sequential simulation of different strategies, as well as the steps (or cycles) required in the distributed computation.

3) The robustness and immunization efficiency in Section III-D.
4) The ability of scalable computing in the AOC-based strategy in Section III-E.

Some symbols mentioned in our experiments are listed in Table I.

TABLE I

PARAMETERS AS USED IN OUR EXPERIMENTS.

Symbol	Description
N	Total number of nodes in a network
E	Total number of edges in a network
$\langle K \rangle$	Average degree of nodes in a network
α	Power-law exponent of a network, which is estimated with the maximum likelihood based on [21]
n	Number of the nodes that are selected as immunized nodes, i.e., $n = \|V_e\|$ (First, n nodes are selected as seed nodes. Then, different entities select destination nodes around the initial seed nodes based on their own actions.)
k	k neighbors of one seed node that are selected based on the acquaintance immunization strategy
d	d number of steps as used in the D-steps strategy to select destination nodes

TABLE II

NETWORKS WITH DIFFERENT STRUCTURES AS USED IN OUR EXPERIMENTS.

	Type	Evaluations	Structure
NET1	Synthetic network	Search efficiency, scalability, cost	N=10^4, E=42058, $\langle K \rangle$=8.41, α=2.49
NET2	Synthetic network	Search efficiency, scalability, cost	N=10^5, E=419190, $\langle K \rangle$=8.38, α=2.50
NET3	Enron email network	Search efficiency	N=1238, E=2106, $\langle K \rangle$=3.40, α=2.29
NET4	University email network	Search efficiency	N=1133, E=5451, $\langle K \rangle$=9.62, α=2.83
NET5	AS network (1)	Search efficiency, cost	N=22963, E=48436, $\langle K \rangle$=4.22, α=2.09
NET6	AS network (2)	Search efficiency, cost	N=12741, E=26888, $\langle K \rangle$=4.22, α=2.08
NET7	Coauthorship network	Search efficiency, cost positive feedback	N=56276, E=631632, $\langle K \rangle$=11.22, α=3.50 4 large communities and 600 small communities
NET8	Synthetic network	Scalability	N=10^3, E=4288, $\langle K \rangle$=8.46, α=2.20
NET9	Community-based network	Positive feedback	N=4000, E=16733, $\langle K \rangle$=8.34, α=2.39
A1-A9	Synthetic network	Average values of search efficiency and cost	N=10^4, the detailed structures are listed in Table III

A. The Structures of Networks to be Tested

Table II presents an overview of the synthetic and benchmark networks that we have used in our studies to evaluate the performance of the AOC-based strategy.

The structures of many real-world networks have long-tail statistical distributions. The degrees of these networks follow a power-law distribution [1] where the power-law exponent is often between 2 and 3 [22]. Thus, for our experimental validation, we have generated some synthetic networks of different scales by using the GLP algorithm [23] in which the exponent can be tuned (i.e., NET1, NET2, NET8, NET9, and some networks as listed in Table III). The degree distributions of the synthetic networks are shown in Figs. 6(a), 6(b), 6(i) and 6(h). The structure of a synthetic community-based network is given in Fig. 7(b).

TABLE III

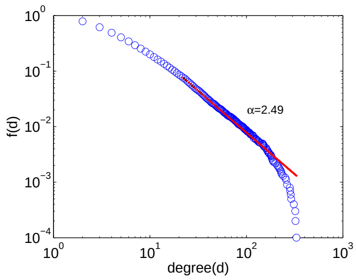
SYNTHETIC NETWORKS AS USED FOR CONDUCTING EXPERIMENTS TO OBTAIN THE AVERAGE VALUES OF SEARCH EFFICIENCY AND COST.

Network	Nodes and edges	Properties
A1	N=10^4, E=50066	$\langle K\rangle$=10.01, α=3.26
A2	N=10^4, E=50032	$\langle K\rangle$=10.00, α=3.03
A3	N=10^4, E=49739	$\langle K\rangle$=9.95, α=2.86
A4	N=10^4, E=29857	$\langle K\rangle$=5.97, α=3.07
A5	N=10^4, E=30241	$\langle K\rangle$=6.05, α=3.50
A6	N=10^4, E=59380	$\langle K\rangle$=11.88, α=2.60
A7	N=10^4, E=60238	$\langle K\rangle$=12.05, α=3.50
A8	N=10^4, E=42373	$\langle K\rangle$=8.47, α=2.40
A9	N=10^4, E=72109	$\langle K\rangle$=14.42, α=2.23

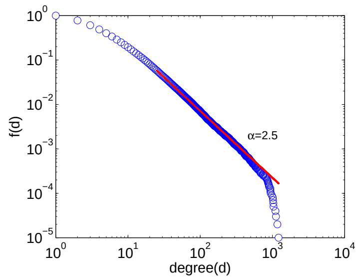
Besides the synthetic networks, we will use five publicly available benchmark networks, including two email networks, two autonomous systems networks (i.e., AS networks), and one typical community-based network. The email networks are Andrew Fiore and Jeff Heer's Enron email network[1] (i.e., NET3) and the University email network[2] (i.e., NET4) complied by the members of University Rovira i Virgili (Tarragona). The AS networks are NET5 and NET6. The

[1]http://bailando.sims.berkeley.edu/enron/enron.sql.gz

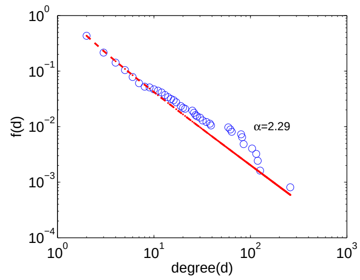
[2]http://deim.urv.cat/$\sim$aarenas/data/welcome.htm



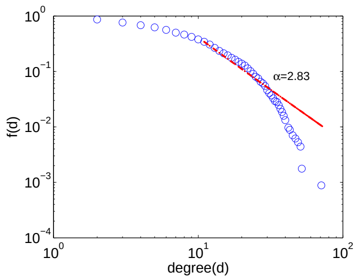
(a) NET1



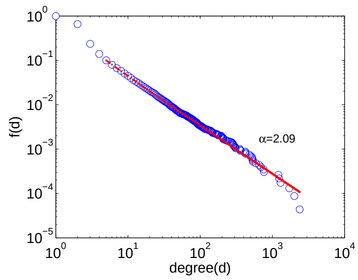
(b) NET2



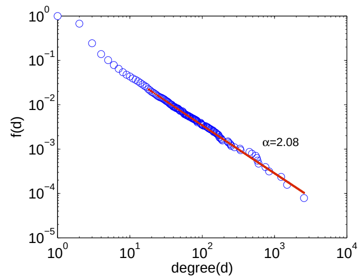
(c) NET3



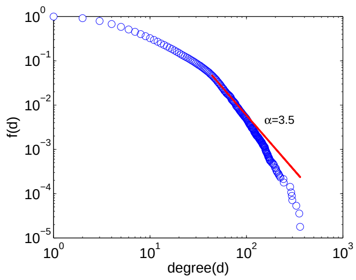
(d) NET4



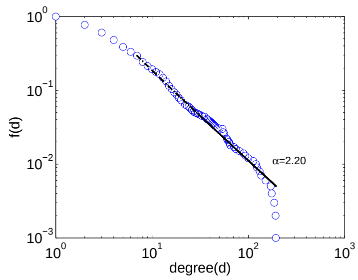
(e) NET5



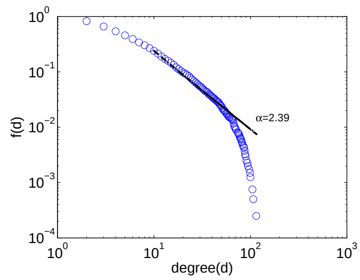
(f) NET6



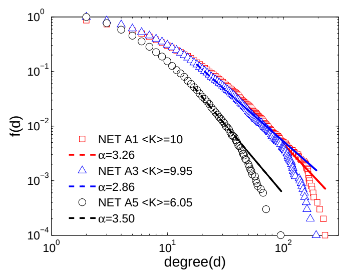
(g) NET7



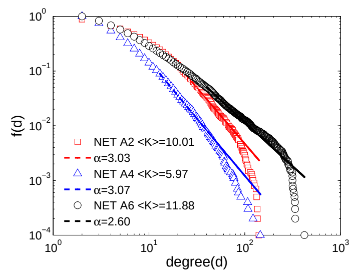
(h) NET8



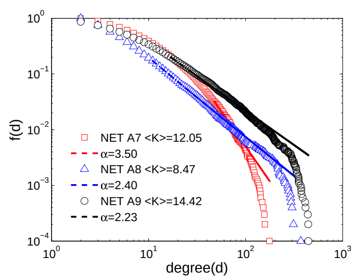
(i) NET9



(j) A1A3A5

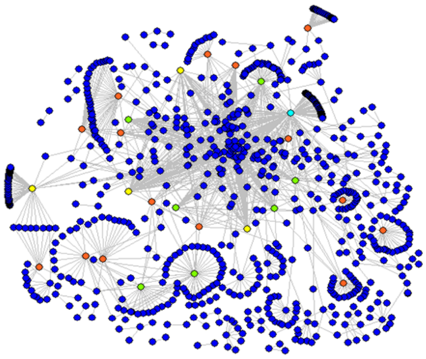


(k) A2A4A6

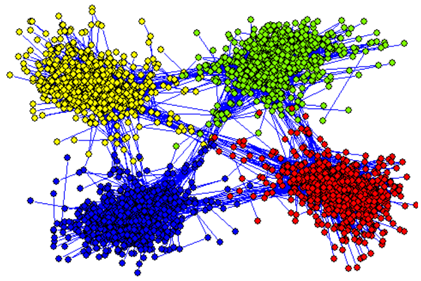


(l) A7A8A9

Fig. 6. The cumulative distribution function f(d) and their maximum likelihood power-law fits for the data sets as used in our experiments (shown in logarithmic charts). The structures of Figs. 6(j), 6(k) and 6(l) are listed in Table III, as used for obtaining the average values in Figs. 8h, 9h, 11a, and 12a



(a) NET3



(b) NET9

Fig. 7. The structures of the Enron email network (NET3) and synthetic community-based network (NET9).

raw data of NET5 comes from the University of Oregon Route Views Project[3] and the snapshot was created by Newman in 2006[4]. NET6 was created by Vaishnavi Krishnamurthy[5]. The typical community-based network is the coauthorship network (i.e., NET7). It was built based on the published papers on the widely-used Physics E-print Archive at arxiv.org. The structure of the Enron email network is shown in Fig. 7(a). And, the degree distributions of benchmark networks are shown in Fig. 6.

As mentioned by Guimerá, there exist some self-similar community structures in the University email network (NET4) [24]. Meanwhile, Newman has pointed out that the coauthorship network consists of about 600 small communities and 4 large communities [25]. In our current work, these networks are used to evaluate the search efficiency and robustness of the AOC-based strategy.

B. The Coverage Efficiency and Analysis

1) Coverage Efficiency:

We have examined coverage rates both in synthetic networks and benchmark networks. Fig. 8 and Fig. 9 present, respectively, the results of coverage rates and average entity steps for different numbers of immunized nodes (entities). It can be noted that the AOC-based strategy can find most of the highest-degree nodes in various networks with fewer steps.

[3]http://routeviews.org/

[4]http://www-personal.umich.edu/∼mejn/netdata/

[5]http://www.cs.ucr.edu/%7Evkrish/

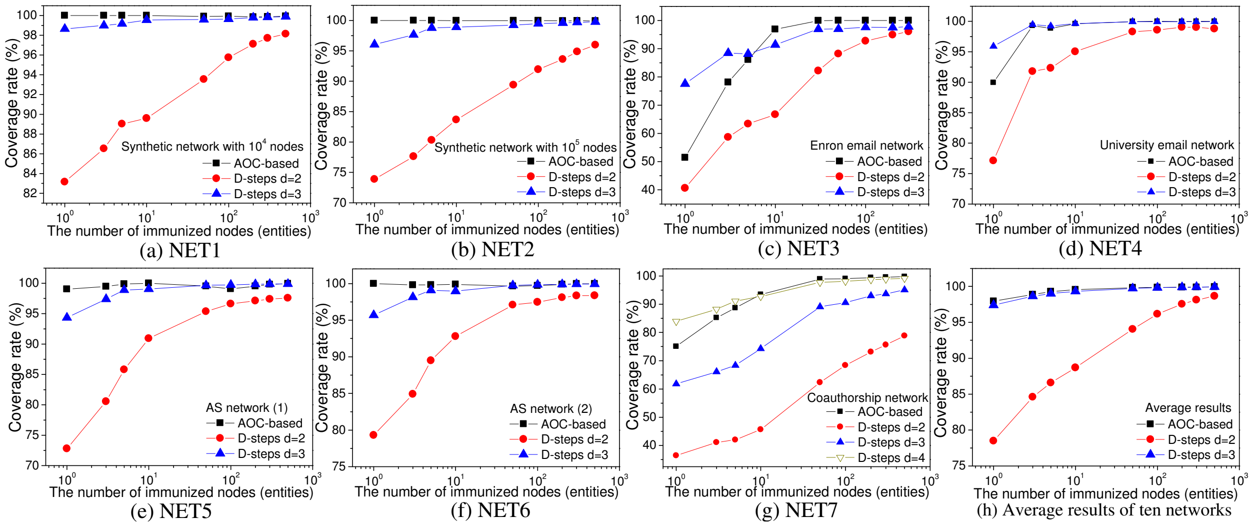


Fig. 8. The coverage rate on different networks (semilog charts). The AOC-based strategy leads to better results than the D-steps strategy. The results in Fig. 8(h) are averaged over NET1 and nine networks as listed in Table III.

There can be seen an obvious *inflection point* in Fig. 9(c)(g). This is the result of indirect interaction-based *positive feedback* among autonomous entities. If there are only a few entities in a network based on AOC-based strategy, 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. 8 and 9), 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.

Fig. 8(g) presents comparison results for the coauthorship network which has typical community-based structures. The results have further confirmed our analysis; that is, the AOC-based strategy can overcome the effect of the community structure. Similar to the case of other networks, there also exists an *inflection point* in Fig. 9(g). The coverage rate of the D-steps strategy (d=4) is close to that of the AOC-based strategy, when both the step length and the number of seed nodes are large enough. However, the computational cost (time complexity) of the D-steps strategy will be much more than that of the AOC-based strategy. A comparison of time complexities will be given in Section III-C.

2) Emergent Computing by Positive Feedback and Self-Organization:

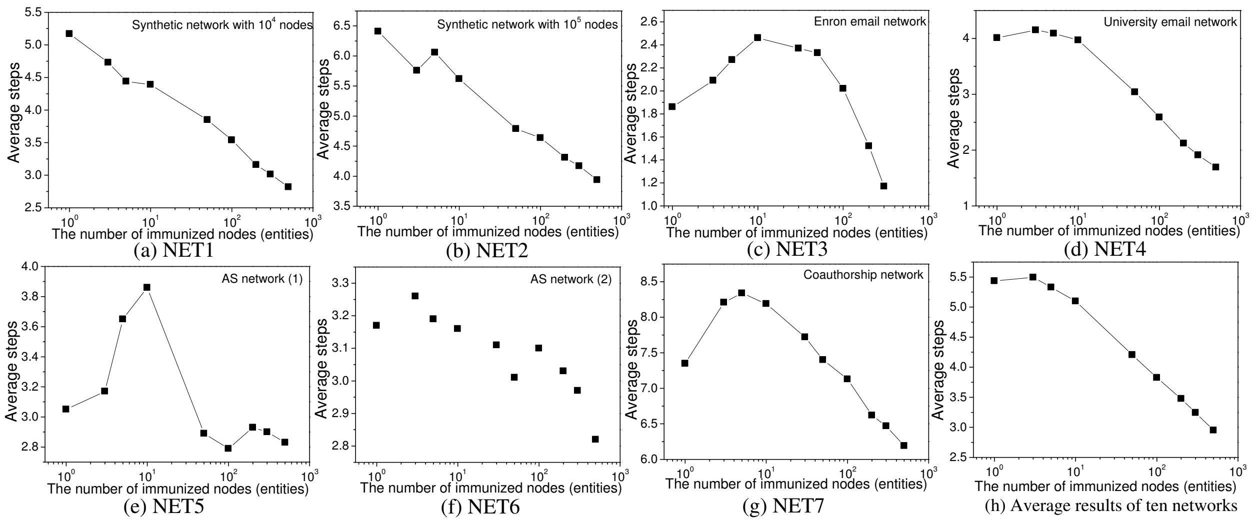


Fig. 9. The average moving steps on different networks (semilog charts). The results in Fig. 9(h) are averaged over NET1 and nine networks as listed in Table III. All entities need only a few steps to find the targeted positions.

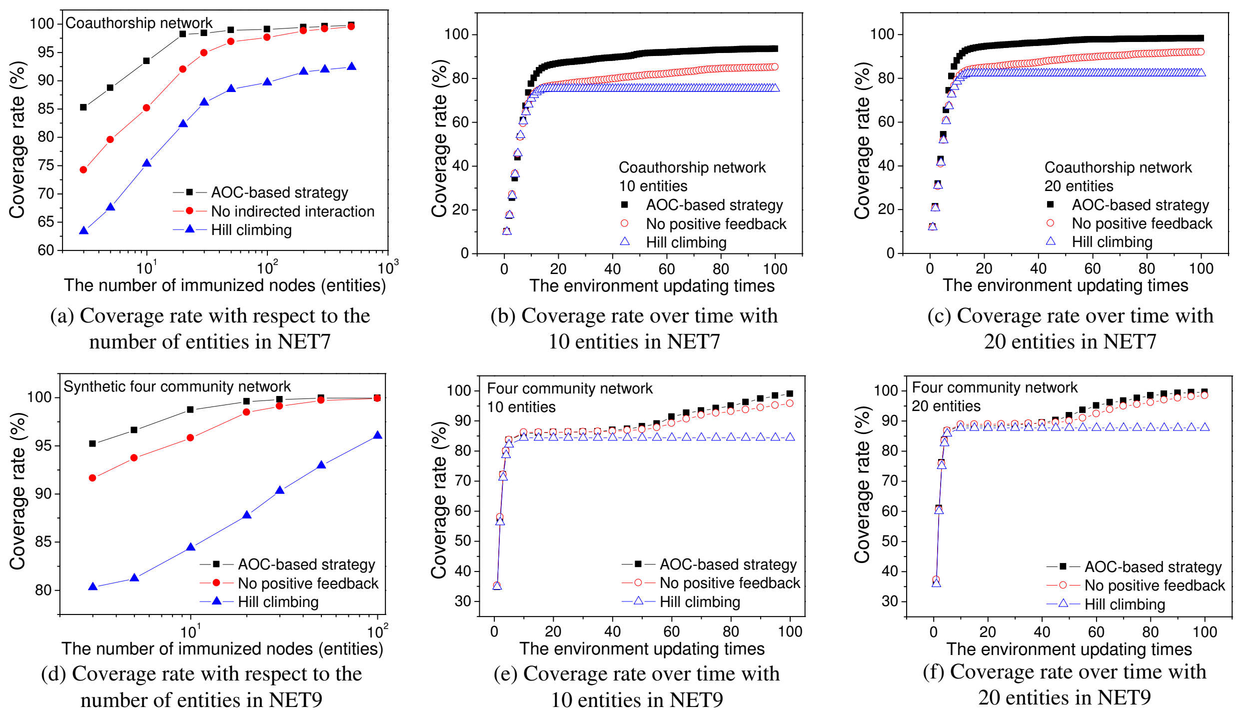


Fig. 10. The influence of positive feedback in the coauthorship network (NET7) and a synthetic community network (NET9). (a, d) are semilog charts and (b, c, e, f) are linear charts. The triangles correspond to the case of applying a hill climbing method, i.e., there are no jump behaviors in the strategy. 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.

In the AOC-based search strategy, the emergence of self-organized local search is due to an indirect interaction between entities and their local environments. The shared local environment of entities are the core of positive feedback. From an AOC system point of view, with respect to this distributed search in a complex network, the *inputs* are a set of entities which reside in different nodes. The *outputs* are a set of new positions for entities and new local environments for entities. In the next step, these outputs are used in the AOC system as new inputs. Because the new inputs (e.g., the utilities from local environments) are non-linearly strengthened in the previous update, the system can achieve a better performance (e.g., finding other higher nodes). That is, the entities will find more highly-connected nodes in the next step, and thus the whole system quickly converges to a steady state [26]. And, Fig.2 in Ref. [26] shows that various collective effects can be reached and observed over either different temporal and/or spatial scales.

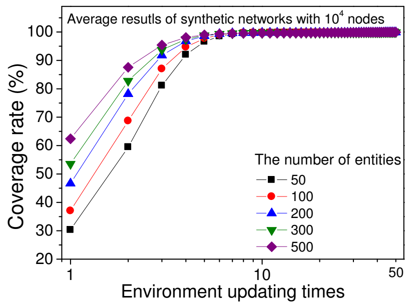
The influences of positive feedback on the coauthorship network and a synthetic community-based network are shown in Fig. 10. The results show that the AOC-based strategy embodies a positive feedback mechanism to speed up the search for desired positions, i.e., all entities can quickly converge to a stable position that has a high degree, which is quite useful for obtaining an approximate search solution with a hard deadline.

In the above-mentioned self-organization and positive feedback, the *rational-move* behavior is dominant throughout the process. The distributions of different behaviors are shown in Fig. 3 and 4(a) in Ref. [26]. The results show that most of the entities are encouraged to move to higher-degree neighbors based on local information. There are only a few *random-jump* behaviors which help entities jump out of local optima as well as realize the diversity of solutions. This conclusion also can be used to explain why the real sequential runtimes of AOC-based strategy is lower than that of D-steps strategy (as shown in Fig. 12) even if the computational complexity of AOC-based strategy is higher than that of D-steps strategy.

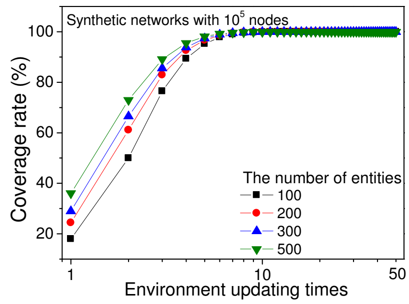
In order to examine the changes of coverage rates with respect to environment updating times (steps), varying numbers of entities are allocated into different networks. The results are shown in Fig. 11.

C. Comparisons of Computational Complexities

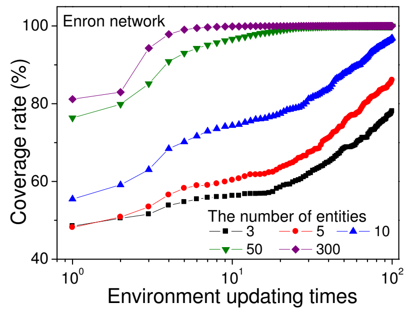
In this section, we will further examine the worst-case computational complexities of the existing immunization strategies. And then, some experiments are used to validate the computational



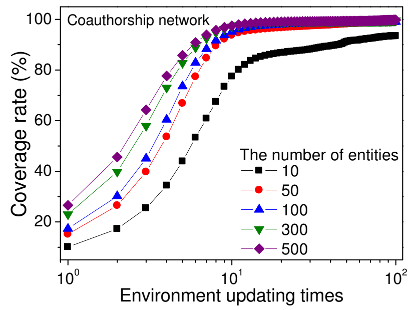
(a) Average results of ten networks



(b) NET2



(c) NET3



(d) NET7

Fig. 11. The coverage rates with respect to the updating steps on different networks (semilog charts). The coverage rate with a larger number of entities reaches higher than those with fewer entities. (a) is averaged result over NET1 and nine networks as listed in Table III

cost of AOC-based strategy. Several parameters involved in the descriptions are listed in Table I.

1) Analysis of Computational Complexities:

As mentioned earlier, the *targeted immunization* and *betweenness immunization* [27] strategies are centralized algorithms. Both of them require global information about the degrees of nodes in order to rank them through some sorting algorithms or compute the shortest paths of a network. Take the bubble sort and a faster algorithm based on betweenness centrality, introduced in [28], as examples, the worse-case computational complexity of the targeted immunization and betweenness immunization will become $O(N^2)$ and $O(N * E)$.

Based on the definition of the *acquaintance immunization*, it selects k neighbors of one node. If the neighbors are randomly selected, i.e., *random neighbor immunization*, the computational complexity will be $O(n * k)$. If we select k highest-degree neighbors of one node, i.e., *max neighbor immunization*, the computational complexity will be $O(n * (\langle K \rangle^2 + k))$. Because it is difficult to know how many neighbors one node has, the network degree $\langle K \rangle$ is here used to replace the average number of neighbors.

The *D-steps immunization* enlarges the search scope from direct neighbors to indirect neighbors. It selects the node with the highest degree around seed nodes in *d* steps. When *d*=1, the D-steps immunization can be seen as max neighbor immunization with *k*=1, the computational complexity is $O(n * (\langle K \rangle^2 + 1))$. When $d > 1$, each step will increase $n * \langle K \rangle$ nodes. Thus, the computational complexity is $O(n * (\langle K \rangle^{d-1}) * (\langle K \rangle^2 + 1))$.

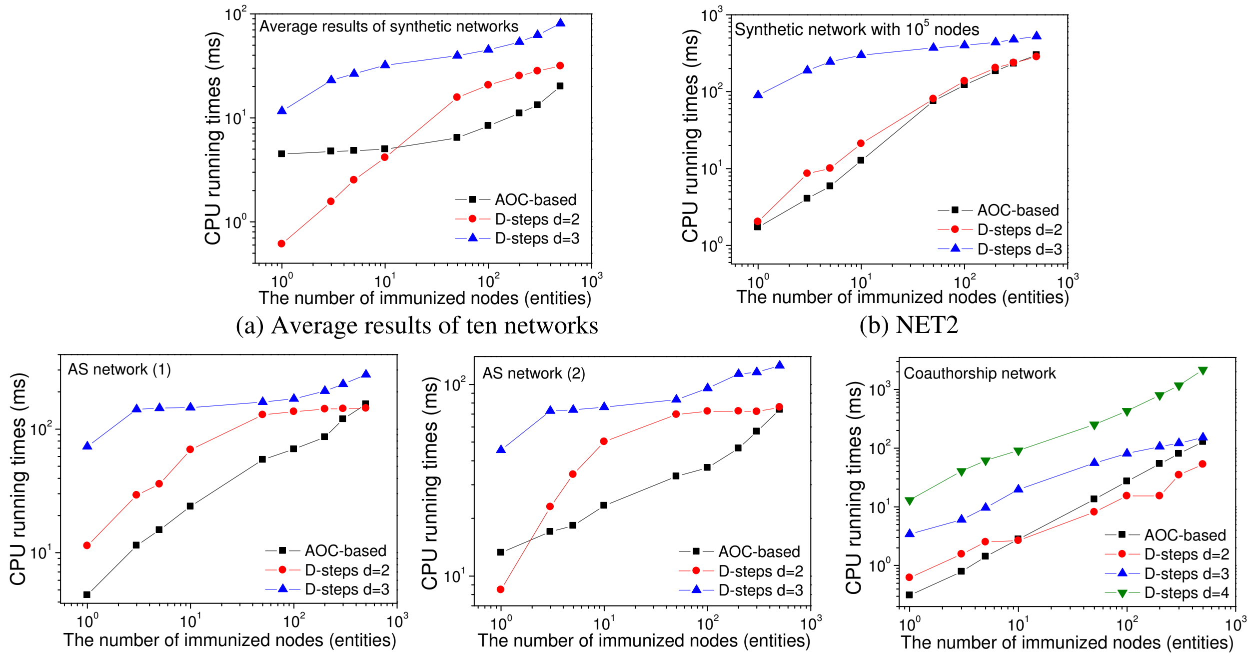
The *AOC-based immunization* self-organizes entities' behaviors and selects a rational move or a long-range move based on dynamically-adjusted probabilities (refer to Section II). The rational move selects one node with the highest degree from its direct neighbors. And, the long-range move randomly selects one node in the network. At worst, assume that the whole environment needs $\overline{p}$ times of updating, in order to converge to a globally stable state. Therefore, the computational complexity is $O(n * \overline{p} * (\langle K \rangle^2 + 1))$. The number of updating times $\overline{p}$ changes with respect to the network scale and structure.

2) Computational Results:

We have selected some large-scale networks for comparing both the sequential and distributed time of the D-steps and AOC-based strategies. The networks include: two AS network with 22963 and 12741 nodes, respectively, coauthorship network with 56276 nodes, synthetic network with 10^5 nodes, and ten synthetic networks with 10^4 nodes.

From the above-mentioned experimental results (e.g., Fig. 11), we have found that the coverage rate of the AOC-based strategy is stable after the strategy updates more than 25 times. And, it will slightly increase after updating 50 times. Therefore, we measure the sequential time in the cases of benchmark networks for updating 50 times and synthetic networks for updating 30 times with the AOC-based strategy. The results are shown in Fig. 12. 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, especially in the cases of large-scale networks. Since the D-steps immunization strategy needs more recursive computations, its computational complexity can be quite high. On the other hand, direct computations among local neighbors can reduce the complexity of the AOC-based strategy.

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 coauthorship network, $\langle K \rangle$ is 11, and system updating time $\overline{p}$ is equal to 50. In this case, the runtime of the AOC-based strategy



(a) Average results of ten networks (b) NET2

(c) NET5 (d) NET6 (e) NET7

Fig. 12. The sequential cost with respect to the number of immunized nodes on the synthetic and benchmark networks. The results in (a) are averaged over NET1 and nine networks listed in Table III. In order to distinctly show the change of runtime, we use logarithmic charts. The AOC-based strategy updates 30 ad 50 times, respectively.

is more than that of the D-steps strategy (when d=2), but less than that of the D-steps (when d=3 and 4). However, the above analysis is based on the worst case. In the actual operations, the computing time of each entity is less than $\overline{p}$. Fig. 12e shows that the cost of the AOC-based strategy is lower than that of the D-steps strategy (when d=2) in the initial stage.

Take NET2 as an example, one node is selected as the seed node. The AOC-based strategy compares $\overline{p}$ times to select the destination node. In the D-steps strategy, it only needs to compare once if d=1. However, with the increase of steps d, more recursions are needed in order to find the highest-degree node. Each step with the D-steps strategy will increase $\langle K \rangle$ nodes. So, the D-steps strategy needs to compare $\langle K \rangle^{d-1}$ times in total. If there are 10 seed nodes in NET2, the theoretical comparison times of the AOC-based strategy are more than that of the D-steps strategy, by comparing $\overline{p}$ with $\langle K \rangle^{d-1}$ in $O(n*(\langle K \rangle^{d-1})*(\langle K \rangle^2+1))$ and $O(n*\overline{p}*(\langle K \rangle^2+1))$, where $\overline{p}$=8.51 and $\langle K \rangle$=8. However, the actual moving steps of each entity is fewer than $\overline{p}$, based on our comparisons both in benchmark and synthetic networks, as well as the analysis in

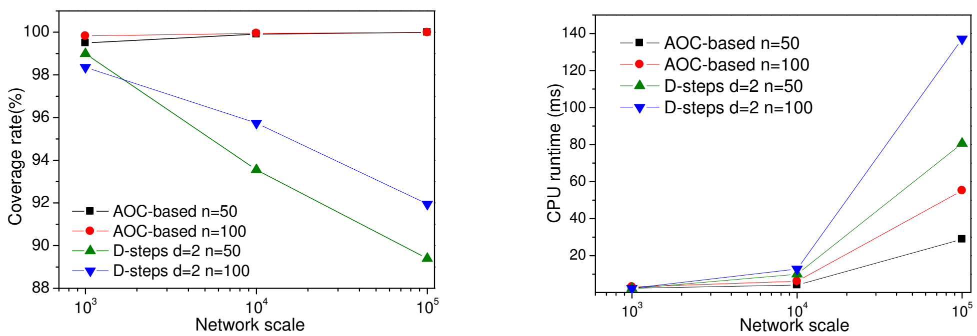
Section III-B.

D. Robustness and Immunization Efficiency

Due to space constraints, the robustness of AOC-based strategy is evaluated in Ref. [29]. The results have shown that autonomous entities can collectively find and immunize the most of highly-connected nodes in dynamically-evolving, community-based networks within a few steps. At the same time, we also validated the immunization efficiency of the AOC-based strategy by measuring the number of final infected nodes in an interactive propagation model, as compared to the targeted immunization strategy and the D-steps immunization strategy. The results in Ref. [29] have shown that the AOC-based strategy can more effectively restrain virus propagation.

E. Scalability

As a supplementary file, we provide more computational results to illustrate the scalable capability of AOC-based strategy. Fig. 13 compares the coverage rate and computational cost of the AOC-based strategy with that of the D-steps strategy (d=2).



(a) Coverage rate with respect to the network scales

(b) Computational cost with respect to the network scales

Fig. 13. Scalability evaluations of the AOC-based and D-steps strategies (semilog charts). The coverage rate of the D-steps decreases 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 in the larger network. The cost of the D-steps drastically increases with respect to the increments of network scales. However, the AOC-based strategy can get a higher coverage rate with a lower cost in a larger network.

IV. Conclusions

In this paper, we have presented a distributed and feasible immunization strategy that is based on a novel computing paradigm, called Autonomy-Oriented Computing (AOC) in order to meet the requirement of reducing computational complexity in the cases of large-scale, decentralized networks. Technically speaking, the problem of decentralized network immunization can be translated into the problem of how to most effectively and efficiently prevent a virus from turning into an epidemic, given a minimum number of vaccinated nodes. Therefore, the AOC-based strategy combines the self-organization and positive feedback into the search process in order to find the highest-degree nodes in decentralized and dynamic networks.

The AOC-based strategy deploys a group of computational entities into a distributed environment, which can autonomously interact with each other and update their local environments. Each entity aims to search and reside on the highly-connected node based on the notion of its own profit and the positive feedback from its local environment. The results have shown that:

1) The AOC-based strategy can find most of the highest-degree nodes within a few steps not only in large-scale, decentralized and dynamic networks, but also in community-based networks, which is quite useful for providing an approximate solution given a hard deadline.
2) The AOC-based strategy has the ability to scale up its computation based on the design of self-organization. The efficiency of AOC-based strategy will nonlinearly grow based on the indirect interactions, i.e., their indirect interaction-based positive feedback mechanism resulting in nonlinear performance improvement.
3) The immunization efficiency of the AOC-based strategy is even better than the targeted immunization strategy sometimes. That is because the targeted immunization strategy only selects immunized nodes by sorting and does not distinguish the importance of nodes. However, the AOC-based strategy will select both the highly-connected nodes and those nodes with higher connectivity and transmission capability.

References

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

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

[8] K. Sycara, S. Roth, N. Sadeh, and M. Fox, "Distributed constrained heuristic search," *IEEE Transactions on System man and Cybernetics*, vol. 21, no. 6, pp. 1446–1461, Nov/Dec. 1991.

[9] J. Liu, X. Jin, and K.C. Tsui, *Autonomy Oriented Computing (AOC): From Problem Solving to Complex Systems Modeling (Multiagent System, Artificial Societies, and Simulated Organizations)*. Boston: Kluwer Academic Publishers, 2005.

[10] J. Liu, S. Zhang, and J. Yang, "Characterizing web usage regularities with information foraging agents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 5, pp. 566–584, May 2004.

[11] J. Liu and K.C. Tsui, "Toward nature-inspired computing," *Communications of the ACM*, vol. 49, no. 10, pp. 59–64, 2006.

[12] X.F. Xie and J. Liu, "Multiagent optimization system for solving the traveling salesman problem (TSP)," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 2, pp. 489–502, Apr. 2009.

[13] B. Yang, J. Liu and D. Liu, "An autonomy-oriented computing approach to community mining in distributed and dynamic networks," *Autonomous Agents and Multi-Agent System*, vol. 20, no. 2, pp. 123–157, Mar. 2010.

[14] J. Liu, "Autonomy-oriented computing (AOC): The nature and implications of a paradigm for self-organized computing (keynote talk)," *Proceedings of the 4th International Conference on Natural Computation (ICNC'08) and the 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'08)*, pp. 3–11, Oct. 2008.

[15] O.J. Mengshoel, "Understanding the role of noise in stochastic local search: Analysis and experiments," *Artificial Intelligence*, vol. 172, no. 8-9, pp. 955–990, May 2008.

[16] S. Climer and W.X. Zhang, "Cut-and-solve: An iterative search strategy for combinatorial optimization problems," *Artificial Intelligence*, vol. 70, no. 8-9, pp. 714–738, Jun. 2006.

[17] R.D. Hughes, *Random Walks: Random walks and Random Environments*. Clarendon: Oxford, 1995.

[18] V.P. Janeja and V. Atluri, "Random walks to identify anomalous free-form spatial scan windows," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 10, pp. 1378–1392, Oct. 2008.

[19] B. Yang and J. Liu, "Discovering global network communities based on local centralities," *ACM Transactions on the Web*, vol. 2, no. 1, pp. 1–32, Feb. 2008.

[20] J.D. Noh and H. Rieger, "Random walks on complex networks," *Physical Review Letters*, vol. 92, no. 11, p. 118701, 2004.

[21] A. Clauset, C.R. Shalizi, and M.E.J. Newman, "Power-law distribution in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, Feb. 2009.

[22] A.L. Lloyd and R.M. May, "How viruses spread among computers and people," *Science*, vol. 292, no. 5520, pp. 1316–1317, May 2001.

[23] 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, 2002.

[24] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical Review E*, vol. 68, no. 6, p. 065103, Dec. 2003.

[25] M.E.J. Newman, “The structure of scientific collaboration networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 2, pp. 404–409, Jan. 2001.

[26] J. Liu, C. Gao, and N. Zhong, “An autonomy-oriented paradigm for self-organized computing,” *Proceedings of 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies*, pp. 100-103, Sept. 2009.

[27] P. Holme, B.J. Kim, C.N. Yoon, and S.K. Han, “Attack vulnerability of complex networks,” *Physical Review E*, vol. 65, no. 5, p. 056109, May 2002.

[28] U. Brandes, “A faster algorithm for betweenness centrality,” *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, May 2001.

[29] 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.