Metadata of the chapter that will be visualized in SpringerLink

Book Title	Neural Information Processing	
Series Title		
Chapter Title	Resilient Consensus for Multi-agent Networks with Mobile Detectors	
Copyright Year	2018	
Copyright HolderName	Springer Nature Switzerland AG	
Author	Family Name	**Yan**
	Particle	
	Given Name	**Haofeng**
	Prefix	
	Suffix	
	Role	
	Division	School of Computer Science and Technology
	Organization	Hangzhou Dianzi University
	Address	Hangzhou, 310018, China
	Email	
Author	Family Name	**Wu**
	Particle	
	Given Name	**Yiming**
	Prefix	
	Suffix	
	Role	
	Division	School of Cyberspace
	Organization	Hangzhou Dianzi University
	Address	Hangzhou, 310018, China
	Email	
Corresponding Author	Family Name	**Xu**
	Particle	
	Given Name	**Ming**
	Prefix	
	Suffix	
	Role	
	Division	School of Cyberspace
	Organization	Hangzhou Dianzi University
	Address	Hangzhou, 310018, China
	Email	mxu@hdu.edu.cn
Author	Family Name	**Wu**
	Particle	
	Given Name	**Ting**
	Prefix	
	Suffix	

	Role	
	Division	School of Cyberspace
	Organization	Hangzhou Dianzi University
	Address	Hangzhou, 310018, China
	Email	
Author	Family Name	**Xu**
	Particle	
	Given Name	**Jian**
	Prefix	
	Suffix	
	Role	
	Division	School of Computer Science and Technology
	Organization	Hangzhou Dianzi University
	Address	Hangzhou, 310018, China
	Email	
Author	Family Name	**Qiao**
	Particle	
	Given Name	**Tong**
	Prefix	
	Suffix	
	Role	
	Division	School of Cyberspace
	Organization	Hangzhou Dianzi University
	Address	Hangzhou, 310018, China
	Email	

Abstract	This paper investigates the problem of resilient consensus for multi-agent systems under malicious attacks. Compared with most of existing works, a more flexible network topology scheme is considered, where a kind of specific agents as the mobile detectors and builders of network robustness are adopted. Specifically, the mobile agents can perceive the message of their nearby agents in the dynamic network, and acquire both in-degree and state information of each node as characteristics to judge the network state as well as communication links between nodes. It is shown that even in poor network robustness, the non-faulty agents can still achieve a consensus in finite time with the help of mobile agents. Finally, the simulation results show the effectiveness of the proposed method.

Keywords (separated by '-')	Resilient consensus - Network security - Mobile detector

Resilient Consensus for Multi-agent Networks with Mobile Detectors

Haofeng Yan[1], Yiming Wu[2], Ming Xu[2(✉)], Ting Wu[2], Jian Xu[1], and Tong Qiao[2]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2] School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
mxu@hdu.edu.cn

Abstract. This paper investigates the problem of resilient consensus for multi-agent systems under malicious attacks. Compared with most of existing works, a more flexible network topology scheme is considered, where a kind of specific agents as the mobile detectors and builders of network robustness are adopted. Specifically, the mobile agents can perceive the message of their nearby agents in the dynamic network, and acquire both in-degree and state information of each node as characteristics to judge the network state as well as communication links between nodes. It is shown that even in poor network robustness, the non-faulty agents can still achieve a consensus in finite time with the help of mobile agents. Finally, the simulation results show the effectiveness of the proposed method.

Keywords: Resilient consensus · Network security · Mobile detector

1 Introduction

With high robustness and strong flexibility, distributed computation plays a key role in multi-agent systems [1–4]. As one of the most effective methods for distributed computation, consensus means that nodes in the network achieve an agreement on a certain state variable by using local information. Most of existing works assume that all agents perform the algorithm faithfully with the prescribed update rules. However, the multi-agent systems are usually deployed in a real-world environment, nodes may update with outliers due to failures or cyber attacks, thus these existing consensus algorithms could become vulnerable or even invalid.

Recently, a family of consensus algorithms named Mean Subsequence Reduced (MSR) algorithms is proposed in [5–7]. In the MSR algorithms, each node disregards the smallest and largest F values collected from its neighbors and then updates its own state to be an average of the remaining values. However, MSR algorithms need to be run on a system satisfying a particular network

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11307, pp. 1–12, 2018.
https://doi.org/10.1007/978-3-030-04239-4_26

topology property called network robustness [16]. Some works analyze and develop this topology property for multi-agent systems in the presence of misbehaving agents [8,17].

In this paper, we attempt to implement MSR algorithm in more general topologies. Specifically, we design a novel method to reduce the dependence of complex graph topology by using clustering method. We first analyze the characteristics of running MSR algorithm in general networks. According to these characteristics, networks are decomposed into some subunits, and then, we adopt a mobile node to identify different subunits and act as links between them. Finally, non-faulty nodes receive the information from the mobile nodes and add it to their own update data set on the basis of the weight.

The rest of paper is organized as follows. Section 2 describes related works. In Sect. 3, we give the problem formulation and the background related to this paper. And in Sect. 4, the details of our method are given. Section 5 shows some simulations. Finally, we conclude this paper with a short review in Sect. 6.

2 Related Work

Consensus control in multi-agent systems with malicious agents has attracted increasingly research interests [10,11]. There are two types of methods to solve this problem: One is fault detection and isolation, for example, Zhao et al. [12] exploit the mobile agents as the detector, and design the protocol with mobile detector called mobile resilient consensus algorithm (MRCA) to detect malicious nodes. The other is fault (or attack) tolerance algorithms, such as the MSR-type algorithms. These algorithms are able to mitigate the effects of malicious agents without the need for non-faulty nodes to explicitly identify the sources of attacker [6,8–10].

2.1 Analyze MSR-Type Algorithms in General Topologies

In [14], the authors consider a large and sparsely connected network, and give some expressions of local convergence in local networks under two distinct fault models. In addition, the authors give a methodology for analyzing global network convergence properties. In [15], the authors give the necessary and sufficient conditions for the MSR-type algorithm to achieve resilient Byzantine consensus in arbitrary directed graphs. And in [16], authors propose a concept of r-robust graph and show that this concept provides the condition for achieving distributed resilient consensus goals. Authors in [8] summarize the work of the predecessors and exploit a novel graph-theoretic property, named *network robustness*. They indicates that traditional properties such as connectivity are not sufficient to support the operation of MSR algorithms. Moreover, in [17], the authors prove that determining the robustness of the given network is NP-complete.

2.2 Community Detection in Networks

Community detection can help us to discover the topics of information networks or cyber-communities of social networks. In [18], Radicchi et al. introduce a divisive algorithm that detect inter-community links and then remove these links from the graph. In [19], authors propose a fast hierarchical agglomeration algorithm for optimizing the modularity of networks. Another agglomerative algorithm which merges similar nodes recursively is proposed by Pons et al. in [20]. In addition, Vincent et al. [21] propose a method based on modularity optimization to extract the community structure of large networks.

3 Preliminary and Problem Statement

In this section, we introduce some fundamental matters related to graph theory, the scope of threats and the concepts of resilient consensus.

Notations. A directed graph is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = 1, ..., n$ is the node set, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The edge $(j, i) \in \mathcal{E}$ indicates that information flows from node j to node i, which is called an incoming edge of node i. The node with the edge pointing to node i is referred to as a neighbor of node i, and the set of the entire neighbors of the node i is denoted by $\mathcal{J}_i = \{j : (j, i) \in \mathcal{E}\}$. The number of neighbors that node i has is called in-degree, which is denoted as $d_i = |\mathcal{J}_i|$.

As mentioned in the Introduction, some works focus on a graph property known as r-robust, which is given by the following definitions from [8] and [16] for analysis of resilient consensus of multi-agent systems.

Definition 3.1: For digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is (r, s)-robust $(r, s < n)$ if for every pair of nonempty disjoint subsets $S_1, S_1 \subset \mathcal{V}$, at least one of the following conditions holds: AQ1

(1) $\mathcal{X}_{S_1}^r = S_1$,
(2) $\mathcal{X}_{S_2}^r = S_2$,
(3) $|\mathcal{X}_{S_1}^r| + |\mathcal{X}_{S_2}^r| \geq s$,

where $\mathcal{X}_{S_l}^r$ is the entire set of nodes in S_l which have at least r incoming edges from outside S_l. In particular, graphs which are $(r, 1)$-robust are called r-robust.

The following lemma shows the basic properties of the robust graphs [8].

Lemma 1: For an (r, s)-robust graph $\mathcal{G}$, the following holds:

(i) $\mathcal{G}$ is (r', s')-robust, where $0 \leq r' \leq r$ and $1 \leq s' \leq s$, and in particular, it is r-robust.
(ii) $\mathcal{G}$ has a directed spanning tree.
(iii) $r \leq \lceil n/2 \rceil$, where $\lceil \cdot \rceil$ is the ceiling function. Also, if $\mathcal{G}$ is a complete graph, then it is (r', s)-robust for all $0 < r' \leq \lceil n/2 \rceil$ and $1 \leq s \leq n$.

Moreover, a graph $\mathcal{G}$ is (r, s)-robust if it is $(r + s - 1)$-robust.

It is clear that (r, s)-robustness is more restrictive than r-robustness. Consider a network with five agents as shown in Fig. 1, which satisfies a $(2, 1)$-robust graph. We can also name it a 2-robust graph. And taking a closer look at this graph, for any pairs of disjoint, nonempty subsets of nodes in the graph, we can see that at least one node in the subset would be sufficiently influenced by two nodes outside its set (thus we could only remove one node which value is abnormal compared with its own). This would drive it away from the values of its subset, and thereby allow it to lead its subset to the values of the other set. Moreover, the consensus will fail if more than one node is abandoned. This causes no node has enough neighbors in the outside set, every node throws away all information from outside of its set.

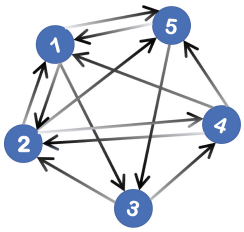


Fig. 1. A (2,1)-robust graph with five nodes.

3.1 Threat Model

In this paper, we consider there should be an upper bound on the number of malicious nodes either in each nodes' neighborhood (f-local) or in whole network (f-total), the definitions are as follows:

Definition 3.2 (f-local model): A node $i \in \mathcal{V}$ is called f-local if the number of malicious nodes in the neighborhood $\mathcal{J}_i$ of each node i is no greater than f, $\forall f \in \mathbb{Z}_{\geq 0}$.

Definition 3.3 (f-total model): A node $i \in \mathcal{V}$ is called f-total if the number of malicious nodes in the network is no greater than f, $\forall f \in \mathbb{Z}_{\geq 0}$.

3.2 Problem Formulation

Now we define the concept of resilient consensus as follows:

Definition 3.4: It is called reaching a resilient consensus if all normal nodes satisfy the following two conditions, for any initial values and malicious inputs:

(1) Safety condition: There exists a bounded interval S defined by the initial α of the normal nodes, and $\alpha_i[t] \in S$, $\forall i \in \mathcal{V} \setminus \mathcal{M}, t \in \mathbb{Z}_{\geq 0}$;

(2) Consensus condition: The state values of all normal nodes agree on a constant c which satisfies $\lim_{t\to\infty} \alpha_i[t] = c$, $\forall i \in \mathcal{V} \setminus \mathcal{M}, t \in \mathbb{Z}_{\geq 0}$.

When applying update MSR-type algorithms to node in a network which satisfies r-connected but not r-robust, we found that all normal nodes will unable to reach consensus under the f-local threat model ($2f+1 \leq r$). Thus, our goal is to design a method to ensure that nodes can reach consensus when the network only satisfies r-connected.

4 Algorithm Description

We now introduce our method to achieve resilient consensus for multi-agent systems in adversarial environment. The algorithm can be able to apply to a more general network topology situation, which is composed of Average Iteration Algorithm, MSR algorithms for normal nodes and Mobility Detection Algorithms for the mobile nodes. Specifically, the framework of our method is shown in Fig. 2, First of all, after receiving the message from its neighbors, each normal node first uses MSR algorithm to eliminate outliers. Then, the remaining message is utilized for the update according to the iteration rule. In addition, mobile nodes move around randomly according to the algorithm, which collect information from node in network and finds high modularity partitions of large networks. Finally, the mobile nodes act as link between subunits. The detailed information is shown in Algorithm 1.

The lower half of the algorithm is allocated to each node to run separately, and the detailed process about this part will be shown in Sect. 4.1. The other part of the algorithm is run by mobile nodes, and the detailed process well be shown in Sect. 4.2, which space and time complexities are both $O(e)$.

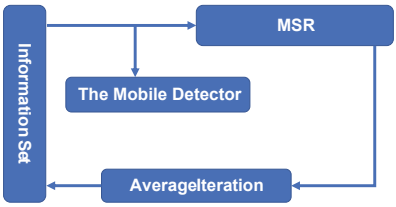


Fig. 2. The framework of the proposed method.

4.1 Average Iteration Algorithm and MSR Algorithm

Each normal node receives the values of neighbors at every time-step t, and according to our attack model, there are at most f of node's neighbors may be malicious nodes. In traditional MSR, each node is unaware of which neighbors may be attackers. Therefore, node simply removes the extreme values with respect to its own value when updates its value. The details are as follows:

Algorithm 1. Network topology compensation algorithm

Input: $\mathcal{T}$, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{L}\}$, F
Output: a stable set of values $\mathcal{L}$

$\mathcal{L}_{error} \leftarrow \{\}$;
while *The values in the L set are unlikeness* **do**
  if *time.equals(*$\mathcal{T}$*)* **then**
    community detection;
    foreach *community* **do**
      if *community.size* $< F$ **then**
        $\mathcal{L}_{error}\{\} \leftarrow$ for each i in this community{};
      else
        continue;
      end if
    end foreach
    $\mathcal{G}.\mathcal{E} \leftarrow$ new $\mathcal{G}.\mathcal{E}$
  end if
  foreach $i \in \mathcal{V}$ **do**
    $i_{value} \leftarrow \mathcal{L}[i]$;
    $\mathcal{J}_i \leftarrow \mathcal{G}.\mathcal{E}$;
    foreach j *in* $\mathcal{J}_i$ **do**
      if j_{value} *in* $\mathcal{L}_{error}$ **then**
        Delete $\mathcal{G}.\mathcal{L}[j]$;
        $d_i \leftarrow d_i - 1$;
      else
        continue;
      end if
    end foreach
    $\mathcal{J}_i.value \leftarrow$ remove the outliers that differ greatly from i;
    Average Iteration;
    $\mathcal{L}[i] \leftarrow$ new $\mathcal{L}[i]$;
  end foreach
end while
return $\mathcal{L}$

1 At each update time t, each normal node i obtains and sorts $x_j[t]$ $(j \in \mathcal{J}_i)$, which is received from its neighbors.
2 If there are less than f neighbors' value larger than its own value, $x_i[t]$, then normal node i removes all values which are larger than its own. Otherwise, the largest f values are removed. Similarly, if there are less than f neighbors' value less than its own value, node i removes all values that are less than its own. Otherwise, the least f values are removed.
3 Let $R_i[t]$ denote the set of nodes who were removed by i in step 2. Each value of node i at this time-step is updated as:

$$x_i[t+1] = \sum_{j \in \mathcal{J}_i[t] \backslash \mathcal{R}_i[t]} \{w_{ij} x_j[t]\} \tag{1}$$

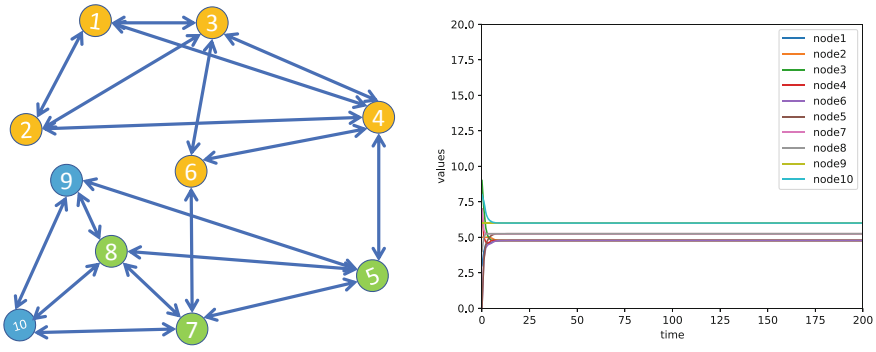
where w_{ij} is the weight of edge from j to i.

As we can see that MSR does not require any node to have knowledge of the identities of non-neighbor nodes. However, it turns out that it needs a specific structure characterized by graph robustness rather than simply possess enough neighbors. Consider this problem, a question we need to answer next is: how to extend this situation to a series of more general networks.

4.2 The MDA Mechanism

The main objective of MDA mechanism is to use community detection method to identify subunits in a multi-agent system and establish links between them.

Critical Phenomena of Networks: We first try to run MSR algorithm on a low-robust network (which set f that $r < 2f + 1$) and observe the changes of all normal nodes' values. From Fig. 3, we can see that as the number of iterations increases, normal nodes in the network form different subunits based on their value and the edge. Nodes in the same subunit have the same value that node 1 and node 2 have the same value 5 and other nodes' value is approximately 7.5. Intuitively, we can find that when each node runs the MSR algorithm, due to the network robustness is insufficient, there will be nodes in one region that cannot communicate with other area. Thus we can acquire a new network topology after running MSR algorithm based on these features and values of the nodes. This network can be divided into areas of densely connected nodes, with the nodes belonging to different areas only sparsely connected.



(a) network (b) value

Fig. 3. Each node's value trajectory in the network with MSR algorithm.

Clustering: In response to the need outlined above, we need a way to find reasonably good partitions in a fast way and establish contact for these partitions. Each agent in a multi-agent system can be regarded as an individual of a social

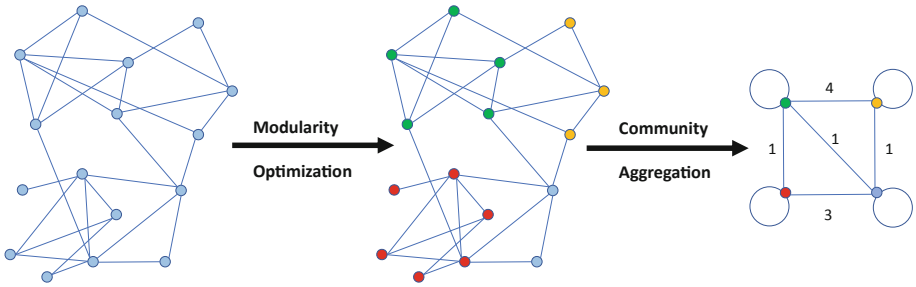


Fig. 4. Visualization of the steps of clustering.

network. The links between the agents can be regarded as communication in the social network. Therefore, we use the community detection method [19] to find a reasonable network partition after using the MSR algorithm. This method is divided in two phases that are repeated iteratively. This method is described as follows:

1 Each node in the network is assigned a different subunit.
2 For each node i consider its neighbors j, try removing i from its subunit and placing it in the subunit of j, evaluate their gain of modularity, and the node i finally placed in the subunit for which this gain is maximum (this gain is positive). If the gain for all neighbors j is negative, i stays in its original subunit.
3 Repeat step 2 until the subunit to which all nodes belong does not change.
4 Build a new network whose nodes whose nodes are now the subunits found during the step 2 and step 3. The weight of the edges between the new nodes is the sum of the weights of the edges between the two previous subunits [22].
5 Repeat the above steps in the new network until there are no more changes and attain the maximum of modularity of the entire network.

The process is shown in Fig. 4, and the formula of gain that a node move in subunit C is provided in [21] as follows:

$$\Delta Q = [\frac{\sum_{in} + k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2] \quad (2)$$

where $\sum_{in}$ denotes the sum of the weights of the edges whose starting and ending points are both in the same subunit C, $\sum_{tot}$ is the sum of the edges which incident to C, k_i denotes the sum of the in-degree with node i, $k_{i,in}$ denotes the sum of the weights of the edges from i to nodes in C and m denotes the sum of the weights of all edges in the network.

In this paper, we do not really need to find the exact value of modularity and just need to know how much the current operating module grows relative to other operations (operations mean move node i into a subunit). So we use the formula to determine relative gain. This formula can reduce the time complexity of the algorithm greatly.

$$\Delta Q = k_{i,in} - \frac{\sum_{tot} \times k_i}{m} \tag{3}$$

And we set the weight of the edge to the reciprocal of the absolute values of difference between two node values.

Mobility Model: We divide the network into multiple regions, in each region, there are mobile nodes that move around randomly according to the protocol in [23]. The mobile node has a large powerful receiver that can receive information broadcast by nodes in the vicinity. In addition, there are mobile nodes in different areas to contact each other. Mobile nodes run the community detection algorithm at a certain period of time based on the information it collects. When it found different subunits, it will help them deliver message. Then we can assume that each mobile node has sufficient mobility so that the nodes in multi-agent network can be contacted with a positive probability in each time-step, which is used in [24]. The contact probability relates to the number of mobile nodes and the frequency of movement. We use the following assumption to simplify the statement.

Assumption 4.1: The probability that each node in the network is in contact with the mobile node is the same and equals ρ,$0 < \rho < 1$, at each time-step.

The network model with mobile nodes is widely used in many fields. For example, mobile nodes in wireless sensor networks can be considered as a secondary node to improve network performance, in addition to wireless charging, sensor coverage, data collection, etc.

5 Simulation

In this section, we design simulations to illustrate the convergence and effectiveness of our method in a more general network.

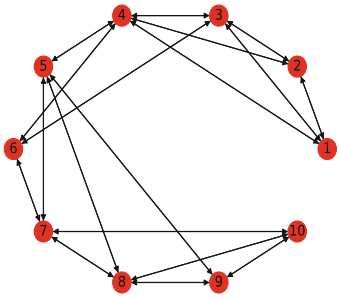


Fig. 5. Network topology.

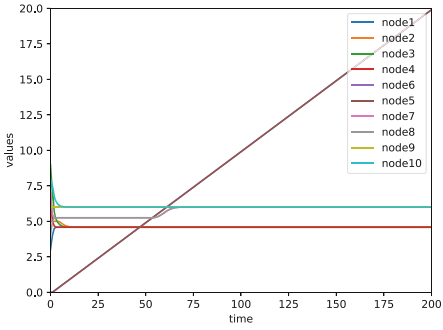


Fig. 6. MSR under 2-local threat model.

We built a $(1,4)$-robust graph as show in Fig. 5, in which the node set is $\mathcal{V} = \{1, 2, ..., 10\}$ and node $i \in \mathcal{V}$ has initial value $x_i(0)$, and $x(0) = [3, 5, 9, 7, 0, 0, 7, 4, 6, 8]^T$. According to [8], the node in this network can run MSR with f-local ($f <= 1$) threat model and reach resilient consensus, if $f > 1$, this network may break down into different subunits and not reach consensus. Now let's set 5 and 6 nodes to be malicious nodes that can inject any false data into its neighborhood at each time-step to break consistency. And next we tried several different forms of data injection to effectiveness. In addition, according to our setup, we can get that h (the number of malicious nodes which neighbor and collude with each other) is 2. The mobile node is set to randomly appear around the nodes in the network with a probability of 0.1 and collects the messages to implement community detection. And when the time period $t = 100$, the mobile node acts as a link between subunits based on the results of the community detection.

As shown in Figs. 6 and 7, the value injected by malicious nodes increases over time, and both of them are not affected by malicious nodes. However, in Fig. 6, we can see that the value of each normal node under the traditional MSR method will be affected by the network robustness. Intuitively, values of normal node tend to be two different values, so that the network can not reach resilient consensus. But average consensus can still be achieved under our method as shown in Fig. 7(a).

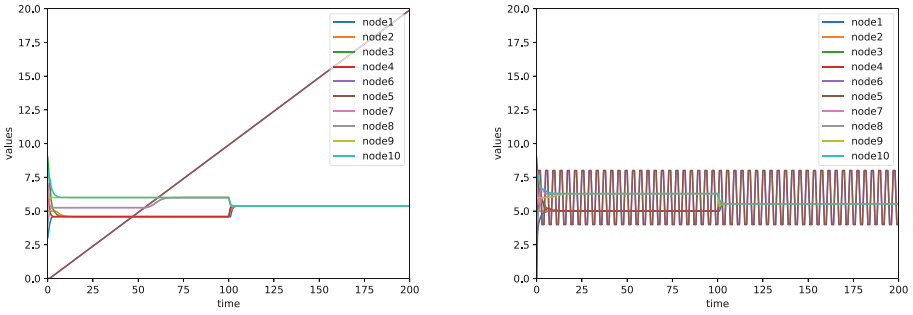


Fig. 7. The performance of the proposed method under 2-local threat model.

Next we tried to make the injected data floating up and down. Figure 7(b) shows that nodes in network can still achieve consensus with our method.

As a result of these examples, we observe that our method can be effective even if the node have different types of malicious nodes as neighbors. By this method, the node in network maintains the advantages of the MSR algorithm, that the node does not detect the received information from neighbors, just needs to eliminate the largest outliers. Moreover, it has lower requirements on the network topology to achieve resilient consensus.

6 Conclusions

In this paper, we proposed a novel MSR-based algorithm to reach resilient consensus for multi-agent systems under attacks. Specifically, we first observed and analyzed the phenomenon of traditional MSR algorithms running on a poor robust network, then we effectively utilized mobile agents to improve the MSR algorithm, and made the algorithm can apply to a more general network. We designed an effective mechanism to protect the information in each agent' broadcast data, and then we analyzed the convergence of the proposed control law under f-local attack model. Simulation results showed the effectiveness of our method.

Acknowledgment. This work is supported by the cyberspace security Major Program in National Key Research and Development Plan of China under grant 2016YFB0800201, Natural Science Foundation of China under grants 61572165, 61702150 and 61803135, State Key Program of Zhejiang Province Natural Science Foundation of China under grant LZ15F020003, Key Research and Development Plan Project of Zhejiang Province under grants 2017C01062 and 2017C01065, and Zhejiang Provincial Basic Public Welfare Research Project under grant LGG18F020015.

References

1. Cheng, L., Wang, Y., Ren, W., Hou, Z.G., Tan, M.: On convergence rate of leader-following consensus of linear multi-agent systems with communication noises. IEEE Trans. Autom. Control. **61**(11), 3586–3592 (2016)
2. Cheng, L., Wang, Y., Ren, W., Hou, Z.G., Tan, M.: Containment control of multi-agent systems with dynamic leaders based on a PI^n-type approach. IEEE Trans. Cybern. **46**(12), 3004–3017 (2016) AQ2
3. Zheng, Y., Ma, J., Wang, L.: Consensus of hybrid multi-agent systems. IEEE Trans. Neural Netw. Learn. Syst. **29**(4), 1359–1365 (2018)
4. Zhu, Y., Li, S., Ma, J., Zheng, Y.: Bipartite consensus in networks of agents with antagonistic interactions and quantization. IEEE Trans. Circ. Syst. II Express Briefs (2018). https://doi.org/10.1109/TCSII.2018.2811803
5. Dolev, D., Lynch, N.A., Pinter, S.S., Stark, E.W., Weihl, W.E.: Reaching approximate agreement in the presence of faults. J. ACM (JACM) **33**(3), 499–516 (1986)
6. LeBlanc, H.J., Koutsoukos, X.D.: Consensus in networked multi-agent systems with adversaries. In: 14th International Conference on Hybrid Systems: Computation and Control, pp. 281–290. ACM (2011)
7. Kieckhafer, R.M., Azadmanesh, M.H.: Reaching approximate agreement with mixed-mode faults. IEEE Trans. Parallel Distrib. Syst. **5**(1), 53–63 (1994)
8. LeBlanc, H.J., Zhang, H., Koutsoukos, X., Sundaram, S.: Resilient asymptotic consensus in robust networks. IEEE J. Sel. Areas Commun. **31**(4), 766–781 (2013)
9. Wu, Y., He, X., Liu, S., Xie, L.: Consensus of discrete-time multi-agent systems with adversaries and time delays. Int. J. Gen. Syst. **43**(3–4), 402–411 (2014)
10. Dibaji, S.M., Ishii, H.: Resilient multi-agent consensus with asynchrony and delayed information. IFAC-Pap. OnLine **48**(22), 28–33 (2015)
11. Wu, Y., He, X.: Secure consensus control for multi-agent systems with attacks and communication delays. IEEE/CAA J. Autom. Sin. **4**(1), 136–142 (2017)

12. Zhao, C., He, J., Chen, J.: Resilient consensus with mobile detectors against malicious attacks. IEEE Trans. Signal Inf. Process. Netw. **4**(1), 60–69 (2018)
13. Mi, S., Han, H., Chen, C., Yan, J., Guan, X.: A secure scheme for distributed consensus estimation against data falsification in heterogeneous wireless sensor networks. Sensors **16**(2), 252 (2016)
14. Kieckhafer, R., Azadmanesh, M.: Low cost approximate agreement in partially connected networks. J. Comput. Inf. **3**(1), 53–85 (1993)
15. Vaidya, N.H., Tseng, L., Liang, G.: Iterative approximate byzantine consensus in arbitrary directed graphs. In: 2012 ACM Symposium on Principles of Distributed Computing, pp. 365–374. ACM (2012)
16. Zhang, H., Sundaram, S.: Robustness of information diffusion algorithms to locally bounded adversaries. In: 2012 American Control Conference (ACC 2012), pp. 5855–5861. IEEE (2012)
17. Zhang, H., Fata, E., Sundaram, S.: A notion of robustness in complex networks. IEEE Trans. Control. Netw. Syst. **2**(3), 310–320 (2015)
18. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. U. S. A. **101**(9), 2658–2663 (2004)
19. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)
20. Pons, P., Latapy, M.: Computing communities in large networks using random walks. J. Graph Algorithms Appl. **10**(2), 191–218 (2006)
21. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
22. Arenas, A., Duch, J., Fernández, A., Gómez, S.: Size reduction of complex networks preserving modularity. New J. Phys. **9**(6), 176 (2007)
23. Ma, C.Y., Yau, D.K., Chin, J.c., Rao, N.S., Shankar, M.: Matching and fairness in threat-based mobile sensor coverage. IEEE Trans. Mob. Comput. **8**(12), 1649–1662 (2009)
24. Duan, X., He, J., Cheng, P., Chen, J.: Exploiting a mobile node for fast discrete time average consensus. IEEE Trans. Control. Syst. Technol. **24**(6), 1993–2001 (2016)

Author Queries

Chapter 26

Query Refs.	Details Required	Author's response
AQ1	Please check and confirm the term "$\mathcal{G} = (\mathcal{V}, \mathcal{E},)$" has been changed as "$\mathcal{G} = (\mathcal{V}, \mathcal{E})$" in Defintion 3.1 under Sect. 3.	
AQ2	Please check and confirm the edit made in Ref. [2].	
AQ3	Reference [13] is given in the list but not cited in the text.	