Reliability Engineering and System Safety 235 (2023) 109252

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

An adaptive attack model to network controllability

Sheng Li, Wenwen Liu, Ruizi Wu, Junli Li *

School of Computer Science, Sichuan Normal University, Chengdu, 610101, China

ARTICLE INFO

Keywords:
Network reliability assessment
Adaptive attack
Network controllability
Controllability robustness
Thompson sampling

ABSTRACT

For the ultimate goal of protecting the network controllability and enhancing the controllability robustness, one can learn from how a network can be effectively destructed. In this paper, based on Thompson Sampling, we propose a Probabilistic Adaptive Attack model (PAA) that aims at destructing the network controllability. A set of multiple established attack strategies are employed in PAA, and a probabilistic model is used to adaptively choose the most destructive attack strategy for the current attack. The probabilistic model is updated based on the feedback of the controllability destruction of each single attack strategy. After applying PAA on both synthetic and real-world networks, extensive simulation results not only verify that PAA is more destructive than the single attack strategies, but also show that the probabilistic model can effectively choose the *true* best-performing attack strategy. Its accuracy reaches 81.96% in synthetic networks and 94.14% in real-world networks. The presented study aims to propose a highly efficient attack scheme and encourage the research of advanced methods to build robust networked systems.

1. Introduction

In the past decades, complex networks have received widespread attention in the research and development. As powerful tools for analyzing complex systems, complex networks have penetrated into all walks of life, such as Internet, power grids [1], logistics [2],transportation [3], protein interaction [4], ecosystems [5], language evolution [6], and so on.

To effectively and efficiently control the networked systems is the ultimate goal of studying complex networks. Network controllability represents the ability that a networked system can be steered from any initial state to any targeted state, within a finite time duration, via a set of proper input signals [7]. The purpose of "pinning control" strategy [8] is to address the fundamental questions of how many and which nodes should be controlled to achieve effective control. It aims to find the method that could change the state or the structure of the networked system via only a few changes of control input signals. Some phenomena in biology reflect this key idea of the "pinning control". For example, Caenorhabditis elegan is a worm with around 300 neurons and 5000 synapses, while controlling 49 neurons (about 17% of the neurons) is enough to totally change its locomotion patterns and posterior body movements [9].

A network may significantly change its functions and basic properties when some parts are attacked. In the real-world, random failures and malicious attacks may severely change the controllability of networks. For example, serious problems such as cascading failures [10, 11] happen when power grids are damaged by connection failures or malicious attacks. The original functions and controllability are severely degenerated under attacks. Therefore, it is necessary for a networked system to have a strong robustness against attacks, such that it can maintain controllability under some attacks, meanwhile gain more time to restore or repair the system. The ability of a complex network to maintain its controllability under various failures or attacks is called the *controllability robustness*. Controllability robustness implies that external control inputs have minimal effect on the system during an attack. A controllability-robust network is able to maintain its stability even when under attack.

For summary, Lou et al. [12] reviewed the recent research progress about controllability robustness of complex networks.

In this paper, we investigate the controllability robustness of the directed networks from the perspective of effective node attacks, which aims at degenerating the system controllability as much as possible by a sequence of node-removals. Instead of proposing more features or attributes to measure the importance of nodes, a Probabilistic Adaptive Attack model (PAA) is designed to choose the best attack strategy among a set of established strategies iteration by iteration. In each iteration, PAA take samples from probability distribution of different attack strategies. By evaluating the samples, PAA recommends the most suitable attack strategy. The model is updated based on the feedback of strategy effectiveness. In this way, the attack model adaptively selects the most proper attack strategies. The process may help us to better

* Corresponding author.
E-mail address: lijunli@sicnu.edu.cn (J. Li).

https://doi.org/10.1016/j.ress.2023.109252
Received 19 July 2022; Received in revised form 24 February 2023; Accepted 14 March 2023
Available online 16 March 2023

understand the node vulnerability, and thus those important nodes can be well protected.

The rest of the paper is organized as follows. Section 2 summarizes the related work of network attack and robustness. Section 3 reviews the network controllability and its robustness against various destructive attacks. Section 4 introduces the details of PAA. Extensive experimental study is carried out in Section 5. Finally, Section 6 concludes the investigation.

2. Related work

Many attack strategies against the network connectivity have been proposed. Intuitively, removing the most influential node is likely to damage the structure of the network significantly. Assuming that there is a feature or an attribute Q to represent the characteristic of a node or the overall network. For example, when Q represents degree of a node, removing the node with the maximum-Q means attacking the influential central nodes with most connections. In previous works, the feature or attribute Q can be: (1) local information such as out-degrees [13]; K-shell (or K-core) [14], which represents the maximal sub-graph with minimum degree greater than or equal to K; (2) global attribute such as betweenness [13,15] and closeness [16,17]; (3) Eigenvector centrality such as PageRank [18,19], which is originally used to measure the importance of web pages; (4) Region centrality; Region centrality is the sum of centrality measures [20]. It takes into account the topological characteristics and geographical structure of the spatial network; (5) Other destructive indicators such as collective influence [21], articulation [22], decycling and tree destruction [23], branch weight [24], etc. Wandelt et al. [25] extracted knowledge from random failures in the network leading to extremely effective attacks, which also avoid calculating the relevant characteristics of each node.

The number of weakly connected components in a directed network can be used to evaluate the effect of a network attack on the connectivity of the network. If there are multiple connectivity components in the network, which means that some nodes will not be able to connect to other nodes. This type of network has poor connectivity.

For network controllability, a well-controlled network requires good connectivity as a condition, and if the network is poorly connected, each connected component forms a sub-network. Each sub-network requires at least one driver node for control. As shown in Fig. 1(a), the network has three connected components, and each connected component needs a driver node for control. However, a well-connected network is not always well-controlled. As shown in Fig. 1(b) and (c). Structures like chain-like or star-like network are well-connected, but they have completely different controllability. One driver node can drive the chain-like network, but star-like network needs a lot.

Although connectivity has relationship with controllability, they are not completely correlated. Good connectivity cannot guarantee good controllability, but good controllability implies good connectivity [26]. Therefore, attack strategies aiming at destructing the network controllability are different from those aiming at destructing the network connectivity.

Liu et al. [27] proposed random upstream or downstream strategies. The upstream/downstream nodes are defined according to the hierarchical structure of the directed networks, where the upstream or downstream nodes are attacked randomly, one by one, resulting in a more destructive strategy than the pure random attacks. Thomas et al. [28] analyzed the tolerance of network controllability to edge attack and found that betweenness edge attacks significantly outperform other targeted edge attacks. Wang et al. [29] studied the impact of bridges on network controllability. A bridge is a kind of specific edge whose deletion disconnects the network. When the average degree of the network is low, the bridge removal strategy is more effective than betweenness edge attacks. Song et al. [30] investigated how the centrality of driver nodes affects the controllability of complex networks. Liu et al. [7] categorized the edges as critical, ordinary, and redundant, according to the edge's contributions to the controllability. Sun et al. [31] designed an attack strategy based on the critical edges where all the critical edges in the initial topology are collected and attacked. After all the initial critical edges are removed, random attacks are then performed. Lou et al. [32] further extended the critical edges criterion to nodes. A hierarchical attack framework was proposed to attack a network according to the priority of the categories of nodes. It can effectively destroy the network controllability. Chen et al. [33] investigated the impact of different attack methods on the controllability robustness of different structured complex networks and found that multi-ring structure is a key element to controllability.

Meanwhile, the development of deep learning also provides new research technologies in this field. Lou et al. [34] introduced an approach to convert the structure of the complex network to images. Then they predict the controllability robustness via a deep learning model. Meng et al. [35] proposed the SIRV-NI-EG model for better vaccination strategies, which identified key nodes based on node importance and centrality ranking to control the spread of infectious diseases. Fan et al. [36] describes failure propagation models by motifs and propose defense strategies to against degree-based targeted attacks.

3. Network controllability and controllability robustness

The state of an LTI (linear and time invariant) system can be described by the following ordinary differential equation,

$$\frac{d\mathbf{x}(t)}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{1}$$

where A represents the adjacency matrix of the networked system; The dimension of the matrix A is $N \times N$. N is the number of nodes in the network. B represents the external input structure; The dimension of the matrix B is $N \times M$. M is the number of external driver signals. $\mathbf{x}(t)$ represents the system state at time t; $\mathbf{u}(t)$ represents the input control signals at time t.

According to the Kalman rank condition [37], the necessary and sufficient condition for a network to be controllable is that the controllability matrix has a full row rank, namely, $rank(C) = N$, where $C = [B\ AB\ A^2B \cdots A^{N-1}B]$ is the controllability matrix;

N_D is the minimum number of driver nodes to retain the controllability of the network. There are two practical ways to measure N_D, namely the minimum inputs theorem (structural controllability) [7] and exact controllability [38].

For structural controllability, N_D can be calculated as follows:

$$N_D = \max\left\{1, N - |E^*|\right\}, \tag{2}$$

where $|E^*|$ is the size of the maximum matching E^* and N is the number of nodes in the network. Exact controllability extends its applicability to the undirected and weighted networks, calculated as follows:

$$N_D = \max\{1, N - rank(A)\}. \tag{3}$$

Network controllability is normalized as the density of the needed driver nodes, denoted by n_D and calculated as follows:

$$n_D = \frac{N_D}{N}. \tag{4}$$

Given a series of successive attacks, the controllability curve records the controllability change during the attacks. It can be described by Eq. (5).

$$n_D(i) = \frac{N_D(i)}{N - i}, \quad i = 0, 1, \ldots, N-1, \tag{5}$$

where $N_D(i)$ is the number of needed driver nodes to retain the controllability after i nodes have been removed; N represents the number of nodes in the original network.

There are different measures to measure the overall network controllability robustness, for example, the unique robustness measure [39,40], the robust control centrality [41], and the ordinal ranks [42].

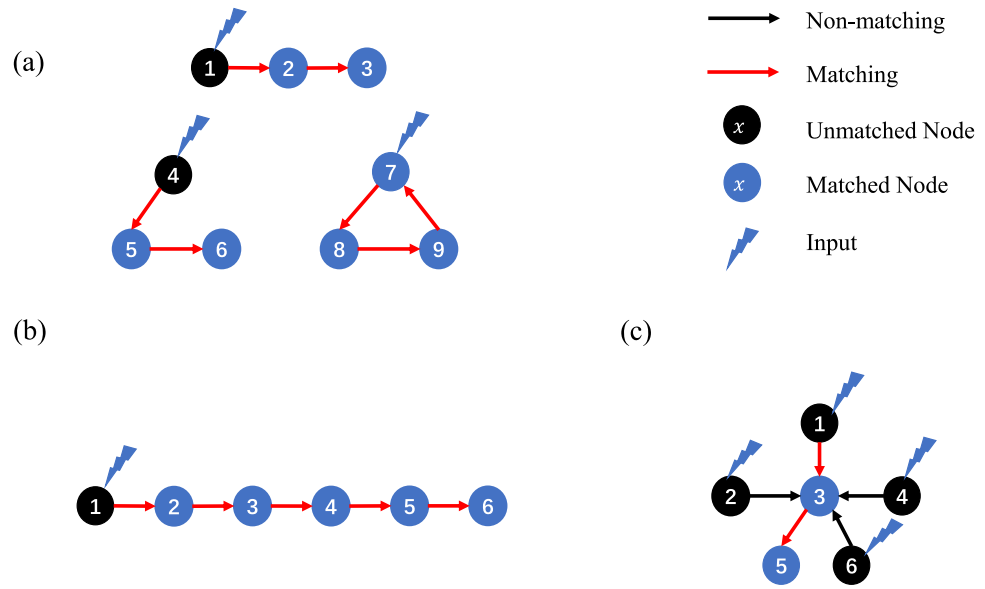

Fig. 1. Network controllability with different connectivity. According to the minimum input theorem, the driver node can be determined by matching link in the network. (a) A network with three connected components. It requires 3 control nodes to input control signals. (b) Chain-like network. It requires 1 control node to input the control signal and the network contains only one connected component. (c) Star-like network. It requires 4 control nodes to input control signals and the network contains only one connected component.

In this paper, we use the unique robustness measure [39,40], denoted by R-index and calculated as follows:

$$R = \frac{1}{N-1}\sum_{i=1}^{N-1} n_D(i). \tag{6}$$

R-index averages the network controllability during the attack process. The range of R-index is $R \in (0, 1]$. The smaller value of R indicates that the system has more superior controlability robustness, and vice versa. Given a network $\mathcal{A}$, we set π^1 and π^2 as two different attack strategies. R_1 and R_2 are the corresponding R-index values of using π^1 and π^2 attack to destruct the controllability of $\mathcal{A}$. If $R_1 > R_2$, then we determine π^1 is a more efficient strategy to destruct the network $\mathcal{A}$.

4. Probabilistic Adaptive Attack (PAA)

Thompson Sampling (TS) is a heuristic method for solving the exploration–exploitation dilemmas. It is an algorithm for decision-making that aims to extrapolate future performance from historical performance and make decisions. TS has been applied in several areas such as online advertising [43], cognitive radio [44], portfolio optimization [45] and control of unknown linear systems [46]. Supposing that we have k strategies to select optimal one strategy, the basic idea is to assume prior distributions of expected reward of different strategies. TS takes one sample from each prior distribution, then carries out the strategy with the largest sample value. TS collects the feedback of the chosen action and shifts the posterior distribution shape according to the feedback. Usually, beta distribution is set as prior distribution, because it is the conjugate prior for the Bernoulli feedback in Bayesian inference.

To determine the shape of the beta distribution, two parameters α, β should be estimated. Both feedback and prior knowledge affect α, β. The feedback is the reward of the last decision, and the prior knowledge is the accumulated feedback of the historical decision.

In TS, the feedback distribution given by environment is independent and identically distributed. The feedback affects the shape parameters of the beta distribution. Positive feedback affects α, while negative feedback affects the parameter, both of them make the shape parameters increase monotonically [47]. According to the formula of the beta distribution variance Eq. (7), the variance becomes smaller

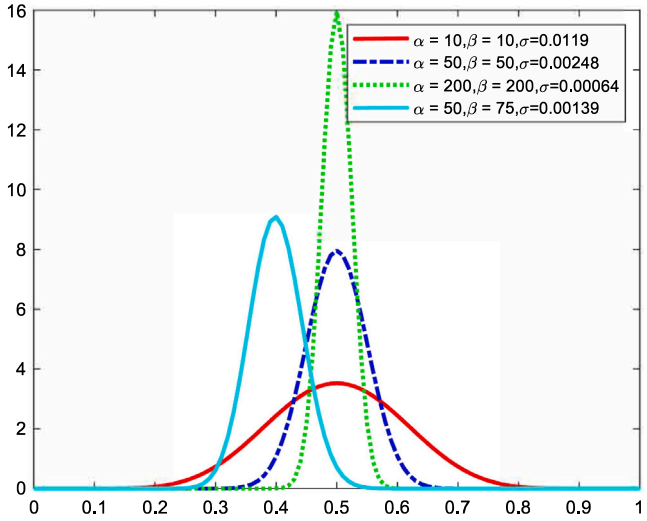

Fig. 2. An example of beta distribution probability density functions with different shape parameter α and β. The horizontal axis x represents the range of values of sample x. With the increase of α or β values, the variance σ of the distribution decreases and the distribution of sampling results become concentrated.

with the increment of the value of α, β. The visualization is explained in Fig. 2. With the accumulation of feedback from each decision, the shape parameters α and β also increases. Thus, the range of sampling results becomes concentrated. TS subjectively judges strategies with lower expectations to be ineffective strategies and rarely explores those strategies. Thus, the range of selection tends to be fixed and cannot be switched dynamically. As a result, the algorithm easily falls into a local optimum.

$$Var(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{7}$$

When taking the directed network as an object to be controlled, the *attack strategy selector* and *network* form a classical control loop.

The *attack strategy selector* is considered as the agent and the *network* is the environment. In the classic closed-loop control, the actuator/intelligent/agent makes decisions based on changes in the environment. The input of the network is the selected strategy from attack strategy selector, while the output is the network topology after the attack. Since the topology of the network is also the input for attack selection, they form a loop to affect each other. In this loop, the state and topology of the network are non-stationary due to the fact that the attack removes the node in every iteration. In this case, we cannot assume the feedback distribution is independent and identically distributed. Therefore, the traditional TS is not applicable to dynamic problems.

The traditional TS is suitable for stationary environment while the network dynamically changes after being attacked. An effective method to tackle this problem is to appropriately reduce the value of shape parameters or slow down the growth rate of shape parameters. In this way, compared with traditional TS, It makes the variance of the distribution larger, which is more conducive to exploration in a dynamic environment.

Network attack forms a non-stationary environment. There is no prior knowledge about the strategy in the exploration stage of the attack, the first priority is to explore different strategies as much as possible. Large distribution variance can make the sampling results more diverse and random, which is very helpful for exploration. As the attack process proceeds and prior knowledge (historical performance) accumulates, the shape parameter of the beta distribution keeps increasing, resulting in the gradual decrease in the variance of the distribution. At this stage, exploitation of the attack strategy is more important than exploring other strategies. The small variance helps to identify the good/poor performing strategies and avoid over-exploration of ineffective strategies. However, if we get stuck in a locally optimal strategy, the smaller the variance is, the more difficult it is to choose the optimal strategy. In order to adapt to the dynamic environment and change the strategy in time, the variance should not be too small compared with traditional Thompson sampling. Thus, the appropriate variance is particularly important in this problem.

Therefore, PAA sets Decay Sliding Window(DSW) to control shape parameter and variance based on the TS. DSW gives monotonically increasing weights to each shape parameter in the window and gives smaller weights to parameter out the window to achieve the purpose of regulating shape parameters indirectly and increase the variance of the distribution. This makes it possible to obtain more diverse samples, so as to explore appropriate attack strategies in a dynamic environment.

4.1. Probabilistic model

In a network attack, we need to completely destroy a network to study its controllability robustness. Given a set of established attack strategies $\mathcal{K} = \{\pi^1, \ldots, \pi^K\}$. The set $\mathcal{K}$ contains K strategies. Based on the probabilistic model, at each time step $t \in \{1, \ldots, T\}$, a strategy $\pi_t \in \mathcal{K}$ is selected to attack the network. If there are N nodes in a network, $N-1$ node attacks on the network can completely decompose the network. We need $N-1$ actions to execute in $N-1$ time steps. So $T = N-1$.

Each attack strategy k in the strategy set establishes a beta distribution $Beta(\alpha^k(t), \beta^k(t))$ as expected feedback distribution. When conducting a network attack at time step t, we need to collect a sample w_t^k from each strategy distribution, the attack strategy to be implemented π_t is estimated by the largest sampling result as the following Eq. (8).

$$\pi_t = \underset{k\in\{1,2,\ldots,K\}}{\arg\max}\ w_t^k, \tag{8}$$

where w_t^k is the sampling result, k is the attack strategy index, t is the current time step. The shape parameters α^k, β^k of the beta distribution control the range of sample values w_t^k. After π_t is applied to the network, $\alpha^k(t), \beta^k(t)$ are updated according to the network controllability feedback $r^{\pi}(t)$.

The shape parameters grow monotonically during the shape parameter update. If the growth of parameters is not limited, the algorithm is easily falling into the local optimum. For PAA, there are two ways to ameliorate oversized shape parameters $\alpha^k(t), \beta^k(t)$ after multiple selections, namely DSW and parameter decay [48]. The DSW assigns different weights to the passed parameters to slow down the growth of parameters. Parameter decay is used to correct the poor performance when the strategy frequently chooses the strategy with negative feedback. The DSW acts like a factory, it processes the parameters that pass through it and wraps the parameters that leave it. The DSW consists of L grids, each grid corresponds to a time step. PAA places L grids one by one according to the time step from far to nearest, and the corresponding weights are incremented in order.

The weight of DSW is defined as Eq. (9),

$$D(i) = \ln(e^{\epsilon} + \frac{i}{L}(e - e^{\epsilon})), \tag{9}$$

$D(i)$ is the weight given to the parameter corresponding to the ith grid in the DSW. L is the length of the window, and ϵ is the minimum reserved proportion of parameter, which means that after the parameter passes through DSW, parameters are modified to its original $D(0)$ times. The range of D is $[\epsilon, 1]$. The range of i is $[0, L]$.

The derivative of $D(i)$ is calculated by Eq. (10). From the derivative of DSW, we can conclude that the larger i is, the slower the weight grows. Therefore, the larger weights are all concentrated in larger i positions

$$D'(i) = \frac{1}{(\frac{e^{\epsilon}}{e-e^{\epsilon}}) \cdot L + i}. \tag{10}$$

The window always slides to the position of the nearest time step. The relationship between the position of the grid in the window and the time step can be described as $i \propto t - L + i$, i is the position of the window, $t - L + i$ is the time step corresponding to the window. The larger the window position, the nearer the corresponding time step. The most recently updated parameters are at the end of window. Those parameters are given large weights. If the parameter is farther away from the end of the window, it is given less weight. In this way, the recent performance has the most impact in this model.

In PAA, shape parameters are defined as

$$\begin{aligned} \alpha^k(t) &= a_h^k(t) + a_c^k(t) \\ \beta^k(t) &= b_h^k(t) + b_c^k(t) \end{aligned}, \tag{11}$$

where $\alpha^k(t), \beta^k(t)$ represent the α and β shape parameter of strategy k when the time step is t. a, b are the adjustment parameter of α and β value, which are used to indicate the feedback of the selected attack.

$$a_h^k(t) = \begin{cases} \sum_{i=1}^{t-L} a^k(i) & , t \geqslant L+1 \\ 0 & , t \leqslant L \end{cases}, \tag{12}$$

$$a_c^k(t) = \sum_{i=t-L+1}^{t} a^k(i) \cdot D(i+L-t). \tag{13}$$

The subscript h means the parameter $a_h^k(t), b_h^k(t)$ of strategy k have been processed by DSW or are located outside the window. The subscript c means the parameter $a_c^k(t), b_c^k(t)$ of strategy k still in the DSW after t attacks. $a_h^k(t), b_h^k(t)$ counts the strategy historical performance after t attacks, and $a_c^k(t), b_c^k(t)$ counts the strategy performance in the DSW. The parameter $b_h^k(t), b_c^k(t)$ are defined in the same way.

It should be noted that after the parameter leaves the window, $a^k(t-L)$ is set as $D(0) \cdot a^k(t-L)$, $b^k(t-L)$ is set as $D(0) \cdot b^k(t-L)$. This means that when the shape parameter passes through DSW, its value is permanently reduced to the original ϵ times($D(0) = \epsilon$). This prevents excessive historical feedback from causing shape parameters to become too large. This process can be expressed in Eq. (14).

$$\begin{cases} a^k(t-L) = a^k(t-L) \cdot D(0) & , t \geqslant L+1 \\ b^k(t-L) = b^k(t-L) \cdot D(0) & , t \geqslant L+1 \end{cases} \tag{14}$$

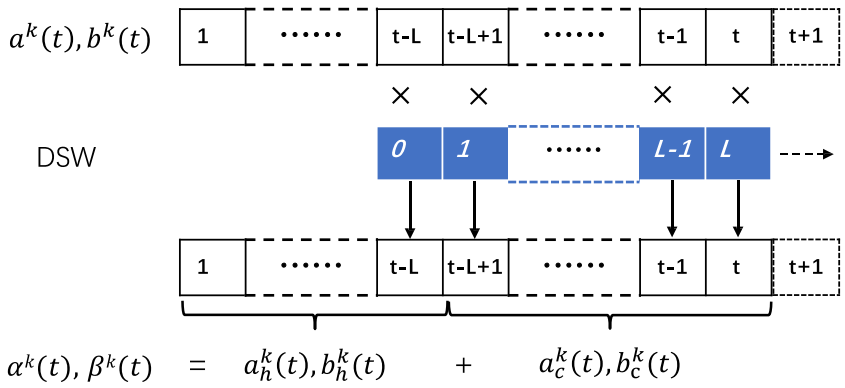

Fig. 3. Flow chart of the shape parameters calculation process. The corresponding time steps are marked in the black grid, when the parameters are located between 1 and $t-L$, a part of the parameters a, b have been processed by DSW, and the values of the parameters are reduced to ϵ times of the original. This part of the parameter is composed of a_h, b_h. Parameters located between the $t-L+1$ and t intervals, which are in the DSW, and this part of the parameters will be multiplied by the weights given by the window grid. These results make up a_c, b_c. α and β are composed of a_h, a_c, b_h, b_c both. At the next time step, with new feedback received, the DSW will move one step forward.

After attacking the network, the variation of the number of driver nodes is regarded as the network feedback $r^{\pi}(t)$ and values for shape parameter adjustment $a^{\pi_t}(t), b^{\pi_t}(t)$ can be obtained from the feedback. Feedback $r^{\pi}(t)$ definition is as shown in Eq. (15).

$$r^{\pi}(t) = N_D(\mathcal{A}_t) - N_D(\mathcal{A}_{t-1}), \tag{15}$$

where $N_D(\mathcal{A}_t)$ represents the number of driver nodes of the network $\mathcal{A}$ at time step t.

We have established criteria to distinguish between valid and invalid attacks. If the removal of a node increases the number of driver nodes that are necessary to maintain network controllability, it is considered a valid attack. Otherwise, it is considered an invalid attack. The shape parameters α and β indicate different characteristics of the attack. The effectiveness of an attack is reflected by the α value, and the ineffectiveness of an attack is reflected by the β value.

For $r^{\pi}(t)$, there are three situations in the process of network attack.

1. $r^{\pi}(t) > 0$ indicates that the network controllability decreases and the number of driver nodes increases, in this case, we regard this attack as an effective attack, and set the parameter $a_t^{\pi_t} = S, b_t^{\pi_t} = 0$.

2. $r^{\pi}(t) = 0$ indicates that the number of driver nodes remains unchanged. It is unable to distinguish whether the strategy is effective or invalid. However, PAA is optimistic about this situation. Although the attack does not cause an increase in the number of driver nodes, the attack also helps to disrupt the network structure and expose potential driver nodes. Therefore, the parameter is set as $a_t^{\pi_t} = S, b_t^{\pi_t} = F$.

3. $r^{\pi}(t) < 0$ indicates that the network controllability increases and the number of driver nodes decreases. We regard this attack strategy as an invalid attack in the current state. Therefore, the parameter is set as $a_t^{\pi_t} = 0, b_t^{\pi_t} = F$.

S represents the value of the shape parameter adjustment after a successful attack. F stand for the shape parameter adjustment value after a failed attack. This ensures that effective attack methods have large sample values. When the number of driver nodes is equal to the number of network nodes, the network has been completely disconnected. At this time, any attack will lead to the reduction of driver nodes. In this case, we will not update the model anymore.

$$\begin{cases} a_t^{\pi_t} = S, b_t^{\pi_t} = 0 & , r > 0 \\ a_t^{\pi_t} = S, b_t^{\pi_t} = F & , r = 0 \\ a_t^{\pi_t} = 0, b_t^{\pi_t} = F & , r < 0 \end{cases} \tag{16}$$

For the third case, PAA employs parameter decay to prevent being trapped by invalid method, the parameter values of other strategies are reduced by a factor λ times. In this way, the expectation of the distribution is not changed, and the variance of the distribution becomes larger. We suggest that the range of λ lie between [0.7,0.9] to ensure they have a better ability to explore and exploit.

$$\begin{matrix} a(i) = \lambda \cdot a(i) \\ b(i) = \lambda \cdot b(i) \end{matrix}, i \in \{1, \ldots, t\}. \tag{17}$$

The proof is as follows, μ is the expectation of the distribution, σ^2 is the variance of the distribution.

$$\mu^k(t+1) = \frac{\alpha^k(t+1)}{\alpha^k(t+1) + \beta^k(t+1)} = \frac{\lambda \cdot \alpha^k(t)}{\lambda \cdot \alpha^k(t) + \lambda \cdot \beta^k(t)} = \mu^k(t)$$

$$\begin{aligned} \sigma_k^2(t+1) &= \frac{\alpha^k(t+1)\cdot\beta^k(t+1)}{(\alpha^k(t+1)+\beta^k(t+1))^2(\alpha^k(t+1)+\beta^k(t+1)+1)} \\ &= \frac{\alpha^k(t)\cdot\beta^k(t)}{(\alpha^k(t)+\beta^k(t))^2(\lambda\cdot\alpha^k(t)+\lambda\cdot\beta^k(t)+1)} \\ &= \frac{\mu^k(t)(1-\mu^k(t))}{\lambda\cdot\alpha^k(t)+\lambda\cdot\beta^k(t)+1} \geqslant \frac{\mu^k(t)(1-\mu^k(t))}{\alpha^k(t)+\beta^k(t)+1} = \sigma_k^2(t) \end{aligned}$$

At time step T, for the unselected strategies, PAA does not receive its feedback, so a_t, b_t are updated to 0. It can be described by Eq. (18)

$$a_t^{\pi_k} = 0, b_t^{\pi_k} = 0, \pi_k \neq \pi_t. \tag{18}$$

The pseudocodes of PAA are shown in Algorithm 1, and the flow chart of the parameter calculation process is shown in Fig. 3. For a total of $N-1$ attacks on the network, each time needs to sample K strategies from prior distributions in the strategy set. After each attack on the network, the Hopcroft–Karp algorithm is used to calculate the changes of the driver nodes in the network. The complexity of the Hopcroft–Karp algorithm is $O(\sqrt{N} * M)$. M is the number of links in the network. Therefore, the time complexity of PAA is $O(K * M * N^{\frac{3}{2}})$.

Algorithm 1 PAA (Probabilistic Adaptive Attack)

Input: complex network $\mathcal{A}$; attack strategy set $\mathcal{K} = \{\pi^1, \pi^2, \cdots, \pi^K\}$;

for $t = 1$ to T **do**
  for $k = 1$ to K **do**
    $w_t^k \sim Beta(\alpha_t^k, \beta_t^k)$
  end for
  Perform attack strategy $\pi_t = \arg\max_k w_t^k$ on $\mathcal{A}$ once.
  Get the feedback of driver node $r^{\pi_t}(t)$.
  Update $a^{\pi_t}(t), b^{\pi_t}(t)$ according to Eq. (16)
  Update $a^k(t), b^k(t)$,$k \neq \pi_t, k \in \mathcal{K}$,according to the Eq. (18).
  Moving the Decay Sliding Window to the latest time step.
  Update parameter out of the window according to the Eq. (17).
  Recalculate $\alpha^k(t), \beta^k(t)$ according to the Eq. (11), Eq. (12), Eq. (13).
end for

5. Experimental study

In the simulations, four commonly used node attack strategies are compared. They are the betweenness-based strategy [15], degree-based strategy [13], closeness-based strategy [16,17] and PageRank-based strategy [19]. Degree-based strategy is a local strategy, which focuses on removing the node with the maximum number of edges. Betweenness-based strategy is a global strategy that focus on destroying as many shortest paths as possible. Closeness-based strategy damages the node by measuring the average distance between nodes. PageRank-based strategy treats nodes as web pages to attack the node with the greatest influence. For all the strategies, the measures are recalculated iteratively.

Five representative synthetic network models are employed, including the Random-Graph (RG) networks [49], generic Scale-Free (SF) networks [50,51], q-snapback Networks (QSN) [52], Random Triangle Networks (RTN) [42,53], and Random Rectangle Networks (RRN) [42]. In addition, four real-world networks are also used in the attack simulations.

In the remainder of this section, Section 5.1 studies the scaling property of PAA by using different network sizes. Section 5.2 verifies the effectiveness of PAA by using different average degrees. In Sections 5.1 and 5.2, PAA includes three single-feature-based strategies, i.e., $\mathcal{K}$ = {betweenness-based strategy, degree-based strategy, PageRank-based strategy}. Different combinations of attack strategies are discussed in Section 5.3. The combination of hyperparameter S and F are discussed in Section 5.4. Section 5.5 compares the performance of PAA and other attack algorithms in terms of network controllability. The selection precision of PAA is studied in Section 5.6. The effectiveness of PAA is also studied on attacking four real-world networks, as shown in Section 5.7.

For the parameters in PAA, we simply set L as 0.10 times the network size to make sure the window size is sufficient for selection of the strategy, ε is set as 0.2 to control parameter out of DSW, and we set S as 3 and F as 3. λ is set as 0.9.

5.1. Simulation experiments with different numbers of nodes

In this experiment, the performance of PAA is evaluated with different settings of the number of the network nodes. The networks with various sizes, comprised of 500, 1000, and 1500 nodes, are utilized to compare the performance of attack strategies. As the composed attacking strategy selector, the strategy groups have three single attacking strategies in this experiment. They are betweenness-based strategy, degree-based strategy and PageRank-based strategy.

In order to eliminate the influence of randomness, the experimental results are averaged from 30 repeated and independent runs. Mann–Whitney U-test (with 5% significance level) is used to check the significance of differences.

The experimental results are shown in Table 1 and Figs. 4–6. The bold values in Table 1 represent the overall best-performing attack methods (also indicated by the Rank equals 1). The average ranks indicate PAA's superior performance in all tested network structures and sizes. (calculated by Eq. (6)).

At the previous stage of the attack, all the attack methods have equal chances of being chosen by PAA as they have the same shape parameters. PAA explores different methods to determine the most effective one, and the controllability curve lies between the different curves and does not immediately converge to the optimal curve.

After a few rounds of explorations, PAA adjusts the parameters based on feedback from the strategies. Even though the curve does not converge to the optimal quickly, the trend shows that the curve keeps the same upward trend as the optimal strategy.

In the later stages, the effectiveness of some strategies begins to decline such as betweenness. As can be seen from the results, PAA successfully avoids getting trapped in local optima and selects other effective attacks in time.

Table 1
The R-index values, ranks, and significance obtained by the five attack strategies, when the network size N is set as 500, 1000, and 1500, respectively. The network average degree is 5, strategies set $\mathcal{K}$ includes betweenness-based strategy, degree-based strategy, PageRank-based strategy.

Network	Method	N = 500		N = 1000		N = 1500	
		R-index	Rank	R-index	Rank	R-index	Rank
ER	Betweenness	0.45650	2*	0.45348	2*	0.45435	2*
	Degree	0.42766	3*	0.42660	3*	0.42635	3*
	PageRank	0.42693	4*	0.42300	4*	0.42498	4*
	Closeness	0.38656	5	0.38306	5	0.38318	5
	PAA	**0.48067**	**1**	**0.48394**	**1**	**0.48516**	**1**
SF	Betweenness	0.80929	5*	0.82669	5*	0.83787	5*
	Degree	0.85259	2*	0.87346	2*	0.88430	2*
	PageRank	0.84344	3*	0.86686	3	0.87823	3*
	Closeness	0.83952	4	0.86236	4	0.87476	4
	PAA	**0.86622**	**1**	**0.88781**	**1**	**0.89822**	**1**
QSN	Betweenness	0.46269	2*	0.46161	2*	0.45993	2*
	Degree	0.41410	3*	0.41227	3*	0.41365	3*
	PageRank	0.33944	4*	0.34529	4*	0.34708	4*
	Closeness	0.33197	5	0.33008	5	0.32805	5
	PAA	**0.49628**	**1**	**0.49188**	**1**	**0.49301**	**1**
RTN	Betweenness	0.49462	3*	0.49713	3*	0.49758	4*
	Degree	0.49023	4*	0.49704	4*	0.49768	3*
	PageRank	0.52002	2*	0.52340	2*	0.52319	2*
	Closeness	0.47110	5	0.47595	5	0.47516	5
	PAA	**0.54464**	**1**	**0.54955**	**1**	**0.55011**	**1**
RRN	Betweenness	0.46616	2*	0.46747	2*	0.46463	2*
	Degree	0.45826	3*	0.46013	3*	0.45968	3*
	PageRank	0.44640	4*	0.44783	4*	0.45053	4*
	Closeness	0.41148	5	0.41346	5	0.41437	5
	PAA	**0.49005**	**1**	**0.49506**	**1**	**0.49812**	**1**
AVG	Betweenness	–	2.8	–	2.8	–	3
	Degree	–	3	–	3	–	2.8
	PageRank	–	3.4	–	3.4	–	3.4
	Closeness	–	4.8	–	4.8	–	4.8
	PAA	–	**1**	–	**1**	–	**1**

*This indicates rejection of the null hypothesis at the 5% significance level.

Table 2
Time consumed for network attacks of different scales.

	N = 500	N = 1000	N = 1500
Betweenness	4.23 (s)	33.72 (s)	114.78 (s)
Degree	0.15 (s)	1.21 (s)	4.06 (s)
PageRank	0.49 (s)	2.37 (s)	6.32 (s)
Closeness	0.62 (s)	3.94 (s)	10.69 (s)
PAA	2.87 (s)	20.33 (s)	74.88 (s)

The results also show that the effectiveness of PAA is not affected by the network size scaling. In particular, in large-scale networks, PAA can play more advantages from our observation. For example, the network with 1500 nodes, 1499 feedbacks are returned from strategy evaluations in the process of removing the node. When the node removal ratio is greater than 0.6, PAA still does not fall into the local optimum and outperforms other strategies in attack effectiveness.

Table 2 shows the time consumption of PAA and compared algorithms. The time consumption of PAA includes the execution of the chosen attack algorithm and the time for evaluating the algorithm's effectiveness. On average, the selection and evaluation of PAA account for approximately 45% of the total time. In terms of running time, PAA is faster than betweenness but slower than other centrality strategies. However, PAA outperforms other attack strategies in terms of its effectiveness in R-index.

5.2. Simulation experiments with different average degrees

In this experiment, we test the strategies on networks with different average degrees. A network's average degree affects its robustness

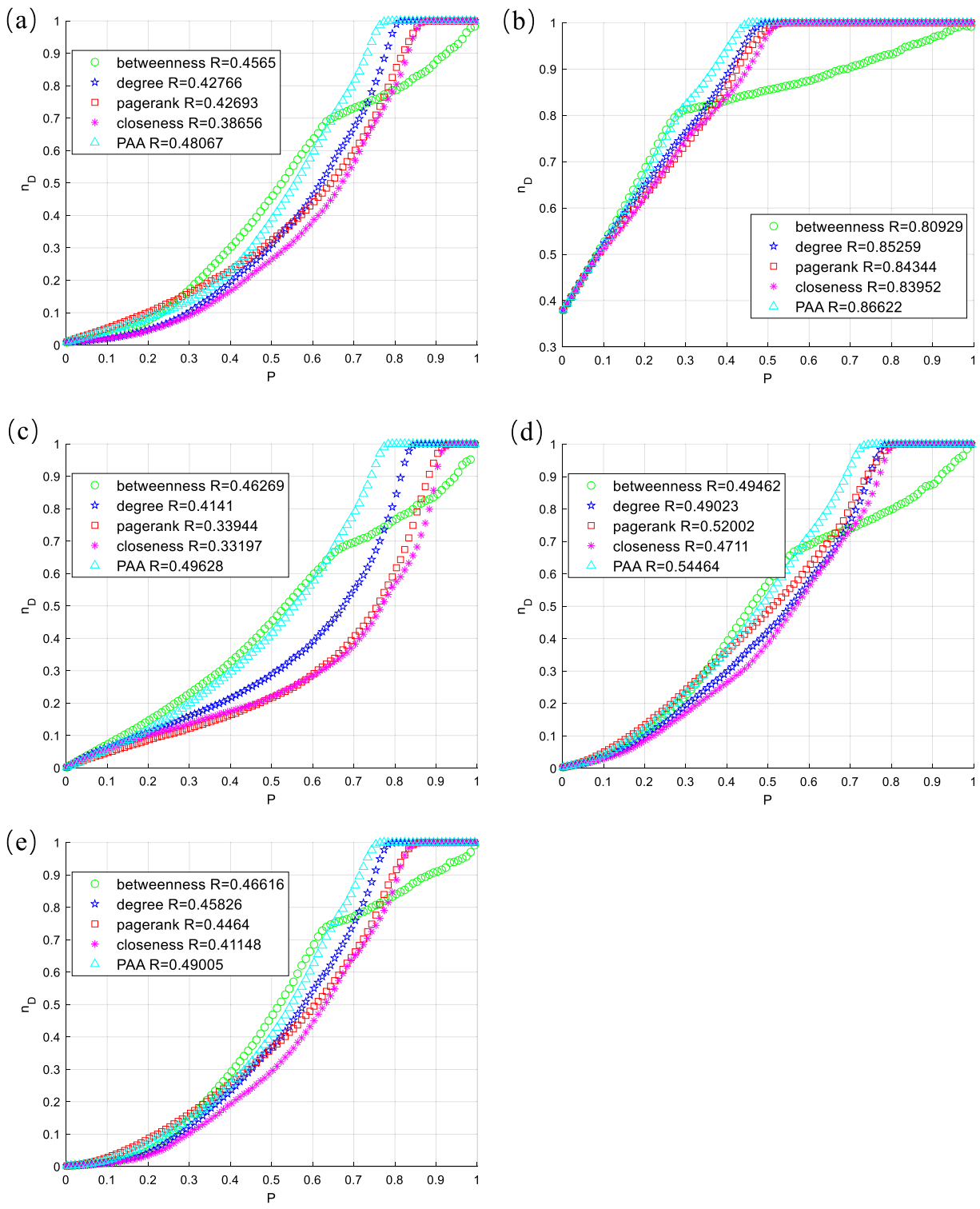

Fig. 4. The ratio of needed driver nodes n_D against the ratio of removed nodes P. The network size $N = 500$; average degrees $\langle k \rangle = 5$. (a) Random Graph. (b) Scale-Free Network (c) q-snapback Network (d) Random Triangle Network (e) Random Rectangle Network.

intuitively since each node has more redundant connections with others in the networks with high average degrees. With the increment of the average degree, the attacking effectiveness of a strategy deteriorates. We compare the networks with 1000 nodes and different average degrees $\langle k \rangle = 3$, $\langle k \rangle = 5$, $\langle k \rangle = 7$, and $\langle k \rangle = 10$, respectively. The resultant curves with $N = 1000$ and $\langle k \rangle = 5$ are shown in Fig. 5. Others are shown in Fig. 7,8,9.

Table 3 shows the R-index values, ranks, and significance obtained by the four attack strategies. The bold values represent the overall best-performing attack methods.

From our observation, for some networks with a relatively large average degree, such as the case in which the network with an average degree of 10, it is difficult to change the network structures after a few attacks. This phenomenon is more apparent when using PAA

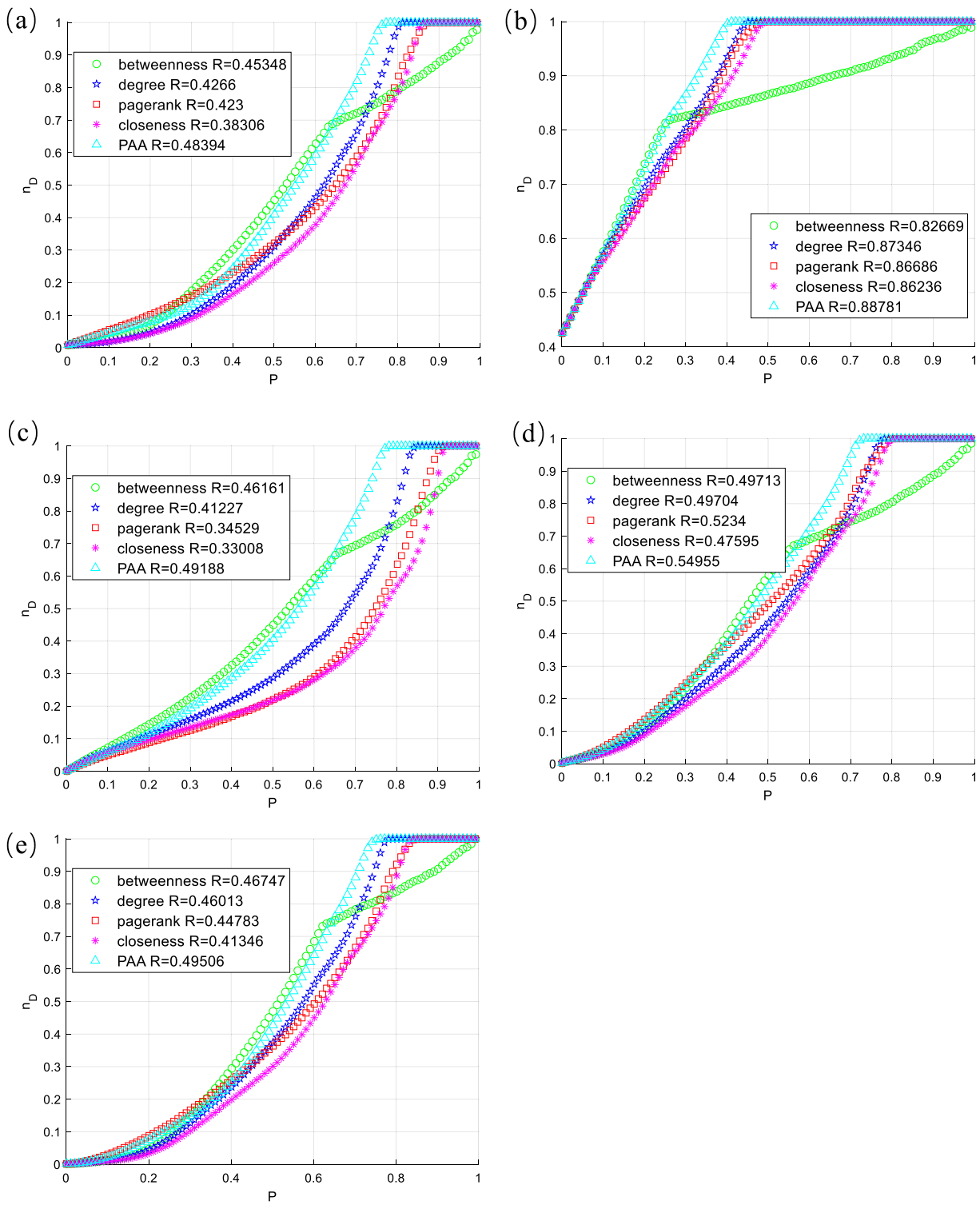

Fig. 5. The ratio of needed driver nodes n_D against the ratio of removed nodes P. The network size $N = 1000$; average degrees $\langle k \rangle = 5$. (a) Random Graph. (b) Scale-Free Network (c) q-snapback Network (d) Random Triangle Network (e) Random Rectangle Network.

because PAA estimates the selection based on the performance of historical feedback. When the state of the network does not change explicitly, PAA spends more computational time for exploration instead of adopting one particular strategy in the selector.

At the beginning of the attack, it is more difficult for PAA to select the optimal strategy from multiple strategies with similar attack results. Therefore, the strategy is stuck in a longer exploration period. However, when some nodes are removed, the differences between different attack strategies gradually become prominent. PAA can quickly adjust to exploitation mode to select the most effective attacking strategy. This experiment indicates the exploration and exploitation balance of PAA.

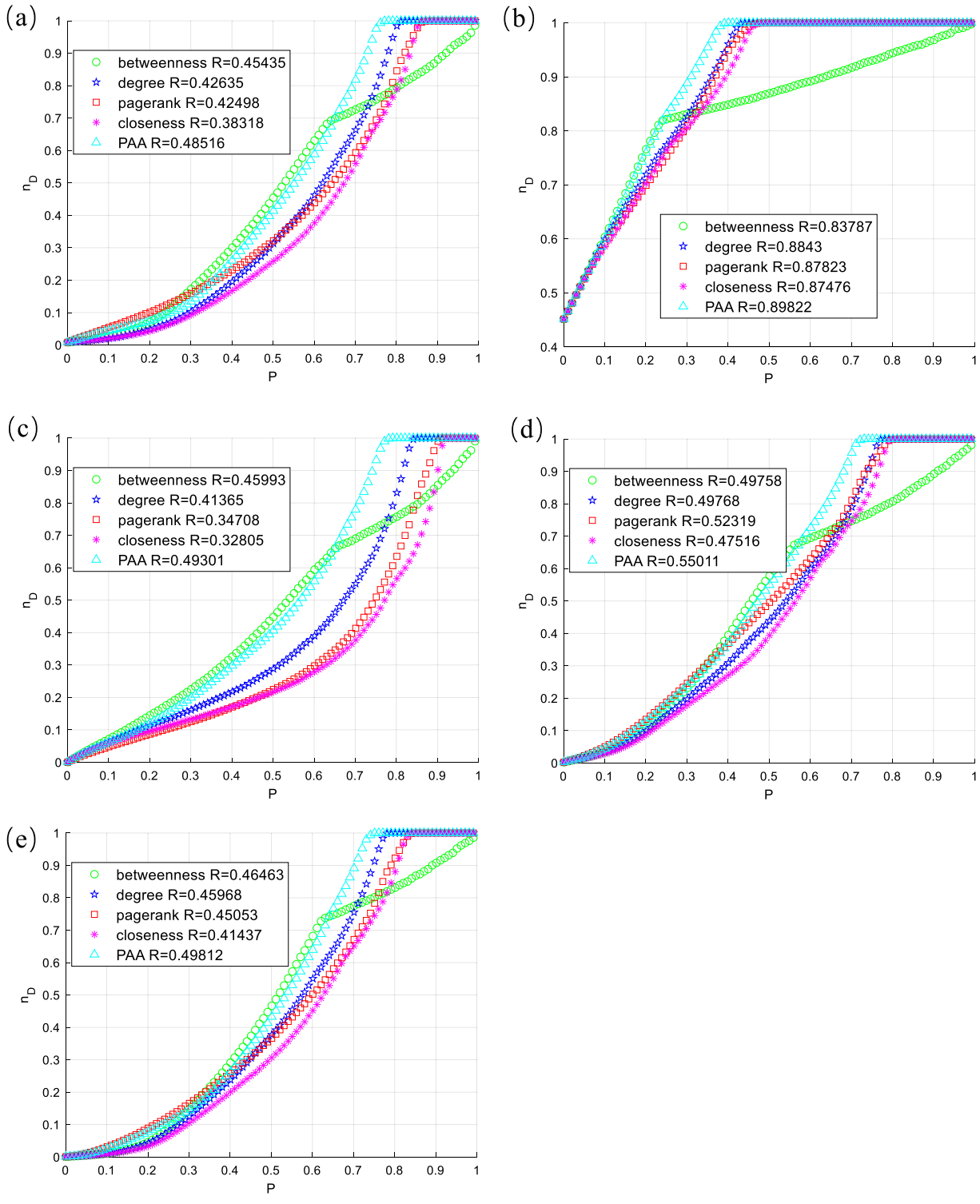

Fig. 6. The ratio of needed driver nodes n_D against the ratio of removed nodes P. The network size $N = 1500$; average degrees $\langle k \rangle = 5$. (a) Random Graph. (b) Scale-Free Network (c) q-snapback Network (d) Random Triangle Network (e) Random Rectangle Network.

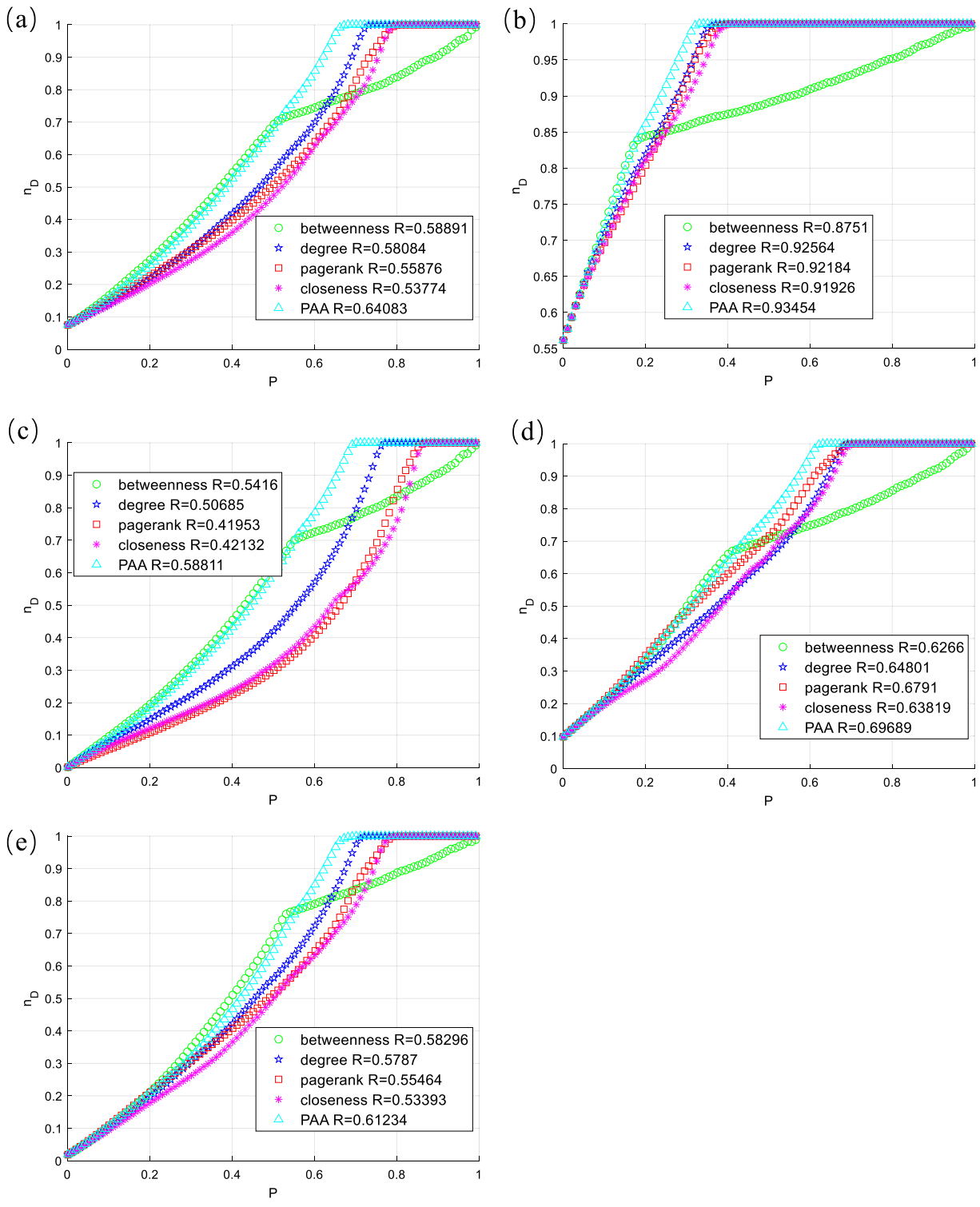

Fig. 7. The ratio of needed driver nodes n_D against the ratio of removed nodes P. The network size $N = 1000$; average degrees $\langle k \rangle = 3$. (a) Random Graph. (b) Scale-Free Network (c) q-snapback Network (d) Random Triangle Network (e) Random Rectangle Network.

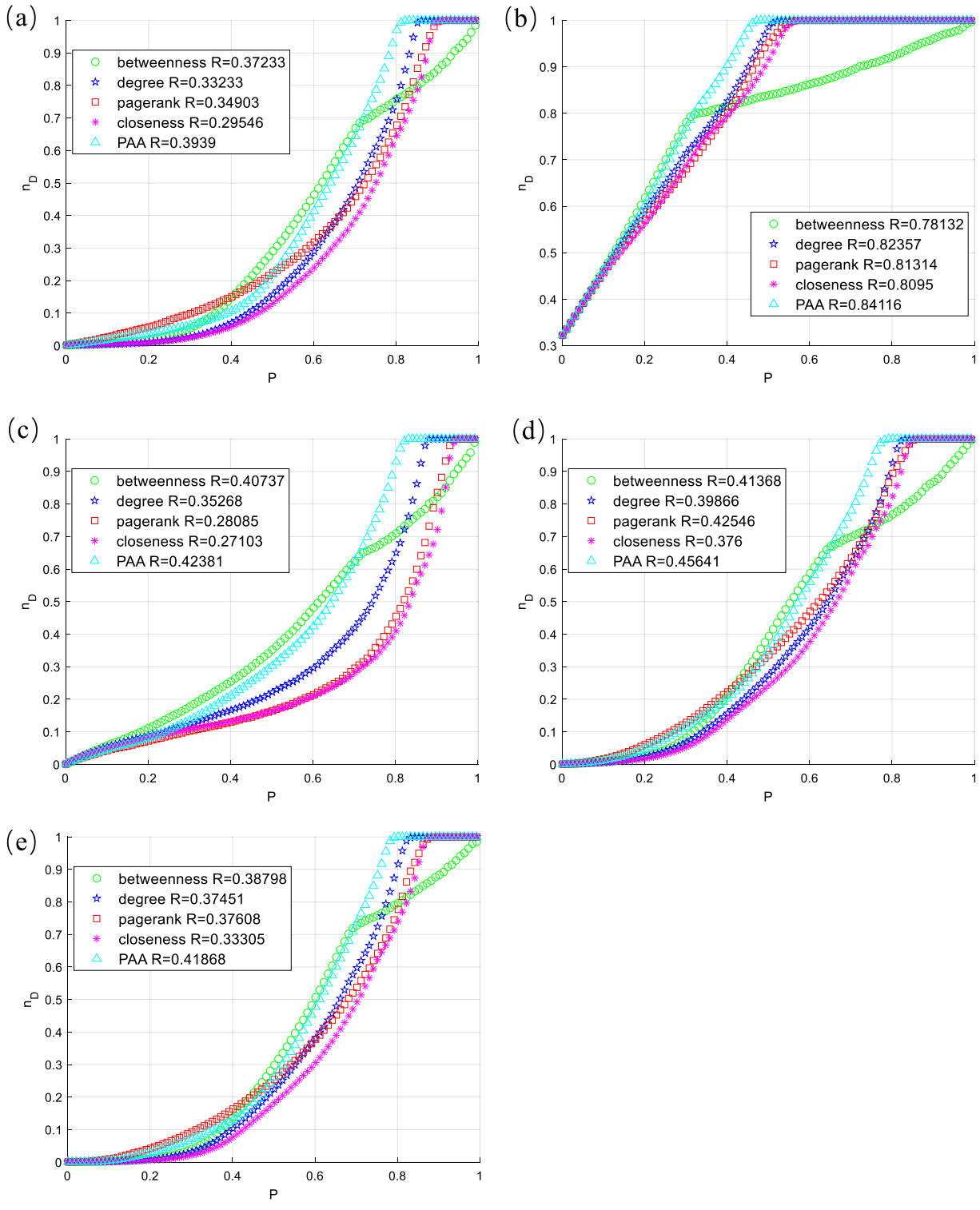

Fig. 8. The ratio of needed driver nodes n_D against the ratio of removed nodes P. The network size $N = 1000$; average degrees $\langle k \rangle = 7$. (a) Random Graph. (b) Scale-Free Network (c) q-snapback Network (d) Random Triangle Network (e) Random Rectangle Network.

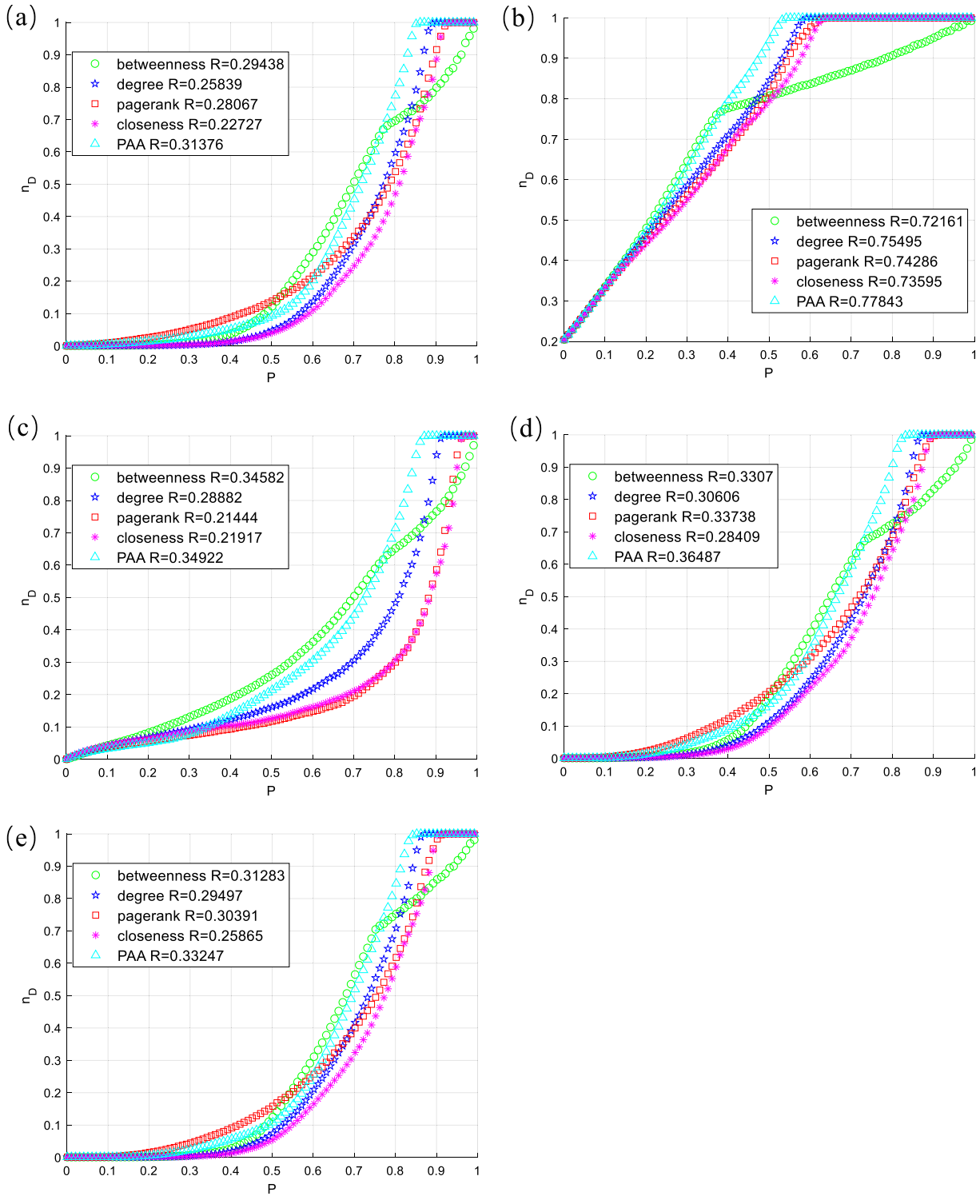

Fig. 9. The ratio of needed driver nodes n_D against the ratio of removed nodes P. The network size $N = 1000$; average degrees $\langle k \rangle = 10$. (a) Random Graph (b) Scale-Free Network (c) q-snapback Network (d) Random Triangle Network (e) Random Rectangle Network.

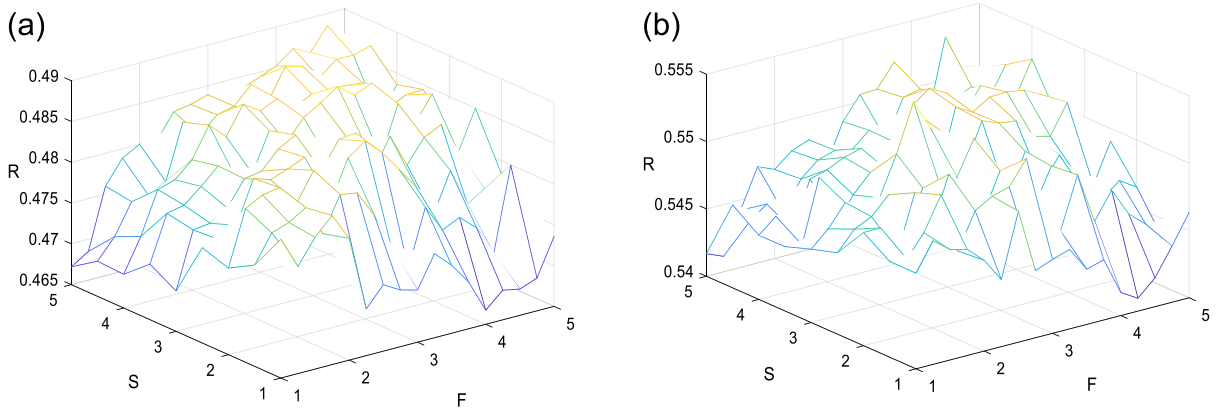

Fig. 10. The R-index under different hyperparameter S and F combination of PAA attack. The network size $N = 1000$; average degrees $\langle k \rangle = 5$. (a) Random Graph(RG). (b) Random Triangle Network(RTN).

Table 3
The R-index values, ranks, and significance obtained by the four attack strategies, when the average degree $\langle k \rangle$ is set as 3, 5, 7, and 10, respectively. The network size is 1000, strategies set $\mathcal{K}$ includes betweenness-based strategy, degree-based strategy, PageRank-based strategy.

Network	Method	$\langle k \rangle = 3$		$\langle k \rangle = 5$		$\langle k \rangle = 7$		$\langle k \rangle = 10$	
		R-index	Rank	R-index	Rank	R-index	Rank	R-index	Rank
ER	Betweenness	0.58891	2*	0.45348	2*	0.37233	2*	0.29438	2*
	Degree	0.58084	3*	0.42660	3*	0.33233	4*	0.25839	4*
	PageRank	0.55876	4*	0.42300	4*	0.34903	3*	0.28067	3*
	Closeness	0.53774	5	0.38306	5	0.29546	5	0.22727	5
	PAA	**0.64083**	**1**	**0.48394**	**1**	**0.39390**	**1**	**0.31376**	**1**
SF	Betweenness	0.87510	5*	0.82669	5*	0.78132	5*	0.72161	5*
	Degree	0.92564	2*	0.87346	2*	0.82357	2*	0.75495	2*
	PageRank	0.92184	3*	0.86686	3	0.81314	3	0.74286	3*
	Closeness	0.91926	4	0.86236	4	0.80950	4	0.73595	4
	PAA	**0.93454**	**1**	**0.88781**	**1**	**0.84116**	**1**	**0.77843**	**1**
QSN	Betweenness	0.54160	2*	0.46161	2*	0.40737	2*	0.34582	2*
	Degree	0.50685	3*	0.41227	3*	0.35268	3*	0.28882	3*
	PageRank	0.41953	5	0.34529	4*	0.28085	4*	0.21444	5*
	Closeness	0.42132	4	0.33008	5	0.27103	5	0.21917	4
	PAA	**0.58811**	**1**	**0.49188**	**1**	**0.42381**	**1**	**0.34922**	**1**
RTN	Betweenness	0.62660	5*	0.49713	3*	0.41368	3*	0.33070	3*
	Degree	0.64801	3*	0.49704	4*	0.39866	4*	0.30606	4*
	PageRank	0.67910	2*	0.52340	2*	0.42546	2*	0.33738	2*
	Closeness	0.63819	4	0.47595	5	0.37600	5	0.28409	5
	PAA	**0.69689**	**1**	**0.54955**	**1**	**0.45641**	**1**	**0.36487**	**1**
RRN	Betweenness	0.58296	2*	0.46747	2*	0.38798	2*	0.31283	2*
	Degree	0.57870	3*	0.46013	3*	0.37451	4*	0.29497	4*
	PageRank	0.55464	4*	0.44783	4*	0.37608	3*	0.30391	3*
	Closeness	0.53393	5	0.41346	5	0.33305	5	0.25865	5
	PAA	**0.61234**	**1**	**0.49506**	**1**	**0.41868**	**1**	**0.33247**	**1**
AVG	Betweenness	–	3.2	–	2.8	–	2.8	–	2.8
	Degree	–	2.8	–	3	–	3.4	–	3.4
	PageRank	–	3.6	–	3.4	–	3	–	3.2
	Closeness	–	4.4	–	4.8	–	4.8	–	4.6
	PAA	–	**1**	–	**1**	–	**1**	–	**1**

*This indicates rejection of the null hypothesis at the 5% significance level.

Overall, PAA shows to be the best performing algorithm among the comparison networks with different average degrees.

5.3. Simulation experiments with different attack combination strategies

This experiment compares the different combinations of attack strategies. Here, $\mathcal{K}$ = {betweenness-based strategy, degree-based strategy, closeness-based strategy, PageRank-based strategy}. The network size is $N = 1000$ with average degree $\langle k \rangle = 5$.

In Table 4, some of the combinations are tested. The results show that the combination of betweenness, degree and closeness(BDC) outperforms, followed by the combination of betweenness, degree(BD), the combination of betweenness, degree, closeness, PageRank(BDCP). Overall, two combinations are recommended, namely BDP and BD, as optimal strategies for PAA.

In Sections 5.1 and 5.2, it can be seen that closeness is a suboptimal strategy for network controllability, and its inclusion in the strategy set can result in poor performance by PAA. PAA may tend to select closeness during exploration, which can lead to ineffective attack strategies. However, even when poorly performing strategies are included in the set, the overall difference in R-index is less than 0.02. This further confirms that PAA is a reliable strategy selector for finding the optimal strategy and is able to resist the influence of poorly performing strategies. Overall, two combinations are recommended, namely BDP and BD.

Table 4
The R-index values and ranks of different combinations of attack strategies. The network size is 1000 and the network average degree is 5. (B stands for betweenness-based strategy; C stands for closeness-based strategy; D stands for degree-based strategy; and P stands for PageRank-based strategy.)

Network	Method	R-index	Rank
ER	BD	**0.48840**	**1**
	BDC	0.48232	3
	BDP	0.48469	2
	BDCP	0.47644	4
SF	BD	0.88509	3
	BDC	0.88452	4
	BDP	**0.88812**	**1**
	BDCP	0.88694	2
QSN	BD	0.49241	2
	BDC	0.47346	4
	BDP	**0.49519**	**1**
	BDCP	0.47941	3
RTN	BD	0.54370	3
	BDC	0.54312	4
	BDP	0.54886	2
	BDCP	**0.54933**	**1**
RRN	BD	**0.49872**	**1**
	BDC	0.49352	2
	BDP	0.49241	3
	BDCP	0.49006	4
AVG	BD	–	2
	BDC	–	3.4
	BDP	–	**1.8**
	BDCP	–	2.8

Closeness is a suboptimal strategy for network controllability, and its inclusion in the strategy set can result in poor performance by PAA. PAA may tend to select closeness during exploration, which can lead to ineffective attack strategies. However, even when poorly performing strategies are included in the set, the overall difference in R-index is less than 0.02. This demonstrates that PAA is a reliable strategy selector that can resist the influence of suboptimal strategies. Based on our results, we recommend two combinations: BDP and BD.

5.4. Simulation experiments with different parameter combinations of S and F

This experiment compares the different parameter combinations of attack strategies. Here, $\mathcal{K}$ = {betweenness-based strategy, degree-based strategy, PageRank-based strategy}. The network size is $N = 1000$ with average degree $\langle k \rangle = 5$. Both S and F are obtained by uniform sampling with an interval of 0.25 in the range of $[1, 5]$.

Fig. 10 shows the attack results, where the z-axis represents the R-index.

It can be seen from the experimental results when the value of S is close to F, its R-index is relatively high. When the value difference between S and F is large, the R-index is relatively low, which means that its attack effect is relatively poor. Therefore, employing similar values of S and F helps to enhance the effect of the attack.

5.5. Performance of different attack algorithms on network controllability

To compare the impact of four attack algorithms on network controllability, we experimented on five different networks: ER, SF, QSN, RTN, and RRN. The attack algorithms we used were Thompson sampling (TS) and Dirichlet sampling (DS), which are based on probabilistic models, as well as the Generalized network dismantling algorithm (GND) [54] and the Min-Sum algorithm [55], which break circular structures of the network.

We set the evaluation metrics for the GND attack to the controllability of the network and the number of driver nodes. The experiments were conducted in a network of 1000 nodes with an average degree of 5. The results are shown in Fig. 11.

Table 5
Accuracy rate compared to the best result.

Network	TS		DS		PAA	
	$\mathcal{L}$	Accuracy	$\mathcal{L}$	Accuracy	$\mathcal{L}$	Accuracy
ER	0.2912	70.88%	0.2689	73.11%	0.2269	77.31%
SF	0.1886	81.14%	0.1356	86.44%	0.1064	89.36%
QSN	0.2121	78.79%	0.2617	73.83%	0.1825	81.75%
RTN	0.3511	64.89%	0.2744	72.56%	0.1892	81.08%
RRN	0.2984	70.16%	0.2313	76.87%	0.1966	80.34%
AVG	0.2683	73.17%	0.2344	76.56%	0.1803	81.96%

Based on the R-index obtained from the experimental results, the attack methods can be ranked in decreasing order of effectiveness as follows: PAA, Thompson sampling(TS), Dirichlet sampling(DS), Min-Sum, and GND.

Although the circular motif has been shown to enhance network controllability and robustness [42,53], the Min-Sum algorithm, which disrupts the circular structure, may not effectively destroy network controllability during the early stages of an attack. However, it shows significant effects during the middle stages of an attack. To determine which motifs are more supportive of network controllability at different stages of network decomposition, it may be possible to combine different motif attack methods into a set of attack methods. Additionally, we found that the attack algorithm GND, even with modified evaluation metrics, was ineffective in disrupting network controllability for the attack problem.

5.6. Accuracy of choosing the best strategy

To verify whether PAA is able to recommend the *true* best-performing attack strategies, the following experiment records the entire recommendation history, and compares it to the *true* best-performing attack strategy at each time step that is collected from exhaustive search by performing all the component strategies. Networks with $N = 1000$ and $\langle k \rangle = 5$ are attacked under the BDCP. We define that at time step t, the current optimal attack method is π_t^*. During attack process, if PAA fails to select the best attack strategy, we regard that selection as a failure. We call the sum of all the failures throughout the attack as loss $\mathcal{L}$. Its definition is shown in Eqs. (19), (20).

$$\pi_t^* = \arg\max_k r_{\pi_k}(t). \tag{19}$$

$$\mathcal{L} = \frac{\sum_{t=1}^{N-1} r_{\pi_t^*}(t) - r_{\pi_t}(t)}{N-1}. \tag{20}$$

The accuracy of choosing the best strategy is calculated as $\text{accuracy} = 1 - \mathcal{L}$.

As can be seen from Table 5, the overall accuracy of PAA recommending the *true* best attack strategy is 81.96%. It is also observed that, when attacking SF networks, PAA can suggest the *true* best attack strategy with the highest probability, followed by QSN, RTN, RRN and ER. Note that given the 4 strategies in BDCP, the probability that we randomly select the optimal strategy is 0.25. PAA suggests the best strategy with a much higher accuracy than the random strategy. And the accuracy of PAA is also higher than the 76.56% of DS and 73.17% of TS.

5.7. Application on real-world networks

In this experiment, we test the performance of PAA in real-world networks. Four real-world networks are employed for the attack simulations, including the US Airlines, Protein–protein interaction network in budding yeast, crime-moreno and mouse-kasthuri-graph-v4. The

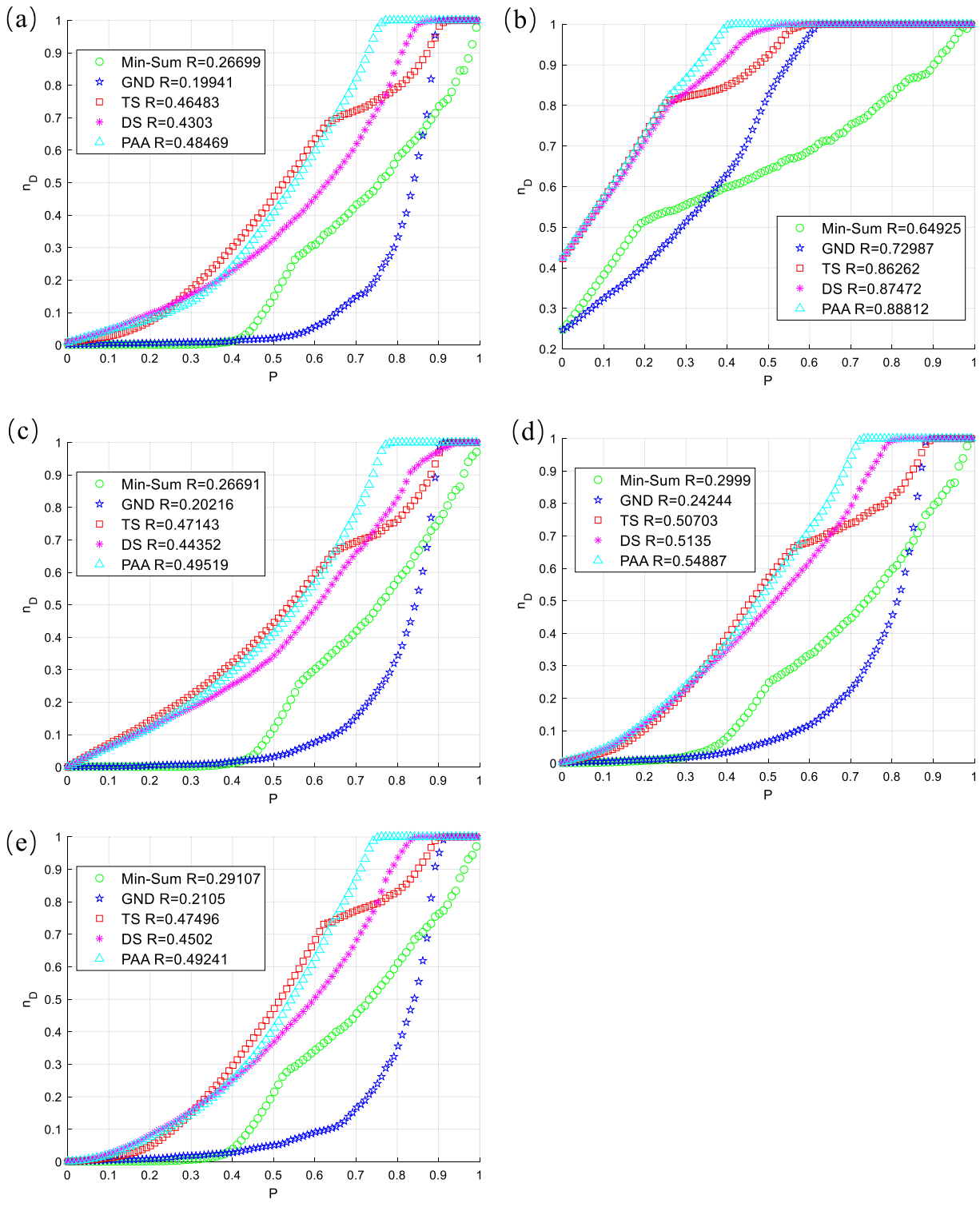

Fig. 11. The R-index values under different attack methods in various networks with 1000 nodes and an average degree of 5. (a) Random Graph (b) Scale-Free Network (c) *q*-snapback Network (d) Random Triangle Network (e) Random Rectangle Network.

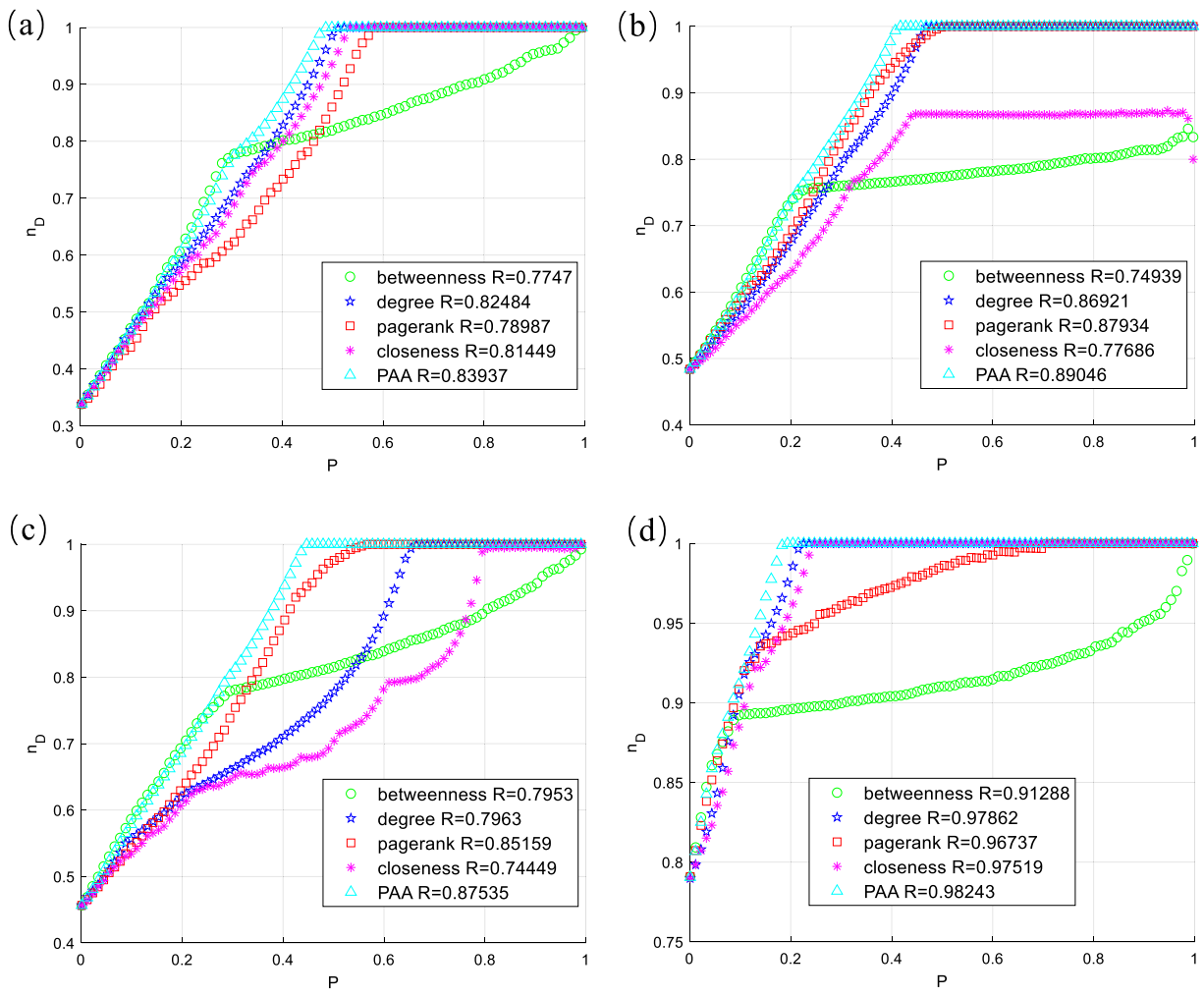

Fig. 12. The ratio of needed driver nodes n_D against the ratio of removed nodes P, on attacking the real-world networks. (a) US Airline Network (b) budding Yeast Network (c) crime-moreno Network (d) mouse-kasthuri-graph-v4 Network. (Different attack methods for each network, when P = 1, the value of n_d is 1).

Table 6
Real-world network information.

Network	Network type	N	$\langle k \rangle$
US Airlines	Infrastructure networks	332	12.80
budding yeast	Biological networks	2361	1.52
crime-moreno	Interaction networks	829	1.78
mouse-kasthuri-graph-v4	Brain-network	1029	1.15

network sizes and average degrees are presented in Table 6, while the detailed network information can be found in Network Repository [56, 57].

Here, $\mathcal{K}$ = {betweenness-based strategy, degree-based strategy, PageRank-based strategy}. The results of the five different attack methods are shown in Table 7 and Fig. 12. The accuracy rates in the real-world network are shown in Table 8.

The results show that PAA outperforms other attack strategies for all four real-world networks. Real-world networks do not have tunable nodes and edges number. Therefore, different real-world networks have multiple properties compared with synthetic networks. For example, US Airlines Network has a relatively high average degree. PAA can consistently recommend strategies that are applicable to the networks.

Additionally, some other networks with high heterogeneity, such as the mouse-kasthuri-graph-v4 Network. It can be observed that the network is completely dismantled after removing about 20% of the nodes after applying PAA. Overall, PAA outperforms other attack strategies for all four real-world networks and PAA's predictions accuracy on all four real-world networks reaches 94.14%.

In general, this also suggests that the robustness of real-world networks is not as strong as that of synthetic networks because the features of synthetic networks are strengthened. Meanwhile, attacking the real-world network is straightforward. From the perspective of attacking, the real-world networks need to enhance controllability robustness to withstand attacks.

6. Conclusions

In this paper, a probabilistic adaptive attack model for destructing network controllability is designed and investigated on both synthetic and real-world networks. Although the trade-off between exploring alternative strategies and focusing on choosing the most effective strategy is difficult, PAA tackles this problem by controlling the decay sliding window to adjust the size of shape parameters in the probability model. In this way, the suitable attack strategies are recommended according to their feedback to destroy the network controllability. The advantage of the method is the more destructive strategies get higher probability of being selected in subsequent attacks. The probabilities are also adaptively adjusted based on the attack effectiveness. The exploration and exploitation of using different attack strategies are delicately balanced.

Extensive simulation results show:

Table 7
The R-index values and ranks of attack simulations on the real-world networks. Strategies set $\mathcal{K}$ includes betweenness-based strategy, degree-based strategy, PageRank-based strategy.

Network	Method	R-index	Rank
US Airlines	Betweenness	0.774703	5
	Degree	0.824841	2
	PageRank	0.789869	4
	Closeness	0.81449	3
	PAA	**0.839367**	**1**
budding yeast	Betweenness	0.749391	5
	Degree	0.869213	3
	PageRank	0.879339	2
	Closeness	0.776858	4
	PAA	**0.890462**	**1**
crime-moreno	Betweenness	0.795303	4
	Degree	0.796303	3
	PageRank	0.851587	2
	Closeness	0.74449	5
	PAA	**0.875347**	**1**
mouse-kasthuri-graph-v4	Betweenness	0.91288	5
	Degree	0.978621	2
	PageRank	0.967366	4
	Closeness	0.975191	3
	PAA	**0.98243**	**1**
AVG	Betweenness	–	4.75
	Degree	–	2.5
	PageRank	–	3
	Closeness	–	3.75
	PAA	–	**1**

Table 8
Accuracy rate compared to the best in the real-world network.

Network	Accuracy
US Airlines	91.87%
budding yeast	93.99%
crime-moreno	92.46%
mouse-kasthuri-graph-v4	98.24%
AVG	94.14%

1. PAA can select the appropriate method for each stage of the network attack process, which reduces time consumption compared to many heavy computational algorithms, such as betweenness attack and PageRank, which need to cost a lot of time for one node/edge removal. PAA can switch among multiple algorithms to find the most effective method and ensure good performance. Although PAA spends time exploring different algorithm sets, its performance is still competitive with other attack algorithms.

2. PAA can avoid being stuck in local optima compared to other strategy selection methods. In a non-stationary environment, the rewards provided by the environment are not fixed. To avoid getting trapped in a local optimum, PAA use DSW and parameter decay to ensure the exploration capability of the algorithm so that other more efficient algorithms can be chosen in time.

3. PAA has an increased accuracy in selecting the optimal algorithm compared to other probabilistic model-based controllability attack methods. PAA have achieved a higher accuracy rate of 81.96% in algorithm selection compared to 73.17% for TS and 76.56% for DS. This highlights the effectiveness of PAA in finding the optimal attack method.

In the future, PAA can be further utilized to investigate the disruption of network structures, particularly in the context of motif attack scenarios. For instance, for various motif structures such as ring, chain and star, more attack scenarios can be explored to either strengthen or weaken the network's controllability. This research can provide us with a better understanding of the connection between network structure and control performance, as well as enhance the accuracy and effectiveness of network attacks, thus providing a more comprehensive security guarantee for networks.

CRediT authorship contribution statement

Sheng Li: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization. **Wenwen Liu:** Writing – review & editing, , Writing – original draft, Investigation, Formal analysis. **Ruizi Wu:** Writing – review & editing, Visualization. **Junli Li:** Supervision, Project administration, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (No. 62002249), in part by the Open Project Program of the State Key Lab of CAD&CG (A2112), Zhejiang University, and in part by the Foundation of Key Laboratory of System Control and Information Processing, Ministry of Education, P. R. China (No. Scip202103).

References

[1] Dehghani NL, Zamanian S, Shafieezadeh A. Adaptive network reliability analysis: Methodology and applications to power grid. Reliab Eng Syst Saf 2021;216:107973.
[2] Govindan K, Gholizadeh H. Robust network design for sustainable-resilient reverse logistics network using big data: A case study of end-of-life vehicles. Transp Res E 2021;149:102279.
[3] Wen T, Gao Q, Chen Y-w, Cheong KH. Exploring the vulnerability of transportation networks by entropy: A case study of Asia–Europe maritime transportation network. Reliab Eng Syst Saf 2022;108578.
[4] Cong Q, Anishchenko I, Ovchinnikov S, Baker D. Protein interaction networks revealed by proteome coevolution. Science 2019;365(6449):185–9.
[5] Lou Y, Wang L, Chen G. Local diversity–stability of the q-snapback network model. Phys A 2019;536:121020.
[6] Chen G, Lou Y. Naming game: models, simulations and analysis. Springer; 2019.
[7] Liu Y-Y, Slotine J-J, Barabási A-L. Controllability of complex networks. Nature 2011;473(7346):167–73.
[8] Wang XF, Chen G. Pinning control of scale-free dynamical networks. Phys A 2002;310(3–4):521–31.
[9] Yan G, Vértes PE, Towlson EK, Chew YL, Walker DS, Schafer WR, Barabási A-L. Network control principles predict neuron function in the caenorhabditis elegans connectome. Nature 2017;550(7677):519.
[10] Cetinay H, Devriendt K, Van Mieghem P. Nodal vulnerability to targeted attacks in power grids. Appl Netw Sci 2018;3(1):34.
[11] Zhang X, Liu D, Tu H, Tse CK. An integrated modeling framework for cascading failure study and robustness assessment of cyber-coupled power grids. Reliab Eng Syst Saf 2022;108654.
[12] Lou Y, Li J, Sheng L, Hao D. Recent progress in controllability robustness of complex networks (in Chinese). Acta Automat Sinica 2020;45:1–18.
[13] Holme P, Kim BJ, Yoon CN, Han SK. Attack vulnerability of complex networks. Phys Rev E 2002;65(5):056109.
[14] Shang Y. Generalized k-core percolation on correlated and uncorrelated multiplex networks. Phys Rev E 2020;101(4):042306.
[15] Wandelt S, Sun X, Feng D, Zanin M, Havlin S. A comparative analysis of approaches to network-dismantling. Sci Rep 2018;8(1):1–15.
[16] Iyer S, Killingback T, Sundaram B, Wang Z. Attack robustness and centrality of complex networks. PLOS ONE 2013;8(4):e59613.
[17] Lekha DS, Balakrishnan K. Central attacks in complex networks: a revisit with new fallback strategy. Phys A 2020;549:124347.
[18] Qin J, Xu JJ, Hu D, Sageman M, Chen H. Analyzing terrorist networks: A case study of the global salafi jihad network. In: International conference on intelligence and security informatics. Springer; 2005, p. 287–304.
[19] Cai Q, Pratama M, Alam S. Interdependency and vulnerability of multipartite networks under target node attacks. Complexity 2019;2019.
[20] Wang Z-G, Deng Y, Wang Z, Wu J. Disintegrating spatial networks based on region centrality. Chaos 2021;31(6):061101.

[21] Morone F, Makse HA. Influence maximization in complex networks through optimal percolation. Nature 2015;524(7563):65–8.
[22] Tian L, Bashan A, Shi D-N, Liu Y-Y. Articulation points in complex networks. Nature Commun 2017;8(1):1–9.
[23] Braunstein A, Dall'Asta L, Semerjian G, Zdeborová L. Network dismantling. Proc Natl Acad Sci 2016;113(44):12368–73.
[24] Šimon M, Dirgová Luptáková I, Huraj L, Host'oveckỳ M, Pospíchal J. Combined heuristic attack strategy on complex networks. Math Probl Eng 2017;2017.
[25] Wandelt S, Lin W, Sun X, Zanin M. From random failures to targeted attacks in network dismantling. Reliab Eng Syst Saf 2022;218:108146.
[26] Lou Y, Wu R, Li J, Wang L, Chen G. A convolutional neural network approach to predicting network connectedness robustness. IEEE Trans Netw Sci Eng 2021;8(4):3209–19.
[27] Liu Y-Y, Slotine J-J, Barabási A-L. Control centrality and hierarchical structure in complex networks. PLOS ONE 2012;7(9):e44459.
[28] Thomas J, Ghosh S, Parek D, Ruths D, Ruths J. Robustness of network controllability to degree-based edge attacks. In: International workshop on complex networks and their applications. 2016, p. 525–37.
[29] Wang L, Zhao G, Kong Z, Zhao Y. Controllability and optimization of complex networks based on bridges. Complexity 2020;2020:1–10.
[30] Song G-H, Li X-F, Lu Z-M. How centrality of driver nodes affects controllability of complex networks. IEICE Trans Inf Syst 2021;104(8):1340–8.
[31] Sun P, Kooij RE, He Z, Van Mieghem P. Quantifying the robustness of network controllability. In: 2019 4th international conference on system reliability and safety. ICSRS, IEEE; 2019, p. 66–76.
[32] Lou Y, Wang L, Chen G. A framework of hierarchical attacks to network controllability. Commun Nonlinear Sci Numer Simul 2021;98:105780.
[33] Chen G, Lou Y, Wang L. A comparative study on controllability robustness of complex networks. IEEE Trans Circuits Syst II 2019;66(5):828–32.
[34] Lou Y, He Y, Wang L, Chen G. Predicting network controllability robustness: A convolutional neural network approach. IEEE Trans Cybern 2020.
[35] Meng X, Han S, Wu L, Si S, Cai Z. Analysis of epidemic vaccination strategies by node importance and evolutionary game on complex networks. Reliab Eng Syst Saf 2022;219:108256.
[36] Fan D, Sun B, Dui H, Zhong J, Wang Z, Ren Y, Wang Z. A modified connectivity link addition strategy to improve the resilience of multiplex networks against attacks. Reliab Eng Syst Saf 2022;221:108294.
[37] Kalman RE. Mathematical description of linear dynamical systems. J Soc Ind Appl Math Ser A 1963;1(2):152–92.
[38] Yuan Z, Zhao C, Di Z, Wang W-X, Lai Y-C. Exact controllability of complex networks. Nature Commun 2013;4(1):1–9.
[39] Ruths J, Ruths D. Robustness of network controllability under edge removal. In: Complex networks IV. Springer; 2013, p. 185–93.
[40] Schneider CM, Moreira AA, Andrade Jr JS, Havlin S, Herrmann HJ. Mitigation of malicious attacks on networks. Proc Natl Acad Sci USA 2011;108(10):3838–41.
[41] Usman U, Mahmood A, Wang L. Robust control centrality. In: 2019 Chinese control conference (CCC). IEEE; 2019, p. 5486–91.
[42] Chen G, Lou Y, Wang L. A comparative study on controllability robustness of complex networks. IEEE Trans Circuits Syst II 2019;66(5):828–32.
[43] Chapelle O, Li L. An empirical evaluation of thompson sampling. Adv Neural Inf Process Syst 2011;24:2249–57.
[44] Maskooki A, Toldov V, Clavier L, Loscrì V, Mitton N. Competition: Channel exploration/exploitation based on a thompson sampling approach in a radio cognitive environment. In: EWSN-International conference on embedded wireless systems and networks (Dependability Competition). 2016.
[45] Shen W, Wang J. Portfolio blending via thompson sampling. In: IJCAI. 2016, p. 1983–9.
[46] Ouyang Y, Gagrani M, Jain R. Control of unknown linear systems with thompson sampling. In: 2017 55th annual allerton conference on communication, control, and computing (Allerton). IEEE; 2017, p. 1198–205.
[47] Gupta N, Granmo O-C, Agrawala A. Thompson sampling for dynamic multi-armed bandits. In: 2011 10th international conference on machine learning and applications and workshops, Vol. 1. IEEE; 2011, p. 484–9.
[48] Raj V, Kalyani S. Taming non-stationary bandits: A Bayesian approach. 2017, arXiv preprint arXiv:1707.09727.
[49] Erdős P, Rényi A. On the strength of connectedness of a random graph. Acta Math Acad Sci Hung 1964;12(1–2):261–7.
[50] Goh K-I, Kahng B, Kim D. Universal behavior of load distribution in scale-free networks. Phys Rev Lett 2001;87(27):278701.
[51] Sorrentino F. Effects of the network structural properties on its controllability. Chaos 2007;17(3):033101.
[52] Lou Y, Wang L, Chen G. Toward stronger robustness of network controllability: a snapback network model. IEEE Trans Circuits Syst I Regul Pap 2018;65(9):2983–91.
[53] Yang D, Liu M, Zhang Y, Lin D, Fan Z, Chen G. Henneberg growth of social networks: Modeling the facebook. IEEE Trans Netw Sci Eng 2020;7(2):701–12.
[54] Ren X-L, Gleinig N, Helbing D, Antulov-Fantulin N. Generalized network dismantling. Proc Natl Acad Sci 2019;116(14):6554–9. http://dx.doi.org/10.1073/pnas.1806108116, arXiv:https://www.pnas.org/content/116/14/6554.full.pdf URL: https://www.pnas.org/content/116/14/6554.
[55] Braunstein A, Dall'Asta L, Semerjian G, Zdeborová L. Network dismantling. Proc Natl Acad Sci USA 2016;201605083. http://dx.doi.org/10.1073/pnas.1605083113, URL: http://www.pnas.org/content/early/2016/10/18/1605083113.
[56] Rossi RA, Ahmed NK. The network data repository with interactive graph analytics and visualization. In: AAAI. 2015, URL: http://networkrepository.com.
[57] Bu D, Zhao Y, Cai L, Xue H, Zhu X, Lu H, Zhang J, Sun S, Ling L, Zhang N, et al. Topological structure analysis of the protein–protein interaction network in budding yeast. Nucleic Acids Res 2003;31(9):2443–50.