SCIENTIFIC REPORTS

OPEN

Multilevel Evolutionary Algorithm that Optimizes the Structure of Scale-Free Networks for the Promotion of Cooperation in the Prisoner's Dilemma game

Received: 29 September 2016
Accepted: 9 May 2017
Published online: 28 June 2017

Penghui Liu & Jing Liu

Understanding the emergence of cooperation has long been a challenge across disciplines. Even if network reciprocity reflected the importance of population structure in promoting cooperation, it remains an open question how population structures can be optimized, thereby enhancing cooperation. In this paper, we attempt to apply the evolutionary algorithm (EA) to solve this highly complex problem. However, as it is hard to evaluate the fitness (cooperation level) of population structures, simply employing the canonical evolutionary algorithm (EA) may fail in optimization. Thus, we propose a new EA variant named mlEA-C_{PD}-SFN to promote the cooperation level of scale-free networks (SFNs) in the Prisoner's Dilemma Game (PDG). Meanwhile, to verify the preceding conclusions may not be applied to this problem, we also provide the optimization results of the comparative experiment ($EA_{cluster}$), which optimizes the clustering coefficient of structures. Even if preceding research concluded that highly clustered scale-free networks enhance cooperation, we find $EA_{cluster}$ does not perform desirably, while mlEA-C_{PD}-SFN performs efficiently in different optimization environments. We hope that mlEA-C_{PD}-SFN may help promote the structure of species in nature and that more general properties that enhance cooperation can be learned from the output structures.

The Prisoner's Dilemma Game (PDG) is a popular abstract mathematical method and has been employed in biology to explain the emergence and persistence of cooperation behavior among selfish individuals[1–8]. After all, survival of the fittest is a widely accepted natural selection rule, and individuals employing the selfish strategy might be expected to be more likely to persist. After carefully studying PDG, researchers found that organisms may still form a cooperative community even if they all act entirely for their own interest. Even so, researchers still found it hard to explain large-scale cooperation in reality, as defection usually dominates in their simulations. To explain this puzzle, researchers have long been exploiting the deeper mechanisms.

In the past decades, network reciprocity, proposed by Nowak *et al.*, has had wide influence in this avenue of research. Individuals are constrained by spatial structure to play only with their immediate neighbors[8]. Nowak *et al.* concluded that topology constraints influence the evolution of cooperation (confirmed years later). After that, many extended studies have contributed to network reciprocity. In the early stages, researchers focused on the single layer networks: They found that population structure plays a determinate role in the evolution of cooperation[9] and cooperators in PDG are likely to form clusters to defend against defectors[10]. They revealed the potential positive relationship between cooperation and some network properties, such as heterogeneity[7, 11] and clustering coefficients[12]. And they also focused on how error and attack on the poulation structures may influence the evolution of cooperation[13]. Recently, researchers have analyzed the evolutionary game in interdependent networks, as populations in reality are not isolated and interaction exists between different layers[14–17]. These studies have reflected that interdependence may induce some new mechanisms that enhance cooperation and fixed the cooperation behavior on the system. And György *et al.* in ref. 18 reviewed how the population structure can modify long-term behavioral patterns in evolutionary games.

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, 710071, China. Correspondence and requests for materials should be addressed to J.L. (email: neouma@163.com)

In addition to studying how network reciprocity may influence the evolution of cooperation, some researchers have focused on investigating the potential behavior whereby players may adjust their interaction with others based on the gaming results. This is a natural phenomenon since population structure in reality may dynamically change during the game process. A representative method in this subject area is the coevolutionary rule, which was designed and proposed by Zimmermann *et al.* in ref. 19. Even if a large number of studies have contributed to this subject, most works in this area can be divided into those that employ strategy independent rules for connection adaptation[20–22] and those that take strategies or their performance as factors to influence the population reorganization[23, 24]. Perc *et al.* have also provided a review of this research in ref. 25.

While network reciprocity seems to have preliminarily explained large-scale cooperation in reality, some researchers have practically analyzed the real human game. Their experiment results revealed that humans do not base their strategy decisions on other's payoffs while playing PDG. In addition, Gracialázaro *et al.* in their experiments have found the existence of a population structure does not seem to have an influence on the global outcome of cooperation[26]. Following these experiments, researchers have also found that cooperation obviously depends on the strategy updating rule. Cimini *et al.* in ref. 27 have analyzed cooperation frequency in a simulation where different strategy updating rules are introduced. They found cooperation frequency assessed under the imitation-based strategy updating rule depends heavily on the population structure, but network reciprocity seems to have little effect on the game dynamics when individuals do not take neighbors' payoffs into consideration (non-imitative rule). These experiments and extended works seem to have put an end to network reciprocity. However, it remains difficult to conclude that population structure has little effect on promoting cooperation, as different strategy updating rules place different levels of emphasis on different game processes in nature. Evolutionary dynamics based on payoff comparisons are appropriate to model biological evolution, while they may not apply to social or economic contexts[26]. Moreover, Carlos *et al.* has emphasized in their research that their conclusion applies only to human cooperation, and network reciprocity may still be relevant to cooperation in other contexts.

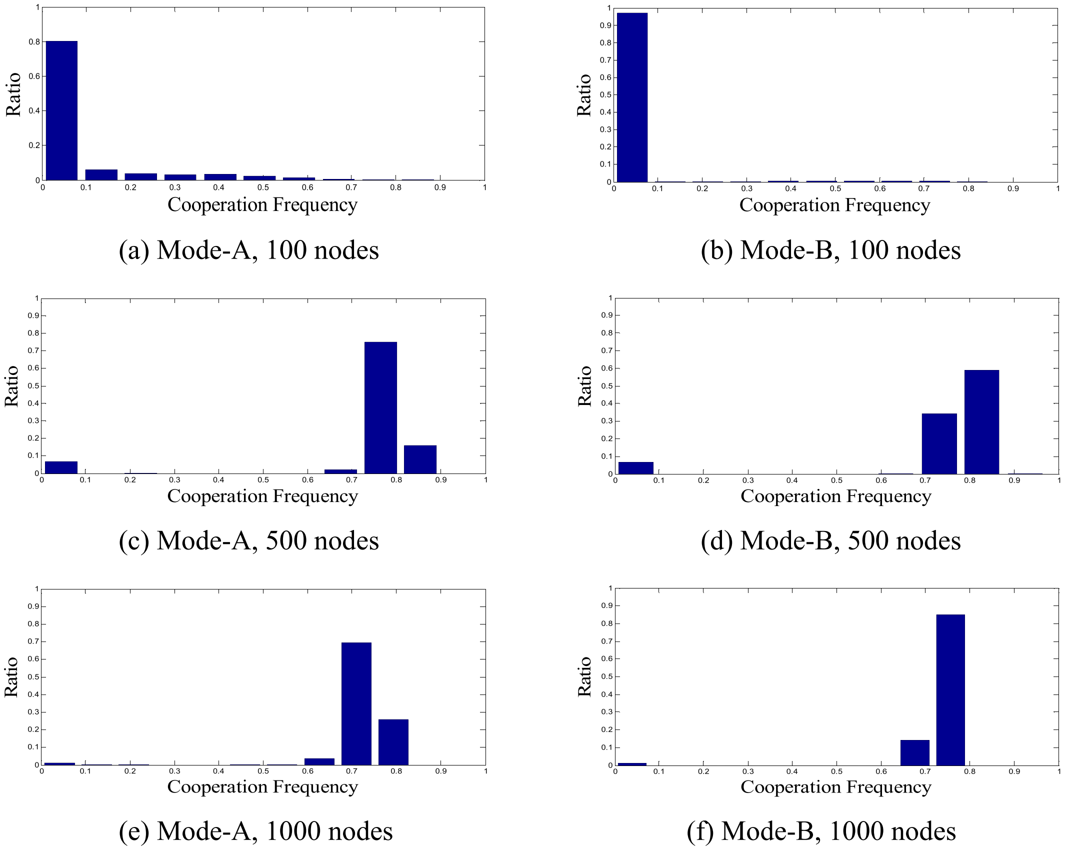
Therefore, even if the relevance between network reciprocity and cooperation in social or economic issues remains controversial, population structure is still essential to cooperation in biological evolution. Moreover, just as group selection indicates that cooperative groups may be more likely to survive in nature than uncooperative ones, cooperation is essential to the survival and evolution of species in nature. A high cooperation level can help species to maintain high competitiveness in nature, which may partially explain why helping family members finally helps the individual itself (Kin selection). Therefore, methods to help optimize the population structure should be important. Even if quite a lot of studies have contributed to the promotion of cooperation in the population, it remains difficult problem[28]. One representative breakthrough that has been made on this subject is the introduction of coevolutionary rules, which has provided a way to help understand the self-reorganizing ability of the population. Even so, little has been done to investigate how a population structure can be constructed or adjusted through man-intervention. And cooperation promotion methods that do not rely on the self-adapt ability of a population may shed light on this problem.

In this paper, we design a variant of evolutionary algorithm to optimize the population structure and thereby enhancing cooperation. Even if much of the preceding research has found a correlation of cooperation and some network properties, naively applying these conclusions to our problems may be quite problematic since these conclusions are mostly obtained based on specific network models and lack generality. Therefore, we employ the cooperation level of population structures as the objective value of our algorithm.

To our knowledge, no appropriate simple approach has been proposed to exactly determine the cooperation level of different population structures. Moreover, the evaluated cooperation level of structures actually fluctuates within a range. Therefore, the evaluation of structures within EA is fuzzy and may interfere with the selection of EA over the elite solutions. In the field of evolutionary computation, researchers also refer to this interference that leads to potential failure in optimization as the "EA cheat". Apparently, simply employing the canonical evolutionary algorithm (EA) may fail in optimization, even if EA have been applied to many engineering problems[29–36]. Even so, there are two widely accepted methods to reduce the evaluation error of different structures' cooperation level: (1) Prolong the simulation time of game evolution. (2) Average over independent evaluations. However, the corresponding computation cost cannot be ignored.

To successfully apply EA to the optimization of population structure, we propose a new EA variant named mlEA-C_{PD}-SFN to promote cooperation in the Prisoner's Dilemma Game (PDG). Within mlEA-C_{PD}-SFN, a modified local search operator named multilevel evolutionary operator is designed for the purpose of revising the wrong-filtering solution in EA population and exploiting solutions with potential higher cooperation levels. Therefore, we designed a memory structure (restoration list) within the operator to record some reliable solutions for the revision and some rules for the operator to control the search bias.

To test the performance of mlEA-C_{PD}-SFN, different types of scale-free structures have been employed, and optimization has been constrained not to change the initial degree distribution. Meanwhile, to verify that the preceding conclusions may not be applied to this problem, we also provide the optimization results of the comparative experiment ($EA_{cluster}$), which optimizes the clustering coefficient of structures. Even if the preceding research concluded that highly clustered scale-free networks enhance cooperation, we still find $EA_{cluster}$ cannot perform as satisfactorily as mlEA-C_{PD}-SFN does. Moreover, we also find that the mlEA-C_{PD}-SFN can perform well and simultaneously maintain a low computation cost (details in III). Finally, to verify the adaptability of mlEA-C_{PD}-SFN subject to different optimization environments, different strategy update rules are employed in our experiments. The simulation results verify the adaptability of mlEA-C_{PD}-SFN.



(a) Mode-A, 100 nodes

(b) Mode-B, 100 nodes

(c) Mode-A, 500 nodes

(d) Mode-B, 500 nodes

(e) Mode-A, 1000 nodes

(f) Mode-B, 1000 nodes

Figure 1. Evaluation distribution of tested population structure (BA network). In this experiment, population structures with different sizes have been tested. It is obvious that the distribution of cooperation frequency obtained under two different evaluation modes is similar (horizontal contrast) and the distribution of cooperation frequency gets broader when size of population gets smaller (vertical contrast).

Results

Prisoner's Dilemma Game and Evaluation of Population Structure.

Understanding the emergence of cooperation in the context of Darwinian evolution has been a challenge in recent decades. Even if relevant works in this field are almost entirely theoretical, it is quite likely to have broad-reaching implications for the future.

PDG is one of the most commonly used tools to help explain how cooperation endures in nature. In PDG, the defectors receive the highest reward T (temptation to defect) when defecting to a cooperator who receives the lowest payoff S (sucker value). If both of the players choose the same strategy, they receive a payoff R as a reward for cooperation or P as punishment for defection. Moreover, T, P, S, R follow the rule $T > R > P > S$. As a result, in a single round of PDG, defection is the best strategy no matter what the opponents' strategies are, even though all players would be better off if they all cooperate.

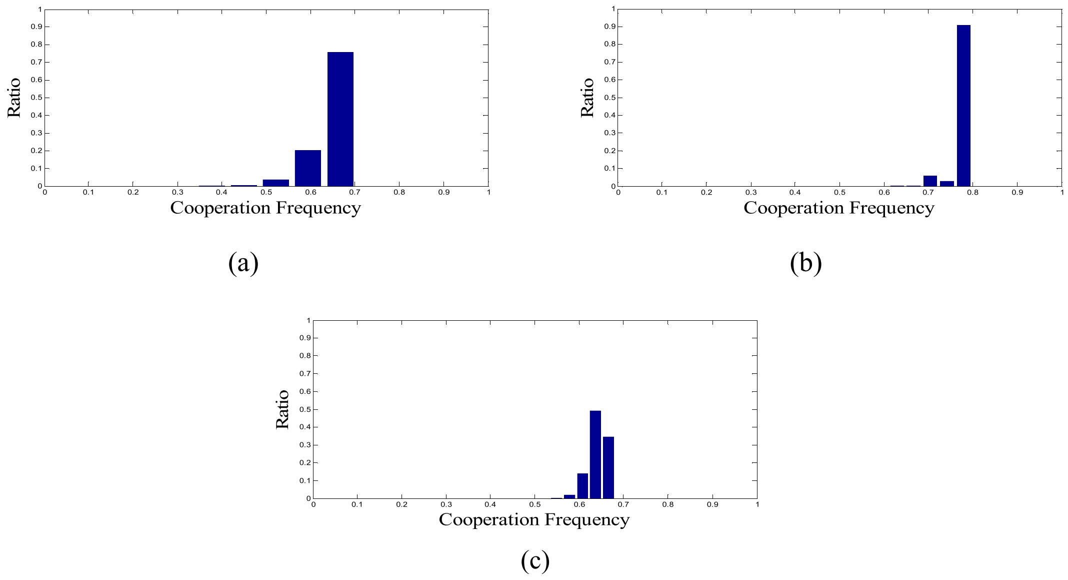
Population structures provide the basic organization of the game. In ref. 10, players interact only within a limited local neighborhood. When a site x is updated, the current occupant and all neighbors around compete to recolonize this site with their offspring. Those offspring keep the same strategy as their parents. The probability of neighbor y succeeding in reproduction is:

$$W_{s_x \leftarrow s_y} = \begin{cases} (P_y - P_x)/(Dd_>),\ P_y > P_x \\ 0,\ P_y < P_x \end{cases} \tag{1}$$

where $d_> = \max\{d_x, d_y\}$, $D = T$-S, d_i marks the degree of node i and P_i marks the payoff of i. Therefore, the probability that the focal individual succeeds in reproduction is $W_{s_x \leftarrow s_x} = \prod_l (1 - W_{s_x \leftarrow s_y})$. With probability $1 - W_{s_x \leftarrow s_x}$, an offspring of one neighbor takes over site x. The relative probability for the success of neighbor y is $W_{sx\leftarrow sy}/\sum_l W_{sx\leftarrow sy}$, where l marks neighbors of x. In this paper, the synchronous update method is employed.

Generally, individuals in a population are initially designated as cooperators or defectors with equal probability. And a corresponding cooperation level is obtained through averaging over generations after the equilibrium of a population is reached. Thus, the evaluated cooperation level of structures may naturally fluctuate among independent evaluations.

To illustrate this phenomenon, the distribution of evaluated cooperation level is given in Fig. 1. Each sub-graph in Fig. 1 contains 5000 independent evaluated results of a BA network. Meanwhile, two different evaluation modes have been designed to obtain these simulation results on the same group of population structures:



(a)

(b)

(c)

Figure 2. The evaluation distribution obtained under the multi-sampling method. Different sampling numbers have been tested. (**a**) 5 samplings, (**b**) 10 samplings, and (**c**) 20 samplings. On the whole, the multi-sampling method can help reduce the evaluation error of structures. However, different sampling numbers may be necessary toward different evaluation distributions of different structures, which explains why the fluctuation range in Fig. 2(c) is wider than in Fig. 2(b).

Mode-A and Mode-B. In Mode-A, each equilibrium cooperation level is obtained by averaging 0.1 N generations after a transient period of N generations. In Mode-B, each equilibrium cooperation level is obtained by averaging N generations after a transient period of 10 N generations.

Apparently, the independent evaluation results vary within a certain range, which may get wider with the decrease of the tested structure scale. Moreover, dramatically prolonging the simulation time cannot decrease the fluctuation range of evaluation in an obvious way. Meanwhile, the distributions of evaluated cooperation levels obtained under Mode-A and Mode-B are similar, which indicates that Mode-A and Mode-B both provide a necessary transient period for the equilibrium of population.

Therefore, the fluctuation of evaluation may interfere with the selection of EA over the elite solutions. As overlapping may exist between the evaluation distribution of different structures, there naturally exists a certain probability that a worse structure is mistaken by EA as the superior. This phenomenon is termed the "EA cheat" and is a knock down to the algorithm.

To avoid unnecessary computation cost, we employ Mode-A to evaluate the cooperation level of structures in our paper. Moreover, we introduce the multi-sampling method to average over independent evaluations, thereby approaching the ideal mean value of the evaluation distribution. To analyze the effect of this approach, the new evaluation distributions obtained under the multi-sampling method are provided in Fig. 2. On the whole, the multi-sampling method can help reduce the evaluation error of structures. However, different sampling numbers may be necessary toward different evaluation distributions of structures, which explains why the fluctuation range in Fig. 2(c) is wider than that in Fig. 2(b). Therefore, it is difficult to determine an appropriate sampling number, not to mention the fact that this method only decreases the fluctuation range without effectively dealing with the EA cheat (e.g., if the ideal mean value of two evaluation distributions is close enough).

Multilevel Evolutionary Operator and mlEA-CPD-SFN. Previous simulation results have revealed that the evaluation of structures may fluctuate and interfere with the selection of EA over the elite solutions. Multi-sampling may help reduce evaluation error. But determining an appropriate sampling number will be difficult. Moreover, the corresponding increase in computation cost is unbearable.

Therefore, the reserved (potential) structures should be repeatedly sampled more to ensure reliability of their evaluation results. However, the abandoned (mediocre) structures should be repeat sampled less to decrease computational cost. With this in mind, we propose a local search operator variant named multilevel evolutionary operator (see Methods) to achieve this. And for a timely revision of the wrong-filtering solution caused by the EA cheat, some reliable structures are saved as substitutes and a memory structure (see Methods) is designed. Given these, we further propose a new EA variant named mlEA-CPD-SFN (see Methods) to optimize the structure of scale-free networks for the promotion of cooperation in Prisoner's Dilemma game.

Optimizing Clustering Coefficient through EA. Assenza *et al.* have revealed the enhancement of cooperation in highly clustered scale-free networks[12]. Therefore, optimizing the clustering coefficient may help promote the cooperation level of scale-free structures. The objective value in $EA_{cluster}$ is the clustering coefficient of structures, which can be obtained as follows: Suppose neighbors of nodes i construct a graph represented

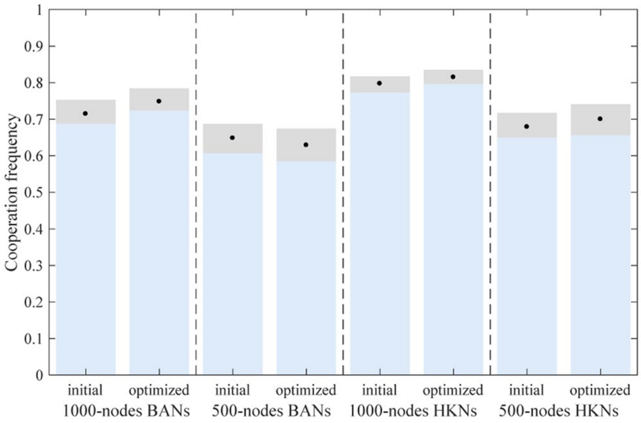


Figure 3. Simulation results of $EA_{cluster}$ in optimizing population structure for the promotion of cooperation in PDG. Each group contains 10 independent structures whose evaluated cooperation level lies within a gray bar (black points mark the mean value). On the whole, naively optimizing the clustering coefficient of a population structure may promote cooperation in PDG, but it is not efficient enough. Moreover, sometimes this method may fail and produce a worse structure, as shown in the optimization of 500-node BANs (the mean value gets smaller).

by Γ_i and its edge number is $|E(\Gamma_i)|$. Then, the clustering coefficient of node i would be $Cluster_i = |E(\Gamma_i)|/(d_i \times (d_i - 1)/2)$. Therefore, the clustering coefficient of a structure should be $Cluster = (\Sigma_i Cluster_i)/N$.

Two types of scale-free networks are employed to test the performance of $EA_{cluster}$: Barabási Albert networks (BANs)[37] and Holme and Kim networks (HKNs: $p = 1$). BANs is a type of commonly used scale-free network, while the HKNs is one of its variants and may have higher clustering coefficients in pairs with a higher p ($p \in [0,1]$) for construction[12]. For fair comparison with mlEA-C_{PD}-SFN, canonical local search operator is designed within $EA_{cluster}$, and the maximum generation of $EA_{cluster}$ is set to 120 where evolution of $EA_{cluster}$ has almost converged. The simulation results are given in Fig. 3.

Apparently, optimizing the clustering coefficient of scale-free structures can help promote cooperation in PDG. However, the efficiency of $EA_{cluster}$ is quite limited. Moreover, as $EA_{cluster}$ fails in the optimization of 500-node BANs, we can conclude that naively applying preceding research conclusions to the practical optimization of structures may not perform desirably and sometimes maybe problematic.

Efficiency of mlEA-C_{PD}-SFN in Optimizing Population Structure. In this part, the same structures are employed to test the performance of mlEA-C_{PD}-SFN (see Methods). Two types of mlEA-C_{PD}-SFN and a hybrid mlEA-C_{PD}-SFN are considered:

- Only one level in each pyramid: mlEA-C_{PD}-SFN_1 (shortened to lv1).
- Five levels in each pyramid: mlEA-C_{PD}-SFN_5 (shortened to lv5).
- Only one level in each pyramid but multi-sampling is employed for initial evaluation: mlEA-C_{PD}-SFN_1-M (shortened to lv1-M).

To protect the best record in the restoration list, α in mlEA-C_{PD}-SFN_1 is set to 0, while $\alpha = 0.5$ in mlEA-C_{PD}-SFN_5. Within mlEA-C_{PD}-SFN_1-M, initial evaluation of structures is obtained by averaging over 5 independent evaluations. The optimization results of these algorithms are given in Fig. 4.

The initial evaluation of structures in mlEA-C_{PD}-SFN_1 and mlEA-C_{PD}-SFN_5 contains only one sampling. Therefore, compared with mlEA-C_{PD}-SFN_1-M, these two mlEA-C_{PD}-SFNs in theory are more likely to overestimate or underestimate the cooperation level of structures. As the performance of mlEA-C_{PD}-SFN_1-M surpasses mlEA-C_{PD}-SFN_1, we can conclude that accurate evaluation of structures obviously influences the optimization results. This conclusion also explains why the performance of mlEA-C_{PD}-SFN_1 worsens when optimizing smaller structures. Even if the complexity of a problem synchronously descends with the structure scale, the corresponding evaluation error gets more obvious (refer to Fig. 1). Even so, in our experiments, mlEA-C_{PD}-SFN_5 performs better than mlEA-C_{PD}-SFN_1-M in general. This phenomenon reveals that the level of mlEA-C_{PD}-SFN is positive to its performance. The restoration list provides records to maintain the evolving of the EA population subject to the attack of "EA cheat". With the increase in level, more history records are available, and these records can be saved longer.

The evaluated cooperation level of structures provided in our paper is obtained by averaging over 5000 independent evaluations. Therefore, these simulation results should be reliable. However, further investigation regarding whether prolonging simulation time will influence our results should be undertaken, as those results are obtained under Mode-A with the purpose of saving computation cost. Thus, we prolong the simulation time and track the trend in cooperation during the game process (Fig. 5). Each data point within the simulation results is obtained through averaging over 500 independent runs (50 runs/structure). It is apparent that prolonging simulation time has little influence upon the evaluation results as cooperation frequency has almost converged around 1.1 N generation (Mode-A).Therefore, the simulation results we obtain should be credible. In addition, we can

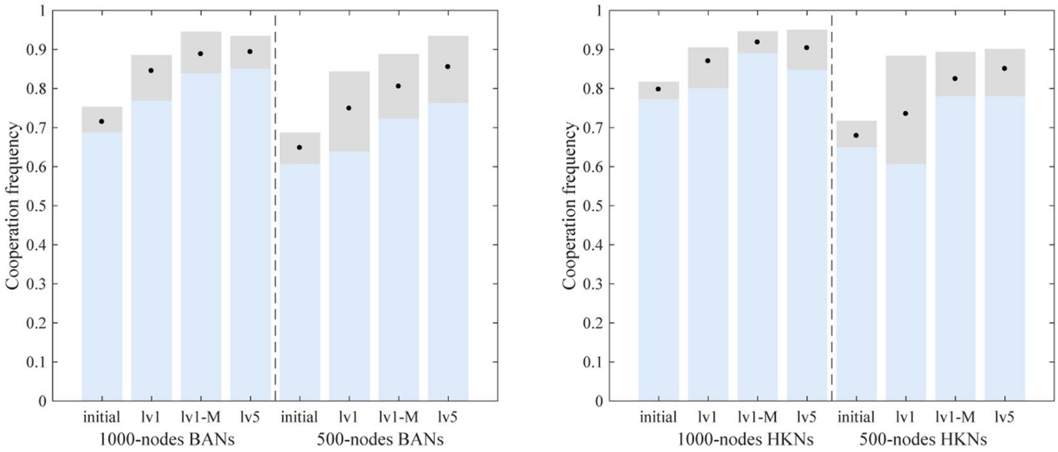
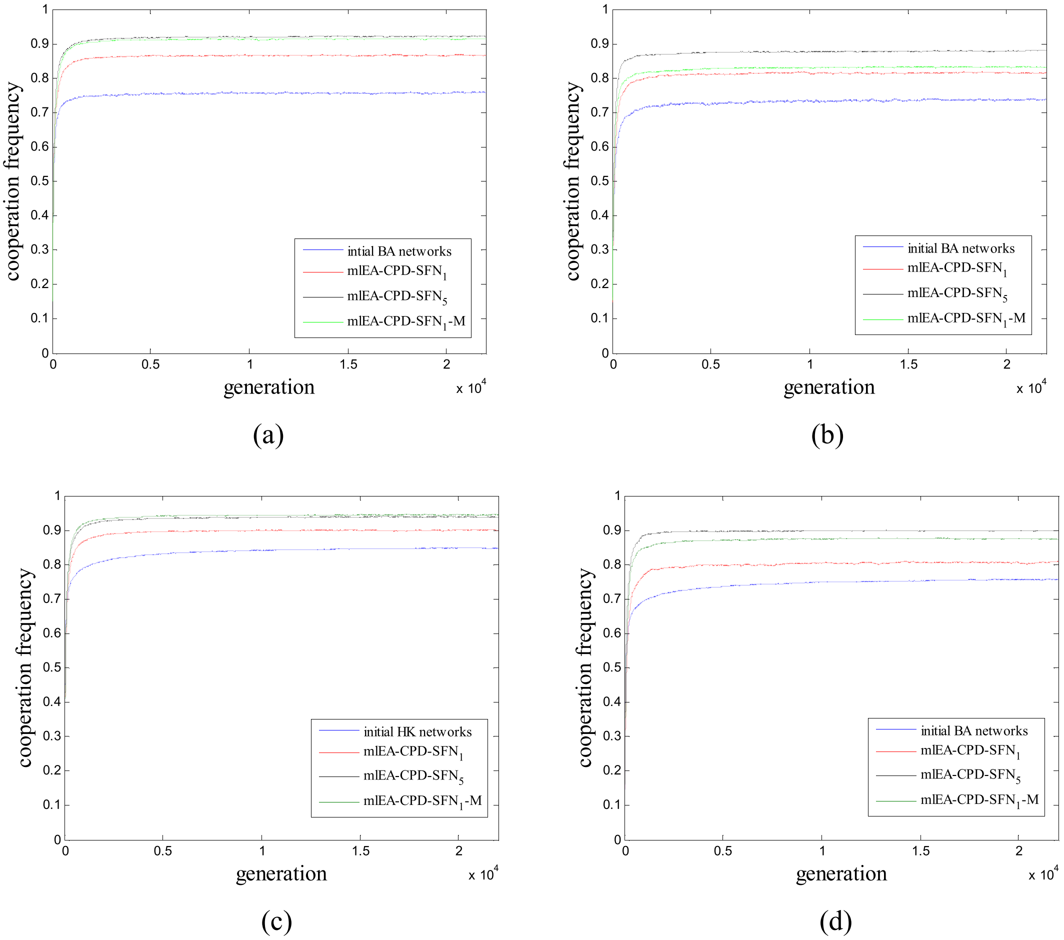


Figure 4. Simulation results of lv1, lv5 and lv1-M in optimizing population structure for the promotion of cooperation in PDG. Each group contains 10 independent structures whose evaluated cooperation level lies within a gray bar (black points mark the mean value). On the whole, these algorithms all efficiently promote cooperation in PDG, and the performance of mlEA-C_{PD}-SFN_5 surpasses those of the others.



(a)

(b)

(c)

(d)

Figure 5. Mean cooperation frequency of structures during the game process. (**a**) 1000-node BANs. (**b**) 500-node BANs. (**c**) 1000-node HKNs. (**d**) 500-node HKNs. Apparently, prolonging simulation time has little influence upon the evaluation results, as cooperation frequency has already converged around 1.1 N generation (Mode-A). Moreover, mlEA-C_{PD}-SFN_5 performs best in optimizing population structure and mlEA-C_{PD}-SFN_1-M follows. The optimization effect of these three algorithms is obvious.

parameter	mlEA-C_{PD}-SFN_1	mlEA-C_{PD}-SFN_1-M	mlEA-C_{PD}-SFN_5
Running time	18893 s	58429 s	20724 s
Evaluation number	143270	454800	135236

Table 1. The evaluation number and running time of these three algorithms.

conclude that mlEA-C_{PD}-SFN performs effectively in optimizing population structure and promoting cooperation in PDG (mlEA-C_{PD}-SFN_5 performs best, and mlEA-C_{PD}-SFN_1-M follows).

We further analyze the optimization characteristic of mlEA-C_{PD}-SFN and explore its advantages in optimization. Therefore, we provide the running time and evaluation number of the above three algorithms in Table 1.

Apparently, the computation cost of mlEA-C_{PD}-SFN_1-M surpasses those of the others (almost triple). Thus, the multi-sampling method is paired with a dramatic increase in computation cost. However, mlEA-C_{PD}-SFN_n performs well and simultaneously maintains a low computation cost. This is due to the basic principle of mlEA-C_{PD}-SFN: The reserved (potential) structures should be repeatedly sampled more to ensure reliability of their evaluation results. However, the abandoned (mediocre) structures should be repeat sampled less to decrease computational cost. In addition, the structure of the restoration list is irrelevant to the evaluation of structures.

Adaptability of mlEA-C_{PD}-SFN to Different Update Rules.

Previous research has concluded that update rules influence the cooperation level of structures[27]. Therefore, investigation of mlEA-C_{PD}-SFN's adaptability to different strategy update rules is necessary. Two common strategy update rules are additionally employed in our experiments:

- Fermi rule: A neighbor (Supposed as y) of x is chosen randomly. The imitation probability for y to learn from x is $W_{s_x \leftarrow s_y} = 1/(1 + \exp((P_y - P_x)/k))$, where P_i is the payoff of individual i and k denotes the amplitude of noise and is set to 0.1.
- Unconditional imitation rule: Each individual x imitates the neighbor y with the largest payoff, provided $P_y > P_x$.

The configuration of algorithms remains unchanged. As cooperation almost dominates upon HKNs($p = 1$) under the Fermi rule when the cost to benefit ratio $r = 0.95$, we partially employ HKNs($p = 0.5$) to test the performance of algorithms. Moreover, performance of mlEA-C_{PD}-SFN_{10} (10 levels in each pyramid: lv10) is also provided in Fig. 6. Notably, the simulation time to evaluate the initial and optimized structures in this part is set to 22000 generations, and the equilibrium cooperation level of structures is obtained by averaging the last 2000 generations (like the results in Fig. 5). Therefore, we do not further provide the corresponding mean cooperation frequency of structures during the game process.

Overall, these algorithms all successfully promote cooperation in PDG (Fig. 6). Therefore, we can conclude that mlEA-C_{PD}-SFN is adaptable to different strategy updating rules. Moreover, the level of mlEA-C_{PD}-SFN apparently influence its efficiency in performance. Finally, we can see mlEA-C_{PD}-SFN_1 apparently fails in some optimization, while mlEA-C_{PD}-SFN_1-M performs well. This phenomenon verifies that the evaluation error may cheat EA and thereby cause the failure of optimization.

Note that unlike other strategy updating rules, the unconditional imitation rule leads to a deterministic dynamic. Therefore, the initialized distribution of strategy plays an important role in the final cooperation frequency, and thereby, the evaluation of structures should be more unstable. This may explain why mlEA-C_{PD}-SFN_1-M performs best under the unconditional imitation rule even if these algorithms all perform effectively.

Discussion

Cooperation is essential in many aspects of life. In biology, the prisoner's dilemma game (PDG) has long been used to help explain how cooperation endures in nature. As cooperation is highly relevant to the competitiveness of groups in nature, understanding cooperation in PDG and proposing methods of promotion should have significant implications. To our knowledge, although many well-known mechanisms have provided ways to understand the self-reorganizing ability of a population toward an optimal situation for cooperation, little has been done to investigate how to construct or adjust the population structure with man-intervention, even if the preceding research recognizes its importance. Therefore, cooperation optimization methods that do not rely on the self-regulation mechanism of a population may shed light on this problem.

The contributions of this paper are summarized as follows: (1) We propose a new EA variant named mlEA-C_{PD}-SFN to optimizes the structure of scale-free networks for the promotion of cooperation in the Prisoner's Dilemma game without changing the initial structures' degree distribution. (2) We reveal that evaluation error of population structures may cause the "EA cheat" and canonical evolutionary algorithm (EA) may fail in optimization. (3) Different types of scale-free structures and updating rules have been applied to verify the performance of mlEA-C_{PD}-SFN. (4) We provide the optimization results of the comparative experiment ($EA_{cluster}$) and reveal that naively applying preceding research conclusions to the practical optimization of structures may not perform desirably and sometimes maybe problematic. (5) The experimental results show that mlEA-C_{PD}-SFN can perform well in various situations and simultaneously maintain a low computation cost.

We hope mlEA-C_{PD}-SFN may help promote the structure of species in nature and that more general properties that enhance cooperation can be learned from the output structures.

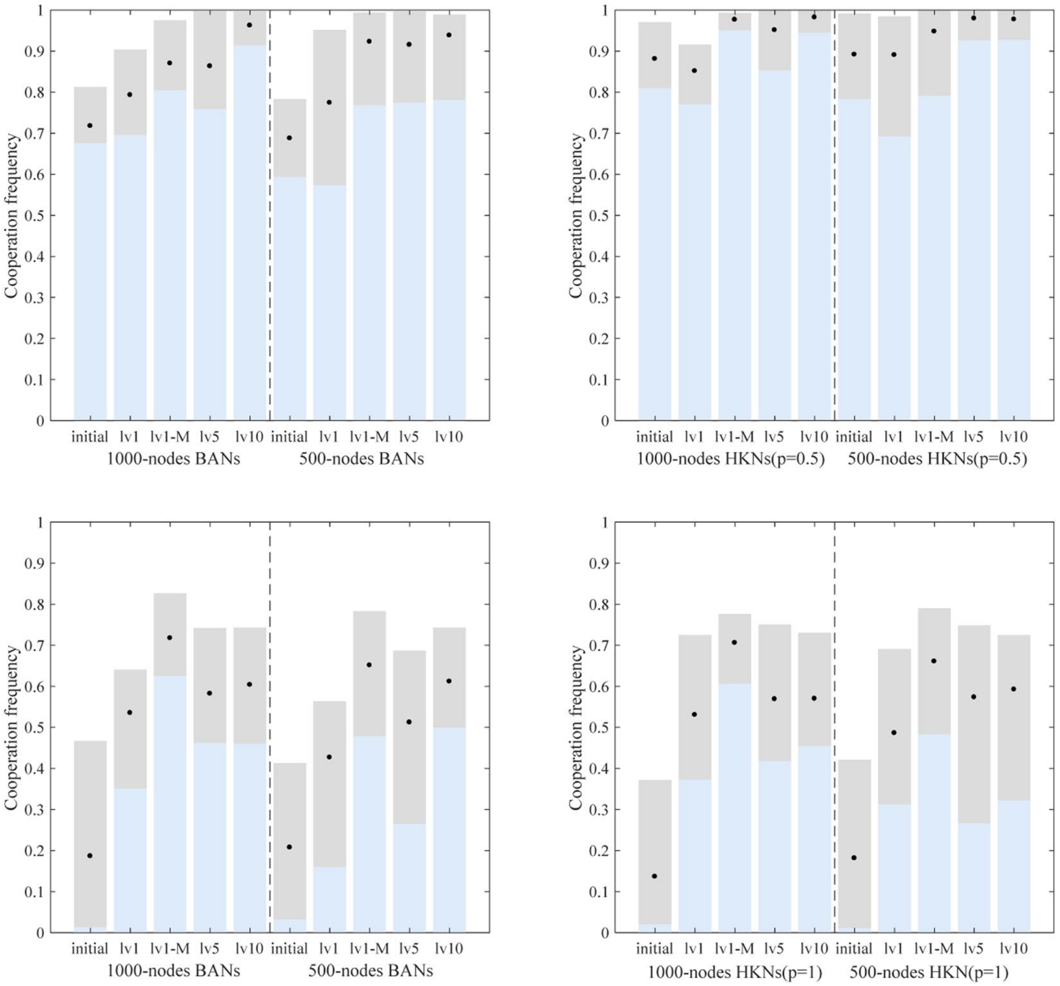


Figure 6. Simulation results of lv1, lv1-M, lv5 and lv10 in optimizing population structure for the promotion of cooperation in PDG. Each group contains 10 independent structures whose evaluated cooperation level lies within a gray bar (black points mark the mean value). The results in the upper (bottom) row are obtained under the Fermi rule (unconditional imitation rule). Overall, these algorithms all effectively promote cooperation in PDG. mlEA-C_{PD}-SFN_{10} performs best under the Fermi rule, while mlEA-C_{PD}-SFN_1-M performs best under the unconditional imitation rule.

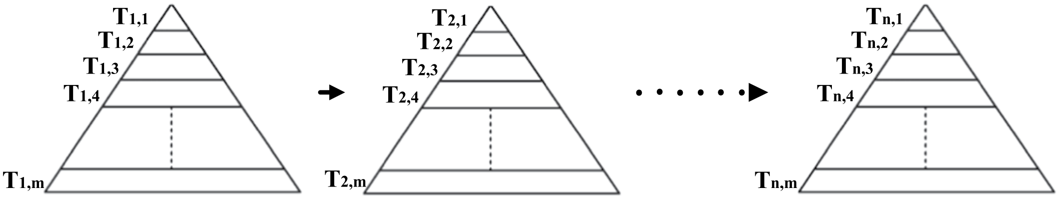


Figure 7. Restoration list.

Methods

Restoration List. Even if the "EA cheat" may lead to some wrong-filtering solution in the EA population, timely revision still may save EA from failure in optimization. Therefore, a hierarchical memory structure (given in Fig. 7) is designed to backup some reliable solutions in case of need. There are 2Ω pyramid-like sub-lists in the restoration list, and each contains some vacant spaces for records. Notably, records in the upper pyramid are prior to those in the bottom.

$L_{n,m}$ marks the *m*th record of the *n*th pyramid, and L_n marks the *n*th pyramid. L_n is responsible for *n*th solution in the EA population. The reason to separate the memory structure into numbers of parts is: (1) Maintain the diversity of the EA population and to avoid premature convergence of algorithm. (2) Maintain the parallelism of the EA. (3) Enable communication among pyramids with the help of computing characteristic of EAs.

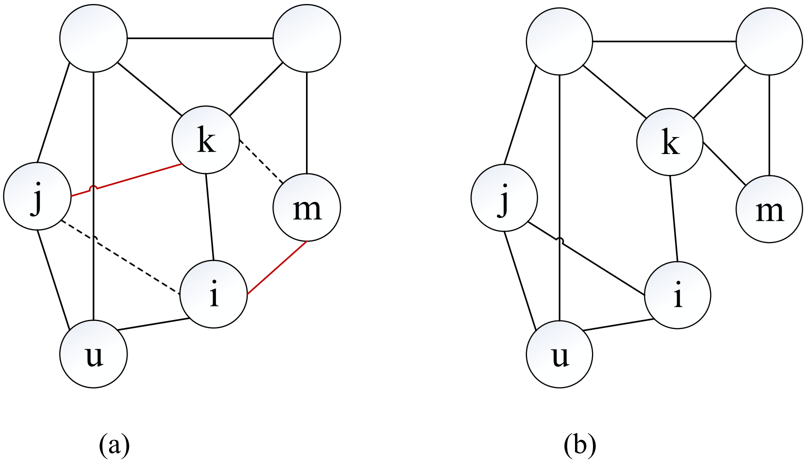


Figure 8. Structure adjusted before and after edge switching. (**a**) Initial structure G_i. (**b**) Structure adjusted (G_i'). Remove e_{jk} and e_{im} from structure G_i in (**a**); then, e_{ji} and e_{km} are added, thereby gaining a new structure G_i' in (**b**).

Basic Strategies for Multilevel Evolutionary Operator.

The basic rules for multilevel evolutionary operator are: (1) Some supplementary rules for the restoration list. (2) Rules to compare structures with different sampling numbers. Note that *sum* marks the sum of evaluation results and *num* marks the number of evaluations (sampling number); e.g.: if a structure has been evaluated twice, $num = 2$ and *sum* is the total of the evaluation results. Therefore $I_i = \{G_i, sum_{Gi}, num_{Gi}\}$ marks the *i*th solution in the EA population. $G = (V, E)$ is used to represent a population structure (graph), where $V = \{v_1, v_2, \ldots, v_N\}$ is the set of nodes and $E = \{e_{ij} \mid i, j \in V \text{ and } i \neq j\}$ is the set of edges in the structure.

(1) Four rules are designed for the restoration list: *insertion rule*, *mutation rule*, *sorting rule*, and *information update rule*.

- The *insertion rule*: Suppose we attempt to backup the *i*th solution to the restoration list. If there are vacant spaces in $\boldsymbol{L_i}$, this solution is inserted directly at the bottom of $\boldsymbol{L_i}$. Otherwise, this solution should clear out some old records with smaller $avg = sum/num$ and *num* first. If no vacant space is ready, this solution will not be backed up.
- The *mutation rule*: Suppose *i*th solution has a larger *avg* than $L_{i,m}$. Even if the insertion rule is unsatisfied, it may still take the place of $L_{i,m}$ with probability $\alpha^{max\text{-}m}$ (*max* is the number of the vacant space in L_i).
- The *sort rule*: Solutions in pyramids are prioritized in terms of their *avg*, and *sort_list*(*i*) marks the sorting operation on the *i*th pyramid.
- The *information update rule*: The backup should be updated synchronously if the *sum* or *num* of its ontology is updated and *record_synchronize*($L_{i,m}$, *sum*, *num*) is used to denote the update operation of $L_{i,m}$.

(2) Two comparison strategies are designed to compare structures with different sampling numbers. **Strategy** 1 is used to compare solutions in the EA population and **strategy** 2 is used to compare the solutions in the EA population and restoration list.

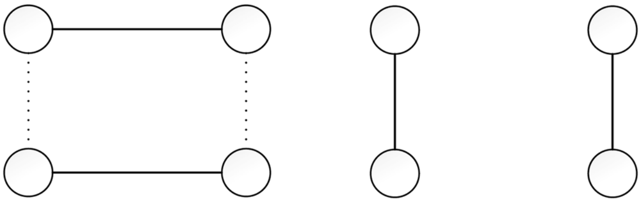
Strategy 1 (I_i and I_j): if $avg_{Gi} > avg_{Gj}$, I_i is better than I_j.

Strategy 2 (I_i and I_j (record)): We suppose G_i' is a variant from G_i. (i) Only when $avg_{Gi} < avg_{Gi'} < avg_{Gj}$, I_j is considered more optimal than I_i. (ii) Only when $avg_{Gi'} > avg_{Gi} > avg_{Gj}$, I_i is considered more optimal than I_j. Details of how G_i' is produced are given below.

Multilevel Evolution Operator: A Variant from the Local Search Operator.

Local search operator is a widely employed method to improve the exploitation of EA through continuous fine-tuning of solutions. In our paper, we employed an edge switching process (in Fig. 8) to complete the fine-tune of the current solution G_i and obtain G_i': (a) A node *u* ($d_u \geq 2$) with its two neighbors *i* and *j* (d_i, $d_j \geq 2$) are selected. (b) Edges e_{jk} and e_{im} ($u \neq i \neq j \neq k \neq m$) are selected. (c) Remove e_{jk}, e_{im} and add e_{ji}, e_{km} ($e_{ji}, e_{km} \notin G_i$). *edge_switch*($G_i$, v_u, v_j, v_i, v_k, v_m) marks this edge switching operation (step c), and *node_select*(v_u, v_j) marks the node selecting process (step a-b).

Canonical local search operator only compares and selects between the initial and adjusted solutions in terms of their evaluation results. However, the records in the restoration list are also considered by the multilevel evolutionary operator (Suppose I_i, I_i' mark the initial solution and adjusted solution, while $L_{i,0}$ is the top level record in the *i*th pyramid):



(a) (b)

Figure 9. Structures before and after being adjusted by simple edge swapping. (**a**) Initial structure. (**b**) Structure adjusted. The solid lines mark the current existing edges, while the dotted lines mark the targeted edges to be constructed.

- If $avg_{Gi} > avg_{Gi'}$: (a) I_i' is abandoned, and I_i is reserved. (b) Evaluate the structure of I_i once more, and update *sum* and *num*. (c) Synchronize information if its backup exists.
- If $avg_{Gi'} > avg_{Gi} > avg_{Li}$: (a) Backup I_i. (b) I_i is replaced by I_i'. (c) Sort records in L_i.
- If $avg_{Li} > avg_{Gi'} > avg_{Gi}$: Replace I_i with $L_{i,0}$ and *restoration*(i) denotes this operation.
- For other situations, the multilevel evolution operator replaces I_i with I_i'.

Algorithm 1 Provide details of the multi-level evolutionary operator and β marks the probability scale.

```
Algorithm 1: Multilevel Evolutionary Operator
Input: $I_k$: kth solution in EA population, $avg_{Gk}$: Evaluation of $G_k$; $\beta$: probability scale.
Output: $\tilde{I}_k$': modified solution from $I_k$. $avg_{\tilde{I}k'}$: Evaluation of $\tilde{I}_k$';
Step 1: Initialization, see algorithm 2 for more information.
Step 2: Local search based on the hill-climbing, iterates over every edge of nodes.
(a) Edge switching (see algorithm 3 for more information).
(b) If the adjusted solution is unreasonable, conduct algorithm 4.
(c) If the adjusted solution is accepted, operates according to the specific situations. See algorithm 5 for more information.
 (d) If the loop is over go to step 3, else go to the next round.
Step 3: Output current $I_k$ and $avg_{GK}$ as $\tilde{I}_k$' and $avg_{\tilde{I}k'}$;
```

```
Algorithm 2: Initialization
$Flag_{List} \leftarrow false$;//$Flag_{List}$ marks whether the current structure has already been backed up;
$sum_{Gk} \leftarrow avg_{Gk}$,
$num_{Gk} \leftarrow 1$;
```

```
Algorithm 3: Edge switching adjustment
if($U(0,1) < \beta \times d_i / \sum_i d_i$) then
  [Is_Nodes_Found, $v_i$, $v_a$, $v_b$, $v_c$, $v_d$]$\leftarrow$node_select($v_i$, $v_a$);
  if(Is_Nodes_Found = false) then
    continue;
  else
    $G_k$‘ $\leftarrow$ edge_switch($G_k$, $v_i$, $v_a$, $v_b$, $v_c$, $v_d$);
else
  continue;
end if;
$avg_{Gk}$‘ $\leftarrow$ $C(G_k$‘$)$;//$C(G_k$‘$)$ evaluate cooperation level of $G_k$‘ once;
```

```
Algorithm 4: Operation when edge switching adjustment is rejected
if($G_k$‘ is not connected or $avg_{Gk'} < avg_{Gk}$) then
  $num_{Gk} \leftarrow num_{Gk} + 1$;
  $sum_{Gk} \leftarrow sum_{Gk} + C(G_k)$;
  $avg_{Gk} \leftarrow sum_{Gk} / num_{Gk}$;
  if($Flag_{List}$) then
    record_synchronize (i, sum, num)
  end if;
end if;
```

Algorithm 5: Operation when edge switching adjustment is accepted

```
//$avg_{Gk'} > avg_{Gk}$: adjustment accepted;
if($avg_{Gk} > avg_{Li}$) then//$avg_{Gi'} > avg_{Gi} > avg_{Li}$;
  Attempt to back-up $G_k$ in $L_k$;
  $sort\_list(k)$;
  $avg_{Gk} \leftarrow avg_{Gk'}$;
  $num_{Gk} \leftarrow 1$;
  $sum_{Gk} \leftarrow avg_{Gk'}$;
  $Flag_{List} \leftarrow false$;
  $G_k \leftarrow G_k'$;
elseif($avg_{Li} > avg_{Gk'}$) then//$avg_{Li} > avg_{Gi'} > avg_{Gi}$
    $restoration(k)$;
    $is\_list\_taken \leftarrow true$;
else
  $avg_{Gk} \leftarrow avg_{Gk'}$;
  $num_{Gk} \leftarrow 1$;
  $sum_{Gk} \leftarrow avg_{Gk'}$;
  $Flag_{List} \leftarrow false$;
  $G_k \leftarrow G_k'$;
end if;
```

Implementation of mlEA-C_{PD}-SFN.

Given a targeted population structure for optimization, we first generate an EA population $\{G_1, G_2, \ldots, G_\Omega\}$ through a simple edge swapping operation (Fig. 9). Crossover operators of mlEA-C_{PD}-SFN is similar to the crossover operation in[33] **Algorithm 6** provides the details of mlEA-C_{PD}-SFN.

Algorithm 6: mlEA-C_{PD}-SFN

Input: G_0: Initial scale-free structure. *GS*: Population size. P_c: Crossover rate. *max*: Levels in pyramids. *gen_max*: Iteration number. β: probability scale.

Output: G^*: The optimized population structure.

Step 1: Initialize population through *simple_edge_swapping*(G_0) and evaluate their cooperation level;

Step 2: EA population reproduce (Crossover operator). Select parents through roulette selection.

Step 3: Mutation (*simple_edge_swapping*) to the offspring and evaluate their cooperation level.

Step 4: Multilevel evolutionary operator performs upon both the parent and the offspring.

Step 5: If the iteration generation satisfies, output the best solution in EA population. Or else, select the best *GS* solutions in the EA population for the next generation and go to step 2.

Some parameters in experiments are fixed as: $P = 0$, $T = 1 + r$, $S = 0$, $R = 1$, $\beta = 20$, $r = 0.95$ (details of cost to benefit ratio r in ref. 10), $GS = 6$ and $gen\_max = 30$. Notably, the cooperation level of structures provided in **Results** is obtained by averaging over 5000 independent evaluations.

References

1. Chiong, R. & Kirley, M. Effects of iterated interactions in multiplayer spatial evolutionary games. *IEEE Trans. on Evolutionary Computation* **16**, 537–555 (2012).
2. Li, J. & Kendall, G. The effect of memory size on the evolutionary stability of strategies in iterated prisoner's dilemma. *IEEE Trans. on Evolutionary Computation* **18**, 819–826 (2014).
3. Wedekind, C. & Milinski, M. Cooperation through image scoring in humans. *Science* **288**, 850–852 (2000).
4. Fehr, E. & Fischbacher, U. The nature of human altruism. *Nature* **425**, 785–791 (2003).
5. Nowak, M. A. & Sigmund, K. Evolution of indirect reciprocity. *Nature* **437**, 1291–1298 (2005).
6. Nowak, M. A. Five rules for the evolution of cooperation. *Science* **314**, 1560–1563 (2006).
7. Chen, X., Fu, F. & Wang, L. Prisoner's dilemma on community networks. *Physica A: Statistical Mechanics and Its Applications* **378**, 512–518 (2007).
8. Nowak, M. A. & May, R. M. Evolutionary games and spatial chaos. *Nature* **359**, 826–829 (1992).
9. Santos, F. C., Rodrigues, J. F. & Pacheco, J. M. Graph topology plays a determinant role in the evolution of cooperation. *Proceedings of Royal Society of London B: Biological Sciences* **273**, 51–55 (2006).
10. Hauert, C. & Doebeli, M. Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature* **428**, 643–646 (2004).
11. Santos, F. C. & Pacheco, J. M. Scale-free networks provide a unifying framework for the emergence of cooperation. *Physical Review Letters* **95**, 098104 (2005).
12. Assenza, S., Gómez-Gardeñes, J. & Latora, V. Enhancement of cooperation in highly clustered scale-free networks. *Physics Review E* **78**, 017101 (2008).
13. Perc, M. Evolution of cooperation on scale-free networks subject to error and attack. *New Journal of Physics* **11**, 033027 (2009).
14. Wang, Z., Szolnoki, A. & Perc, M. Evolution of public cooperation on interdependent networks: The impact of biased utility functions. *Epl.* **97**, 48001 (2012).
15. Wang, Z., Wang, L., Szolnoki, A. & Perc, M. Evolutionary games on multilayer networks: a colloquium. *The European Physical Journal B* **88**, 124–138 (2015).
16. Kenett, D. Y., Perc, M. & Boccaletti, S. Networks of networks – An introduction. *Chaos, Solitons & Fractals* **80**, 1–6 (2015).
17. Wang, Z., Szolnoki, A. & Perc, M. Optimal interdependence between networks for the evolution of cooperation. *Scientific Reports* **3**, 2470 (2013).

18. György, S. & Gábor, F. Evolutionary games on graphs. *Physics Reports* **446**, 97–216 (2007).
19. Zimmermann, M. G., Eguíluz,V. M. & Miguel, M. S. Cooperation, adaptation and the emergence of leadership. *Economics with heterogeneous interacting agents*, *Lecture Notes in Economics and Mathematical Systems* (2001).
20. Szolnoki, A. & Perc, M. Resolving social dilemmas on evolving random networks. *Epl* **86**, 30007 (2009).
21. Szolnoki, A. & Perc, M. Emergence of multilevel selection in the prisoner's dilemma game on coevolving random networks. *New Journal of Physics* **11**, 093033 (2009).
22. Szolnoki, A., Perc, M. & Danku, Z. Making new connections towards cooperation in the prisoner's dilemma game. *Europhysics Letters* **21**, 88–121 (2008).
23. Pacheco, J. M., Traulsen, A. & Nowak, M. A. Active linking in evolutionary games. *Journal of Theoretical Biology* **243**, 437–443 (2006).
24. Pacheco, J. M., Traulsen, A. & Nowak, M. A. Coevolution of strategy and structure in complex networks with dynamical linking. *Physical Review Letters* **97**, 258103 (2006).
25. Perc, M. & Szolnoki, A. Coevolutionary games–a mini review. *Bio Systems* **99**, 109–125 (2010).
26. Gracialázaro, C. *et al.* Heterogeneous networks do not promote cooperation when humans play a Prisoner's Dilemma. *Proceedings of the National Academy of Sciences of the United States of America* **109**, 12922–12926 (2012).
27. Cimini, G. & Sanchez, A. How Evolutionary Dynamics Affects Network Reciprocity in Prisoner's Dilemma. *Journal of Artificial Societies and Social Simulation* **18**, 22–35 (2015).
28. Shivshankar, S. & Jamalipour, A. An Evolutionary Game Theory-Based Approach to Cooperation in VANETs Under Different Network Conditions. *IEEE Transactions on Vehicular Technology* **64**, 2015–2022 (2015).
29. Zhou, M. & Liu, J. A two-phase multi-objective evolutionary algorithm for enhancing the robustness of scale-free networks against multiple malicious attacks. *IEEE Trans. on Cybernetics* **47**, 539–552 (2017).
30. Chi, Y. & Liu, J. Learning of fuzzy cognitive maps with varying densities using a multi-objective evolutionary algorithm. *IEEE Trans. on Fuzzy Systems* **24**, 71–81 (2016).
31. Zhong, W., Liu, J., Xue, M. & Jiao, L. A multi-agent genetic algorithm for global numerical optimization. *IEEE Trans. on Systems, Man and Cybernetics, Part B* **34**, 1128–1141 (2004).
32. Liu, C., Liu, J. & Jiang, Z. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *IEEE Trans. on Cybernetics* **44**, 2274–2287 (2014).
33. Zhou, M. & Liu, J. A memetic algorithm for enhancing the robustness of scale-free networks against malicious attacks. *Physica A: Statistical Mechanics and its Applications* **410**, 131–143 (2014).
34. Yuan, Y., Xu, H., Wang, B., Zhang, B. & Yao, X. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Trans. on Evolutionary Computation* **20**, 180–198 (2016).
35. Qiu, X., Xu, J. X., Tan, K. C. & Abbass, H. A. Adaptive cross-generation differential evolution operators for multiobjective optimization. *IEEE Trans. on Evolutionary Computation* **20**, 232–244 (2016).
36. He, J. & Lin, G. Average convergence rate of evolutionary algorithms. *IEEE Trans. on Evolutionary Computation* **20**, 316–321 (2015).
37. Barabàsi, A. L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).

Acknowledgements

This work is partially supported by the Outstanding Young Scholar Program of National Natural Science Foundation of China (NSFC) under Grant 61522311 and the Overseas, Hong Kong & Macao Scholars Collaborated Research Program of NSFC under Grant 61528205.

Author Contributions

Penghui Liu and Jing Liu devised the research project; Penghui Liu performed numerical simulations; Penghui Liu analyzed the results; Penghui Liu and Jing Liu wrote the paper.

Additional Information

Competing Interests: The authors declare that they have no competing interests.