Journal of Computers Vol. 33 No. 4, August 2022, pp. 107-119
doi: 10.53106/199115992022083304009

A Discrete Particle Swarm Optimization Algorithm Based on Neighbor Cognition to Solve the Problem of Social Influence Maximization

Qi-Wen Zhang, Qiao-Hong Bai*

School of Computer and Communication, Lanzhou University of Technology University, Lanzhou 730050, China
tdyy2010@126.com, 875008642@qq.com

Received 20 September 2021; Revised 12 January 2022; Accepted 13 February 2022

Abstract. In view of the problem that the estimation method of node influence in social network is not comprehensive and the Particle Swarm Optimization (PSO) algorithm is easy to fall into the local optimal and the local search ability is insufficient. In this paper, we proposed a Neighbor Cognitive Discrete Particle Swarm Optimization (NCDPSO) algorithm. Aiming at the problem of influence in social networks, a new node influence measure method is proposed, the three-degree theory is introduced to comprehensively estimate the influence of nodes. In order to improve the global search ability of the PSO, the "neighbor cognition" factor is proposed to enhance the breadth of learning; and the following bee strategy is introduced to propose particle density and survivability to control the number of elite clones, so as to solve the problem of insufficient local search ability of the algorithm. Finally, the validity of the proposed algorithm is verified by testing on real data sets and comparing with other algorithms.

Keywords: influence maximization, three degree theory, neighbor cognition, PSO, elite cloning

1 Introduction

In recent years, the emergence of complex networks, such as rumor containment, prevention and treatment of infectious diseases, and other issues are closely related to the identification of influence nodes. How to find the most influential node set in a complex network, namely influence maximization has become a very important research work.

At present, the research algorithms for the maximization of social influence are divided into two categories: greedy algorithms and heuristic algorithms. Because heuristic algorithms have higher solution efficiency for large-scale networks, they have received extensive attention from researchers, such as Liu et al. [1] proposed a new algorithm for MNH combination, namely greedy algorithm with heuristic algorithm. As one of the heuristic algorithms, PSO [2] has been widely used in recent years to address the influence of social networks, due to its consistency with the dynamic properties of networks. For example, Gong et al. [3] proposed DPSO algorithm, and based on the idea that the influence of nodes in the two-hop region accounts for a large proportion of their global influence, they constructed a LIE function model of local influence estimation to evaluate the propagation capability of each node in the network. Wang et al. [4] proposed a discrete particle swarm optimization algorithm to find the ensemble of the maximum fitness function. At the same time, in order to accelerate the convergence, a degree-based population initialization method and a local search strategy based on mutation learning are introduced. Yang et al. [5] proposed a multi-objective discrete particle swarm optimization algorithm, which can consider individuals and their influences at the same time and can better describe the characteristics of the real network. Zhou et al. [6] combined the PSO method of social network GDM based on trust relationship to make influence propagate and maximize in trust relationship. Tang et al. [7] proposed ELDPSO, which uses local search strategy based on greedy mechanism to improve the optimal solution. Singh et al. [8] proposed a discrete particle swarm optimization based on learning automata, which uses local optima to avoid premature convergence and produces more efficient seed sets to maximize influence propagation. In addition, other scholars proposed other heuristic algorithms to solve the problem of social network influence. For example, Cui et al. [9] used the degree drop strategy to select nodes to generate new node sets, and proposed the degree drop differential evolution influence maximization algorithm based on the differential evolution algorithm. Jiang et al. [10] proposed an expected propagation value (EDV) to approximate the influence spread of node combination, and proposed a simulated annealing-based influence maximization algorithm. Sankar et al. [11] proposed a swarm intelligence algorithm based on the study of bee waddle dance behavior to maximize its influence, and suggested that understanding

* Corresponding Author

swarm intelligence in biological society might be a practical method algorithm for maximizing the influence of design efficiency. Simsek et al. [12] proposed a joint heuristic strategy to reshape network nodes so that the swarm intelligence optimization algorithm can obtain a general slope in the state space of its objective function. Qiu et al. [13] proposed a local influence decreasing differential evolution algorithm. In addition, a local influence descent search strategy is proposed, which can obtain the set of nodes with relatively large influence of each node. The algorithm can generally improve the accuracy and computational efficiency of the swarm intelligence-based influence maximization algorithm.

PSO algorithm is used widely, and it also has some achievements in the study of social network influence maximization. On the basis of the original algorithm, aiming at the problem of maximizing the influence of social network, the NCDPSO algorithm for maximizing the influence of social network is proposed. The technical achievements of this paper can be summarized as follows.

(1) First of all, in response to the problem of the influence of social networks, this paper proposes a new influence measurement method based on the "three degrees theory". For a node, the influence of all nodes within its three hops on it is considered, and to better improve its ability to spread its influence.

(2) Secondly, in order to improve the global search ability of PSO, the "neighbor cognition" particle is introduced into the algorithm as the learning object of the current particle, so that individuals do not jump too far so that all particles converge to quickly, which maintains the continuous search ability of the population. Moreover, in sociology, people usually consider the experience of others as the basis for decision-making. In addition, based on the follower bee theory in the artificial bee colony algorithm, this paper proposes an elite clone strategy for neighborhood search to improve the problem of insufficient local space search in PSO. The particle density and survivability are defined to control the number of elite clones, which can also improve the quality of the solution.

(3) Finally, experimental results simulated on social network data sets demonstrate the effectiveness and superiority of the proposed algorithm.

The rest of this paper is organized as follows. Section 2 describes the influence maximization, propagation models, traditional influence estimation functions, and improves influence estimation. The details of the improved algorithm are then presented in section 3. Finally, experimental results and conclusions are made in section 4 and section 5, respectively.

2 Related Work

2.1 Influence Maximization

Influence maximization can be defined as a problem on how to select k nodes from the network G so that the influence spread σ(S) is maximal. Given a graph $G = (N, E)$ and an integer $k < |N|$, selecting k nodes as the initial node set $S = \{ s_1, s_2, ..., s_k | s_i \in N, i = 1, 2, ..., k\}$, so that the influence spread σ(S) is maximum for a specific spread model, which can be formulated as (1).

$$\begin{cases} S = \arg max\, \sigma(S) \\ subject\ to\ |S| = k \end{cases} . \tag{1}$$

2.2 Propagation Model

There are many models that can be used to simulate the diffusion process with maximum influence, such as independent cascade (IC) models, weighted cascade models, and linear threshold (LT) models. The IC model only considers the relationship between the current active node and the inactive nodes in its immediate neighbors, and does not consider the influence of other active nodes on the inactive nodes. Therefore, it can well simulate the characteristics of social network influence propagation in real situations, so it has received extensive attention and research by scholars. Based on the above reasons, in the research process of the influence maximization problem, the IC model is used as the model of influence propagation.

2.3 The Traditional Approximate Estimation Model of Influence

At present, researchers have proposed a series of influence estimation functions, such as EDV proposed by Jiang; a fast approximation method of influence diffusion proposed by Lee and Chuang [14]; Wang designed an effective fitness function based on local influence to Estimation of influence diffusion, that is, the direct neighbors of the seed set are the main factors affecting the spread of integrated circuit models; Qiu [15] proposed LFV to estimate the influence value of a single node; Gong proposed LIE and Qiu proposed EDIV, EDIV combines the LFV function to calculate the two-hop range of local effects, etc. Liu et al. [16] proposed a new centrality metric algorithm, which is not only based on the nearest neighbors of a node, but also takes into account the neighbor nodes within two and three hops of a node. The methods proposed by EDV, LFV and Lee only calculate the expected number of direct neighbor nodes affected by the seed set, without considering the local influence of each node in the single-hop area of the seed set, and LIE calculates the influence diffusion in the two-hop area of the node. Although EDIV uses different parameters to reflect the influence of each hop, it still does not fully consider other influencing factors of the node. Although these influence estimation functions describe the influence of nodes in different ways, and considering the attenuation effect of information propagation in real social networks, some also assign different weights to the influence of one hop and two hops. However, most researchers still seldom consider the influence of neighboring nodes beyond the node's second hop on the node, so the research on this issue is worth exploring.

2.4 Improved Influence Estimation Function

There are measures based on global and local attributes for calculating influence spread. The node influence measurement based on global attributes can better reflect the topological characteristics of nodes, while the measurement based on local attributes is simple, intuitive and less time complexity, which is suitable for large-scale networks. However, such indicators only consider the influence of nodes in terms of the number of nodes that may affect other nodes, and do not consider the difference between the intensity of influencing other nodes or the location of nodes in the entire network. According to the three degree theory proposed by sociologist Fowler and others [17], nodes can affect not only neighbor nodes (one degree) but also neighbor nodes (two degrees) of neighbor nodes, and even neighbor nodes of neighbor nodes (three degrees) of neighbor nodes, as long as within three degrees are strong connections, there is the possibility of triggering behavior. Beyond three degrees, the influence of the nodes on each other disappears. In addition, the study of communication dynamics [18] shows that the spread of influence in social networks follows what we call the rule of three degrees of influence. Based on the above analysis, the influence diffusion in the three-hop area of the node set S is used to improve the influence spread range. The spread of the three-hop region of the node propagates as (2).

$$\sigma_3^*(S) = \sigma_2^*(S) * \frac{1}{\left|N_S^{(2)} \backslash S\right|} \sum_{u \in N_S^{(3)} \backslash S} p_u^* \, q_u^* \quad . \tag{2}$$

Where, $\sigma_2^*(S)$: the influence spread expectation of the candidate seed set S in its second-order neighborhood. $N_S^{(2)}$: the second-order neighborhood node set of the candidate seed set S. $N_3^{(3)}$: the third-order neighborhood node set of the candidate seed set S. p_u^*: is a small propagation activation probability in a given propagation model. q_u^* is the number of edges between $N_S^{(2)}$ and $N_S^{(3)}$.

In summary, the improved influence estimation function LIE is as follows (3).

$$\begin{aligned} LIE(S) &= \sigma_0(S) + \sigma_1^*(S) + \sigma_2^*(S) + \sigma_3^*(S) \\ &= k + \left(1 + \frac{1}{\left|N_S^{(1)} \backslash S\right|} \sum_{u \in N_S^{(2)} \backslash S} p_u^* \, d_u^* \left(1 + \frac{1}{\left|N_S^{(2)} \backslash S\right|} \sum_{u \in N_S^{(3)} \backslash S} p_u^* \, q_u^*\right)\right) \quad . \\ &* \textstyle\sum_{i \in N_S^{(1)} \backslash S} \left(1 - \prod_{(i,j) \in E, j \in S} \left(1 - p_{i,j}\right)\right) \end{aligned} \tag{3}$$

Where, $\sigma_0(S)$ is the k most influential nodes in the seed set, $\sigma_1^*(S)$ is the expected influence spread of one-hop area of the seed set. $\sigma_2^*(S)$ is the expected influence spread of two-hop area of the seed set. $\sigma_3^*(S)$ is the expected influence spread of three-hop area of the seed set. $N_S^{(1)}$, $N_S^{(2)}$ and $N_S^{(3)}$ represent the S's one-hop, two-hop and three-hop area, respectively. The parameter p_u^* is the constant active probability of node i, and it corresponds to p in the IC model. d_u^* is the number of edges of node u within $N_S^{(1)}$ and $N_S^{(2)}$ and q_u^* is the number of edges of node u within $N_S^{(2)}$ and $N_S^{(3)}$, which represent the number of activated probability for node u. $p_{i,j}$ represents the activation probability of node i activating j. Obviously, the problem of selecting k most influential nodes in formula (1) to maximize their influence propagation range is transformed into the problem of selecting a group of candidate seed nodes with a size of k to maximize the adaptive value of the evaluation function of influence propagation expectation described in formula (3).

3 NCDPSO Algorithm

Through the study of particle behavior in PSO, it is found that particle swarm optimization has the defect that it is easy to fall into the local optimum, because sometimes the PSO will miss the global optimal solution in the process of subduction. The subduction of the particle in flight makes its search behavior not fine enough, and it is not easy to find the global optimal target value, which shows that the particle is at a standstill before the update of the optimal particle. Therefore, the neighborhood search method of "neighbor cognition" is introduced, which makes the particle search more precise, and enhances the continuous search ability of the population. In addition, in order to solve the problem of insufficient local search ability, a neighborhood search operation of elite clone is proposed. The method of cloning is carried out by using the Logistics chaotic sequence; the number of clones is calculated by the defined survivability.

3.1 Coding

In this algorithm, the position vector x_i of each particle in the population is encoded by real value. A set k of nodes represents the feasible solution, represented by $[x_1, x_2, \ldots, x_k]$. Where, the i node in the solution x_i is represented by the node number in the social network graph. The 0-1 decision-making mechanism can be used to control the direction of particles' flight. For example, $V_i = (0, 0, 1, 0, 0)$ represents the velocity vector of particles i, a 0-1 decision-making mechanism can be used to control the flying direction of the particle, 1 means that it needs to be updated or replaced by other nodes; 0 means that it can be temporarily reserved as a candidate seed node.

3.2 Population Initialization

In order to speed up the convergence of the proposed NCDPSO algorithm, a degree based heuristic method is adopted to perform initialization for particles' position vectors. First select the k nodes in the graph G with the highest degree value, denoted as *Degree* (G, k), and then generate a random number for each element of the position vector in the interval [0, 1] , if the random number is greater than 0.5, we will randomly select a node from the node set N that is different from the other nodes in the position vector and represents $Replace(x_{ij}, N)$, with the update as shown in (4).

$$\begin{array}{ll} x_{ij} \leftarrow Replace(x_{ij}, N) & if\ random > 0.5\ . \\ x_{ij} \leftarrow x_{ij} & otherwise\ . \end{array} \tag{4}$$

3.3 "Neighbor Cognition" Factors

For the problem of insufficient global search ability in PSO, a "neighbor cognition" factor is defined to expand its search space. Based on the mechanism of communication within the flock, biologists and physicists consider the range of individual interactions to be within a certain distance. As in [19] shows that a bird interacts with six or seven birds on average in order to maintain a balance between energy conservation and close communication. Similar to the influence of social networks, the influence of friends and family on the self is a more common phenomenon. And many studies have shown that individuals who are affected in many aspects have more influence than individuals who are affected in only one or two aspects. The learning strategy combined with "neighbor cognition" can explore more search space and improve the global search ability of the algorithm to avoid premature convergence, because any particle stuck in local optima can learn from other particles and eventually escape local optima. The modified velocity update formula is as follows (5).

$$V_i = \omega V_i + c_1 r_1 (Pbest_i - X_i) + c_2 r_2 (Gbest - X_i) + c_3 r_3 (Nbest - X_i) \quad . \tag{5}$$

Neighbor cognition *Nbest* is defined as the average value of 7 neighbor particles, which is expressed as (6).

$$Nbest = 1/7 \sum_{i=1}^{7} pbest_i \quad . \tag{6}$$

3.4 Elite Cloning

Elite clone operation has been widely used in various heuristic algorithms. For example, in order to improve the mining and exploration ability of DE algorithm, Wu [20] added clone and mutation mechanism of elite group, and adopted dynamic selection method to determine elite group, which effectively enhanced the global search ability of the algorithm. Xiao [21] proposed the elite genetic algorithm of quantum cloning, which combined the concept of quantum revolving gate in quantum computing and cloning in biology to avoid the algorithm falling into local optimal, and adopted the elite strategy to accelerate the algorithm convergence. Therefore, in view of the insufficient local search ability of particle swarm optimization algorithm, the oscillation phenomenon, easy to disperse, difficult to obtain accurate results and low search efficiency phenomenon. Through the follower bee theory in the artificial bee colony algorithm [22], that is, the follower bee searches for new food sources near the food source based on the information transmitted by the lead bee, and when a good food source is found, the corresponding lead bee is notified to update its food source, and the following bees can improve the accuracy of the solution. Based on the follower bee theory, this paper proposes an elite cloning operation.

Definition 1. Particle density $\boldsymbol{P_i}$. Set a radius $\boldsymbol{r}$, calculate the number of particles in this radius $\boldsymbol{m}$, $\boldsymbol{m}$ and the total number of particles N ratio that is density, such as (7).

$$P_i = m / N \quad . \tag{7}$$

Definition 2. Survivability $\boldsymbol{S_i}$. The survivability $\boldsymbol{S_i}$ of a particle is the weight of the distance between any particle and the optimal particle and the particle density, such as (8).

$$S_i = \mathrm{r1} * \frac{1}{d_{i,Pg}} + r2 * \frac{1}{\rho_i} \quad . \tag{8}$$

Where, r1, r2 ϵ [0, 1], $d_{i,Pg} = |x_i - x_{Pg}|$ in the formula, $d_{i,Pg}$, P_i are between 0 and 1, $1/d_{i,Pg}$ the larger, that is, the closer the particle to the optimal value, the higher the particle fitness; P_i the larger, that is, the higher the number of particles clustered around the particle, the higher the density, continuing the search will result in local optima, so explore other areas. For the survivability S_i of the particle, the larger of S_i, the more adaptive it is, and vice versa, so the survivability of the cloning operation determines the number of clones.

The number of clones N_c is calculated from the following (9).

$$N_c = \sum_{i=1}^{n} round(S_i n / i + b) \quad . \tag{9}$$

Where, n the number of randomly selected elite particles, S_i ϵ [0, 1], in order to avoid the number of clones less than 0 so the addition b, b is an integer greater than or equal to 1, *round* is the integer function.

The cloning method is as follows: select a few elite particles at random, and conduct random and regular cloning with Logistic sequence, such as (10).

$$X_i^{new} = X_i + (X_i^{max} - X_i^{min}) / l * U_{r+1} \quad . \tag{10}$$

Where, X_i^{max} is the most influential particle in the area of elite particle density and X_i^{min} is the smallest particle in the area of elite particle density. l is a constant, depending on the specific problem. U_{r+1} is a chaotic sequential Logistic, where U_{r+1} the map of Logistics is a chaotic sequence is as follows (11).

$$U_{r+1} = \mu U_r (1-U_r) \ . \tag{11}$$

Where, r = 0, 1, 2, ..., $0 < U_0 < 1$, μ is the state control parameters of the system, it has been proved that when μ=4, the initial value $U_0 \notin$ (0.25, 0.5, 0.75, 1), the system represented by the above formula is completely in a chaotic state, U_{r+1} traversing the (0,1) range.

3.5 Selection Operations for Equivalence Partitioning

After the above operation between the population particles, the number of the population has already exceeded the set value, need to carry out further selection of particles in the population. The selection process of particles is as follows, using the equivalent partition strategy to ensure the diversity of the population. Suppose the number of particles in a population is N, the influence function is sorted from large to small, and all particles are divided into n disjoint intervals and recorded as N_1, N_2, ..., N_n, then a particle is randomly selected in each interval to form a new population, for the next search, effective guarantee of the selection of individuals more uniform.

3.6 Particle Update Rules

1) Update rule for velocity: the updating rule of the velocity vector is redefined in a discrete as follows (12).

$$V_i \leftarrow H\big(\omega V_i + c_1 r_1 (Pbest_i \cap X_i) + c_2 r_2 (Gbest \cap X_i) + c_3 r_3 (Nbest \cap X_i)\big) \ . \tag{12}$$

Where, ω is the inertia weight, $c_1 c_2 c_3$ are the cognitive factors, which is a constant.$r_1 r_2 r_3$ are the random numbers between [0, 1].

The operator "∩" in (12) is defined as a similar intersection operation. A detailed illustration for this operator is shown in Fig. 1. The position vector of particle i is X_i, $Pbest_i$ represents the personal best position of particle i and *Gbest* is the global best position in the swarm, *Nbest* is the neighbor average best position of particle i. Firstly, intersect vector X_i and $Pbest_i$ mathematically and the same elements in both two vectors can be obtained. Then the corresponding elements in V_i^{Pbest} are set to 0, which denotes that those directions may be potential and tends to be reserved. Conversely, the other elements in V_i^{Pbest} are set to 1, which indicates that those directions are not good choice and they need to be adjusted. In a similar way, then V_i^{Gbest} and V_i^{Nbest} can be also computed. The two position vectors finally get a velocity vector through position intersection operation. The essence of the "∩" operator is to guide the particles in the population to learn from $Pbest_i$, *Gbest* and *Nbest*. Through the position intersection operation, the particles can clarify how to change their current position to obtain a more accurate solution.

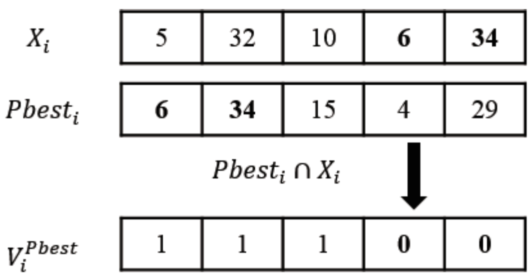


Fig. 1. Discrete evolutionary rules

H(·) is a velocity decision function to calculate velocity V_i, which consists of three 0-1 vectors. Assuming that the parameter is X_i, the function $H(X_i)$ it can be represented as $H(X_i) = (h_1(x_{i1}), h_2(x_{i2}), ..., h_k(x_{ik}))$ and $h_i(x_{ij})(1 \le j \le k)$ is defined as a threshold function shown in (13).

$$h_i(x_{ij})\begin{cases}0 & if\ x_{ij} < 2 \\ 1 & if\ x_{ij} \geq 2\end{cases} \quad . \tag{13}$$

For example, supposing that $c_1r_1 = 0.8$, $c_2r_2 = 1.3$, $c_3r_3 = 0.9$, according to the discrete evolution rule of the Fig. 1, it can be obtained $V_i = H(0.8 * (1, 0, 1, 0, 1) + 1.3 * (0, 0, 1, 1, 0) + 09 * (1, 0, 0, 0, 1)) = H((1.7, 0, 2.1, 1.3, 1.7)) = (0, 0, 1, 0, 0)$.

2) Update rule for position: the position vector X_i of the particle is updated by (14).

$$X_i \leftarrow X_i \oplus V_i \quad . \tag{14}$$

"$\oplus$" is the arithmetic operation between the position vector and the velocity vector. We define it as a "replacement" operator. Assuming that $X_i' = x_{i1}', x_{i2}', ..., x_{ik}'$ elements in the new position X_i' can be updated by (15).

$$x_{ij}' = \begin{cases}x_{ij} & if\ v_{ij} = 0 \\ Replace(x_{ij}, N) & if\ v_{ij} = 1\end{cases} \quad . \tag{15}$$

Where, N is the node set of the targeted network G. And $Replace(x_{ij}, N)$ is a function that can replace the element x_{ij} with a random node in the node set N and it can guarantee that there is no repeated node in X_i after the replacement at the same time.

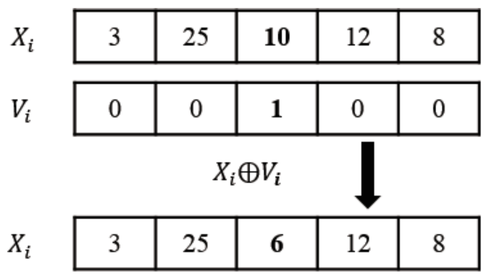


Fig. 2. Positional substitution rule

For example, the substitution operation between the position vector and the velocity vector can be described in Fig. 2, where $V_i = (0, 0, 1, 0, 0)$, so the elements xi_1, xi_2, xi_4, xi_5 of the position vector X_i do not need to be replaced, while the elements x_{i3} are randomly replaced.

3.7 Frame the Proposed Algorithm

Algorithm 1 gives the whole framework of the proposed neighborhood-based particle swarm optimization to solve the maximizing influence.

Algorithm 1. NCDPSO for influence maximization

Input: Graph $G = (V, E)$, the number of iterations g_{max}, the size of particle swarm N, the inertia weight w, the learn factors c_1, c_2 and c_3, the size of the seed set k.

Step 1: Initialization:

Step 1.1: Initialize iterator $g = 0$;

Step 1.2: Initialize position vector: X ← Initialization (G, k, N);

Step 1.3: Initialize *Pbest* vector: *Pbest* ← Initialization (G, k, N);

Step 1.4: Initialize velocity vector: $V \leftarrow 0$;

Step 2: Select out the initial global best position vector *Gbest** according to the *LIE* value of each x_i;

Step 3: Begin cycling

Step 3.1: groups evolution: *Gbest'* ← Neighborhood search (*Gbest**);

10. **Step 3.2:** Update the velocity vector *V* according (12);
11. **Step 3.3:** Update the position vector *X* according (14);
12. **Step 3.4:** Compare and update the ***Gbest****: ***Gbest**** ← max (*Gbest**, *Gbest'*);
13. **Step 3.5:** Update the *Pbest* of the current generation;
14. **Step 4:** Stop criteria: if $g = g_{max}$, stop the algorithm, otherwise, let $g \leftarrow g + 1$ and go to **Step 3.1**;
15. **Output:** Output the ***Gbest**** as the seed set ***S***.

The function Neighborhood search (*Gbest**) (Neighborhood search, as in section 3.4); and the function max (*Gbest**, *Gbest'*) is the particle that selects the larger LIE value of the influence function.

3.8 Algorithm Complexity Analysis

The degree-based initialization method in step 1 requires $O(N \cdot k)$ basic operations. Step 3.1 needs $O(d \cdot k)$ basic operations, where d is the network average. Step 3.2 requires $O(k \cdot logk \cdot N)$ basic operations. Step 3.3 and 3.5 need $O(N \cdot k)$ basic operations, and step 3.4 requires $O(1)$ basic operations. Therefore, the worst case time complexity is $O(k \cdot logk \cdot N) + 4O(N \cdot k) + O(d \cdot k) + O(1)$. In addition, the time complexity of the influence spread expectation LIE is $O(d^2 \cdot k)$. Assuming that the operation time of the other steps need a unit, according to the rules of the symbol O, in the worst case, the complexity of the proposed NCDPSO is $O(k^2 \cdot logk \cdot N \cdot d^2 \cdot g_{max})$.

4 Experiments

The proposed NCDPSO algorithm is compared with other algorithms in terms of running time and influence spread in four real networks. The purpose of the experiment is to illustrate the difference of efficiency and effectiveness between our proposed algorithm and other algorithms.

4.1 Experimental Environment and Data Set

All the program code is written in Python, and the running computer is configured as Intel (R) Core (TM) i5-4590 CPU 3.30 GHZ, 8 GB of memory.

Table 1 shows the information of four real datasets used in the experiment.Where, Karate club network http://www-personal.umich.edu/mejn/netdata/. Viki and Facebook from the http://snap.stanford.edu/data/, Viki datasets are the network of who votes for who in Wikipedia. The data set NetHEPT comes from http://www.arXiv.org, all of which are the cooperative relationship between the authors in the paper. NetHEPT contains data from the High Energy Physics-Theory part. The nodes all represent authors, and the edges represent the paper collaboration relationship between authors. The active probability of the IC model is 0.01 and 0.05 respectively.

Table 1. Statistics for datasets

Number	Data set	Node|V|	Edge|E|	Mean degree<k>
1	Karate	34	78	4.558
2	Facebook	4039	88234	43.69
3	Viki_Vote	7115	103687	26.6
4	NetHEPT	15235	31399	4.12

4.2 Comparison Algorithm and Experimental Parameter Setting

We compare our NCDPSO algorithms with four widely used comparison algorithms. The list of four comparison algorithms in our experiment is as follows.

(1) CELF [23]: The algorithm uses the sub-model of the marginal revenue function to dynamically update the marginal revenue of network nodes by maintaining a priority queue, and adopts the “Lazy-Forward” strategy to select the node with the largest marginal revenue to add to the seed set. And it has the same performance as greedy algorithm in the range of influence.

(2) DPSO [3]: The algorithms all use intelligent optimization algorithms to solve the problem of maximizing influence, and they are all very advanced algorithms in terms of time efficiency.

(3) Random [24]: Random is a baseline algorithm randomly selecting a group of k nodes as a seed set.

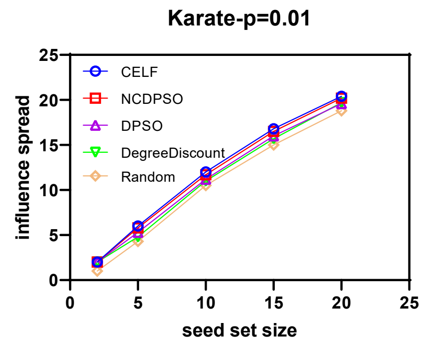
(4) Degree Discount [25]. The algorithm based on the idea that nodes with large discount-degree have large influence spread. When a node is selected as a seed node, the predecessor nodes of this node will lose a part of influence. This is because the influence of these predecessor nodes is cut off by the selected nodes. Therefore, this algorithm outperforms other degree-based algorithms in terms of accuracy.

In the CELF algorithm, Monte Carlo simulation needs to be set to 10,000 times, but in this setting, the ordinary computer needs to run dozens of hours to get 50 seeds. In order to improve the running speed of the algorithm, the simulation times of Monte Carlo are set to 100 in this experiment at the expense of the influence spread of the loss algorithm. In order to approximate the average influence spread range of the seed node set selected by other algorithms, the corresponding simulation times are all set to 1000 times. For Karate network, since there are only 34 nodes in the network, the maximum k value is set to 20; for other networks, the corresponding maximum k value is 50. DPSO and NCDPSO algorithm, parameters c_1, c_2, c_3, w, iteration number and population size is set to: 2,2,2,0.8,100,100, respectively. All the algorithms except CELF are run 30 times to average the results.

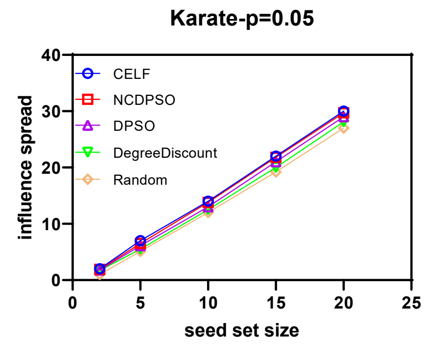
4.3 Experimental Results and Analysis

4.3.1 Influence Spread

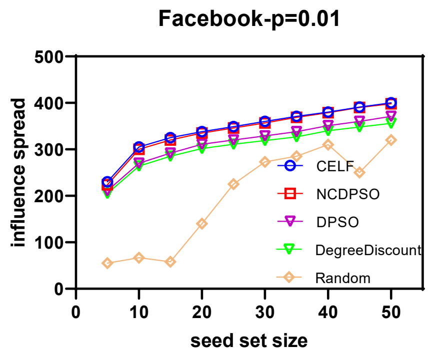
The influence spread refers to the number of active nodes in the network after the information transmission process. The larger the influence spread of seed set, the better the effect of the algorithm. Fig. 3 shows the influence spread of the algorithms on Karate, Facebook, Vivk-vote, and NetHEPT over four datasets when the influence probabilities are 0.01 and 0.05, respectively. The abscissa represents the size of the seed set, and the ordinate represents the influence spread of the seed set.



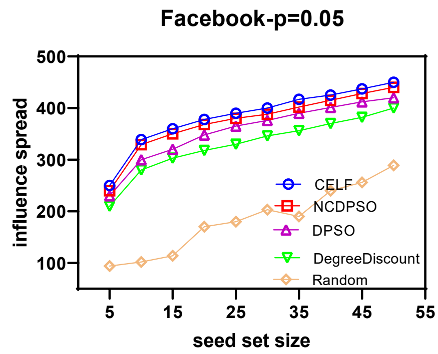
(a)



(b)



(c)



(d)

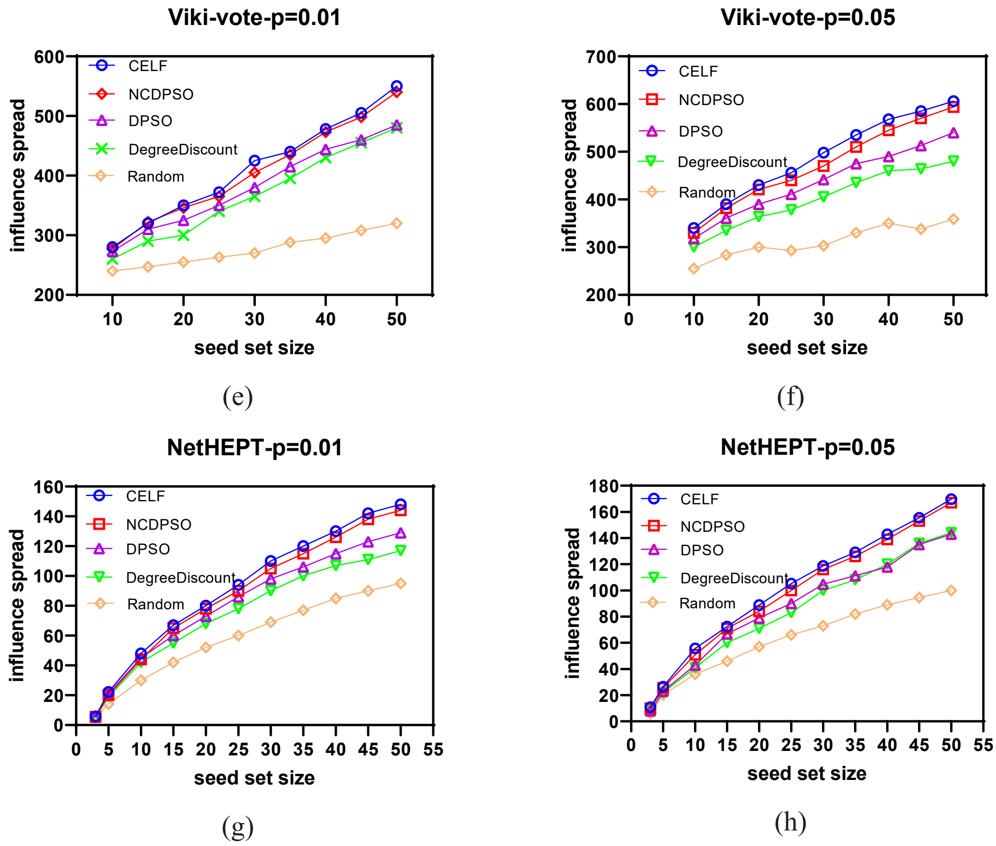


Fig. 3. IC model for the propagation range of 5 algorithms over 4 networks

According to the experimental results in Fig. 3, the performance of NCDPSO and CELF are very similar under two active probabilities in four data sets, and they have better performance than other contrast algorithms. In addition, NCDPSO outperforms DPSO, Degree Discount, and Random. Focusing on DPSO, we can find that as the number of nodes in Fig. 3(g) increases, the search ability of DPSO becomes worse because the search space increases with the size of the seed set. In contrast, the NCDPSO with the local search strategy following the bee theory has more stable search performance. When the seed set k=30, the propagation accuracy of NCDPSO was better than that of DPSO, Degree Discount and Random algorithm on Facebook (p=0.01), 7.84% 、10.64% and 23.53% respectively. On Vivk-vote (p=0.01) 8.65%, 12.26% and 30.10% respectively. On the NeTHEPT dataset (p=0.05), 6.67%, 14.29% and 34.29% respectively. Obviously, the performance of the NCDPSO algorithm benefits from the increase of particle's neighborhood search method, through this strategy, the algorithm can search for nodes in the network with greater marginal revenue. The Random algorithm randomly selects nodes without considering its influence spread, so the performance has always been the worst.

Fig. 3(a) and Fig. 3(b) shows that all five algorithms perform equally well in small-scale networks with fewer nodes. As shown in Fig. 3(a), Fig. 3(c), and Fig. 3(e), the influence spread of NCDPSO is similar to that of CELF, and the influence of nodes is mainly concentrated in its neighborhood under the condition of low probability of propagation, it also shows the efficiency of NCDPSO in solving the problem of maximizing influence. The Fig. 3(c) and Fig. 3(d) diagram in Fig. 3 not only shows that the Random algorithm is the worst performing, but also shows instability. In Fig. 3(e) to Fig. 3(h), when k<10, almost all the other four algorithms, except Random algorithm, have the same expectation of influence spread and show efficient performance. When k>10, the NCDPSO outperforms DPSO, Degree Discount, and Random. In addition, DPSO and Degree Discount have different performance for different data sets under different probabilities. For example, DPSO outperforms Degree Discount in Fig. 3(f) and Fig. 3(g), and when k>40 in Fig. 3(e) and Fig. 3(h), Degree Discount performs just as well as DPSO. Overall, our proposed NCDPSO algorithm is able to closely match the CELF algorithm and outperform other algorithms.

4.3.2 Running Time Analysis of Algorithms

The active probability is p=0.01 when different algorithms select k=30 (Karate, k=20) seeds from four data sets. The experiment results are shown in Fig. 4. The abscissa gives different data sets, the ordinate represents the running time of the algorithm.

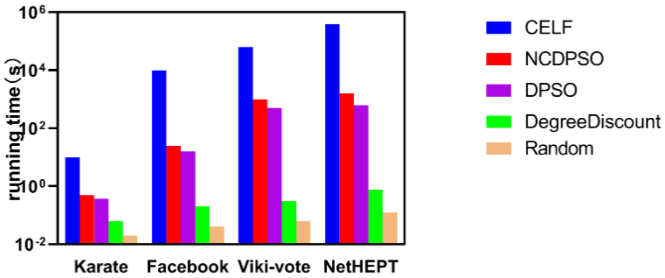


Fig. 4. Comparison of running times of algorithms

As can be seen in Fig. 4, the CELF is the most computationally expensive, since the CELF requires a sufficient number of simulations of the IC model in order to obtain an accurate average sphere of influence. The time complexity of the simulation process is linear with the network size, resulting in lower simulation efficiency on networks with more nodes and edges. As the network size increases, the running time of CELF increases significantly. In these four datasets, for large-scale networks like Viki-vote and NetHEPT, CELF takes tens of thousands of seconds to complete the selection of 30 seed nodes. In comparison, the running time of NCDPSO and DPSO is relatively stable, and DPSO is slightly better than NCDPSO. Compared with DPSO, NCDPSO needs more running time because the influence of three-hop spread and elite clone operation should be considered when calculating the influence of nodes. In addition, the running time of NCDPSO and DPSO in each network is nearly two orders of magnitude faster than CELF. The proposed NCDPSO algorithm is only faster than CELF algorithm in running time, because the proposed fitness function improves the accuracy at the cost of efficiency. In addition, the heuristic Degree Discount is much faster than the NCDPSO for each diffusion model in all networks, because the heuristic only considers the degree of the node to select the seed node. Random combination algorithm produces seed node set. Although the time efficiency is the best, the performance is the worst in the range of spread influence.

5 Summary

Influence maximization is still an unsolved problem in the fields of social network analysis and virus marketing, so it is necessary to study the algorithm with lower time complexity. In order to maximize the influence, this paper proposes a discrete particle swarm optimization to optimize the neighborhood search. In order to improve the propagation range of influence, we propose a new influence measure method, which considers the influence propagation of all neighbor particles within three hops of a node. We carefully analyze the sensitivity of a particle's neighbor to its influence, and introduce a "neighbor cognition" learning strategy to enable the particle to explore a larger search space, so as to improve the global search ability of the algorithm to avoid premature convergence. In order to improve the ability of local search, the strategy of elite cloning is used to expand the scope of searching for the better solution. The experimental results and the comparison with other algorithms verify the optimization ability of the proposed algorithm, although the simulation results show that the NCDPSO algorithm does not exceed the CELF algorithm in the propagation range, however, compared with the original DPSO algorithm, the improved algorithm achieves better LIE values in different scale social networks.

In this work, we have considered the degree information for influence estimation only. Therefore, this work can be extended for the influence maximization in dynamic structured large networks such as networks with various node attributes and edge attributes. With the continuous expansion of the network scale, the problem of maximizing influence is still a huge challenge. Heuristic algorithms have higher practicability and flexibility in actual production and life, and it can efficiently solve problems with characteristics of different fields. Therefore, heuristic algorithm to solve the problem of social network influence is still a research hotspot in the future. In addition, the influence of network topology on nodes is a popular research method at present, but the influence of a single node not only depends on the positional relationship of the node in the network, but also depends on the node's own attributes in the social network, such as the node's interest preference, active factor index, etc. Therefore, it is also a future research direction to consider comprehensively analyzing the influence of nodes from multiple aspects.

Acknowledgements

This work is partially supported by the Research on HMS Scheduling Optimization and Control and Intelligence System in Manufacturing IoT Environment the National Natural Science Foundation of China (62063021) and National Natural Science Foundation of China (62162040) Research on Terrain Representation and Dissemination Adaptive Scheduling Strategy for Large-scale Social Network Influence Adaptability. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Y. Liu, L. Sun, F. Xiong, J. Cheng, A Maximizing Influence of Multiple Nodes Propagation Algorithm Based on Optimal Neighbor Discovery, Journal of Computers 32(4)(2021) 187-200.
[2] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proc. ICNN95-international Conference on Neural Networks, 1995.
[3] M. Gong, J. Yan, S. Bo, L. Ma, Q. Cai, Influence maximization in social networks based on discrete particle swarm optimization, Information Sciences 367-368(2016) 600-614.
[4] Q. Wang, M. Gong, S. Chao, S. Wang, Discrete particle swarm optimization based influence maximization in complex networks, in: Proc. 2017 IEEE Congress on Evolutionary Computation (CEC), 2017.
[5] J. Yang, J. Liu, Influence Maximization-Cost Minimization in Social Networks Based on a Multiobjective Discrete Particle Swarm Optimization Algorithm, IEEE Access 6(2017) 2320-2329.
[6] X. Zhou, F. Ji, L. Wang, Y. Ma, H. Fujita, Particle swarm optimization for trust relationship based social network group decision making under a probabilistic linguistic environment, Knowledge-Based Systems 200(9)(2020) 105999.
[7] J. Tang, R. Zhang, P. Wang, Z. Zhao, L. Fan, X. Liu, A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks, Knowledge-Based Systems 187(2020) 104833.1-104833.12
[8] S.S. Singh, A. Kumar, K. Singh, B. Biswas, LAPSO-IM: A learning-based influence maximization approach for social networks, Applied Soft Computing 82(2019) 105554.
[9] L. Cui, H. Hu, S. Yu, Q. Yan, Z. Ming, Z. Wen, N. Lu, DDSE: a novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks, Journal of Network & Computer Applications 103(2018) 119-130.
[10]Q. Jiang, G. Song, C. Gao, Y. Wang, W. Si, K. Xie, Simulated annealing based influence maximization in social networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2011.
[11]C.P. Sankar, K.S. Kumar, Learning from bees: An approach for influence maximization on viral campaigns, PloS one 11(12)(2016) e0168125.
[12]A. Simsek, R. Kara, Using Swarm Intelligence Algorithms to Detect Influential Individuals for Influence Maximization in Social Networks, Expert Systems with Applications 114(2018) 224-236.
[13]L. Qiu, X. Tian, J. Zhang, C. Gu, S. Sai, LIDDE: A differential evolution algorithm based on local-influence-descending search strategy for influence maximization in social networks, Journal of Network and Computer Applications 178(2021) 102973.
[14]J.-R. Lee, C.-W. Chung, A fast approximation for influence maximization in large social networks, in: Proceedings of the 23rd international conference on World Wide Web, 2014.
[15]L. Qiu, X. Tian, S. Sai, C. Gu, LGIM: A global selection algorithm based on local influence for influence maximization in social networks, IEEE Access 8(2019) 4318-4328.
[16]Y. Liu, M. Tang, J. Yue, J. Gong, Identify influential spreaders in complex real-world networks, in: Proc. 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015.
[17]J.H. Fowler, N.A. Christakis, Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study, BMJ 337(2008) a2338.
[18]S. Pei, L. Muchnik, J.S. Andrade Jr, Z. Zheng, H.A. Makse, Searching for superspreaders of information in real-world social media, Scientific reports 4(1)(2014) 1-12.
[19]W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestri, M. Viale, A.M. Walczak, Statistical mechanics for natural flocks of birds, Proceedings of the National Academy of Sciences 109(13)(2012) 4786-4791.
[20]H. Wu, S. Qian, Y. Liu, G. Xu, B. Guo, Elite clone local search differential evolution algorithm for multi-objective dynamic environment economic scheduling, Journal of Shandong University (Engineering Science) 51(1)(2021) 11-23.
[21]J. Xiao, Y. Liu, J. Zhou, Quantum Clone Elite Genetic Algorithm-Based Evaluation Mechanism for Maximizing Network Efficiency in Soil Moisture Wireless Sensor Networks, Journal of Sensors (2021) 5590472.

[22]D. Karaboga, B. Gorkemli, A combinatorial Artificial Bee Colony algorithm for traveling salesman problem, in: Proc. International Symposium on Innovations in Intelligent Systems and Applications, 2011.
[23]J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007.
[24]D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003.
[25]W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009.