Journal of Computer Science

Original Research Paper

Personalized Recommendation System: Web of Things Using Modular Density-Based Community Discovery

[1]Vinod Patidar, [2]Jayashree A. Patil, [3]Sonika Thapak, [3]Manmohan Singh,
[4]Vikas Prasad, [5]Shaheen Ayyub and [6]Dharmendra Sharma

[1]Department of Computer Science and Engineering, Parul Institute of Technology, Parul University, Vadodara, India
[2]Department of Information Technology, Rajarambapu Institute of Technology, Islampur, Maharashtra, India
[3]Department of Computer Science and Engineering, IES College of Technology, Bhopal, India
[4]NICMAR Business School, NICMAR University Pune, India
[5]Department of Computer Science and Engineering, Technocrats Institute of Technology, Bhopal, India
[6]Department of Computer Science and Engineering, School of Technology Management and Engineering,
NMIMS University, Indore, India

Article history
Received: 21-10-2023
Revised: 13-04-2024
Accepted: 15-04-2024

Corresponding Author:
Manmohan Singh
Department of Computer Science and Engineering, IES College of Technology, Bhopal, India
Email: kumar.manmohan4@gmail.com

Abstract: Community discovery is the cornerstone and core of study in customized recommendation, assembling group features and social network analysis on the web of things. Conventional community discovery methods, however, struggle with difficulties including low accuracy, delayed convergence, modularity resolution limits and more when dealing with more complicated social networks. Because of this, differential evolution and module density are included in community discovery and a better differential evolution and module density community discovery approach is offered. The method first alters the mutation strategy and differential evolution parameters and then uses the module density as a fitness function to get beyond the restriction of modularity resolution improve population quality overall and hasten the process of global convergence. Experiments with various commonly used community discovery techniques using computer-generated network datasets and sample real-world network datasets. When Collective Co-Evolutionary Differential Evolution-based Community Detection (CCDECD) and Classification-based Differential Evolution algorithm for Modularity Optimization (CDEMO) are used simultaneously, the difference is optimal and the Q value increases by 3.3% compared to the Overlapping Community Detection algorithm based on Density Peaks (OCDDP) and 4.6% compared to GN. However, the mutual information value NMI is not optimal since the GN method is better suited for small-scale networks. The division result of Improved Differential Evolution and modularity density Community Detection (IMDECD) on the network is the closest to the actual network since the NMI is ideal, the Q value is low and the standard deviation is minimal. The suggested method has improved accuracy and better convergence performance, according to the experimental findings.

Keywords: Modularity-Resolution, Modular Density, Personalized Recommendation, Web of Things, Classification-Based Differential Evolution Algorithm for Modularity Optimization

Science Publications

Introduction

A social network is a social structure composed of nodes and edges. Nodes represent individuals and edges represent various social relationships between individuals (such as friendship, partnership, etc.,) specific forms of social networking such as social networks, scientific research cooperation networks, etc., Community Discovery is Discovery Cohesion (CDDC), a subset of the set so that nodes within the set are more connected than they are to nodes outside the set. As a result, nodes are more connected (Rebehy and Chung, 2013). Based on the detected community structure, the topology and function of complex networks can be deeply revealed. It has been widely used in fields such as personalized recommendation of structured data (Trier and Bobrik, 2009; Liu *et al*., 2021). A subset of the set called community discovery is called discovery cohesion and it makes the nodes of the set more related to one another than it does to nodes outside the set. The architecture and purpose of complex networks can be fully revealed based on the community structure that has been observed. The community has just discovered the current algorithm in the semantic community of the information network, combining many sources. Community discovery algorithms can be grouped into four categories: Agglomerative and The algorithm of cutting (Kolli and Narayanaswamy, 2019; Liu *et al*., 2019; Bettencourt, 2014) the algorithm based on label propagation (Hima and Singh, 2015; DeOrio and Bertacco, 2010; Cai *et al*., 2019), the optimization based on modularity Algorithms (Cai *et al*., 2019; Gliwa *et al*., 2012; Evans, 2020; Qiao *et al*., 2018), algorithms based on evolutionary algorithms (Gliwa *et al*., 2012; Evans, 2020; Qiao *et al*., 2018) in classic GN proposed by literature (Rebehy and Chung, 2013) The algorithm is an agglomerative algorithm that continuously removes the largest The edge betweenness of can accurately obtain the community with the hierarchical structure, but the calculation speed of the algorithm needs to be improved; Newman proposes a proposed hierarchical clustering method based on the maximization of network modularity (Liu *et al*., 2019). The algorithm goes in the direction that maximizes and decreases the modularity each time. To merge the trends, the accuracy of community discovery can be improved to a certain extent. Literature (Gliwa *et al*., 2012) a community detection combined with differential evolution is a proposed Algorithm (collective co-evolutionary differential evolution-based community detection-CCDECD) introduced in the algorithm. The co-evolution framework combined with differential expansion to optimize network modularity to find the optimal community network; literature (Evans, 2020). Modularity functions the local monotonicity of the number, combined with the genetic algorithm, which proposes a fast and effective algorithm. It is a partial search operator that can be used to solve large-scale community discovery problems. Although these algorithms have been widely used, there is still room for improvement in accuracy and convergence speed. In addition, most of the algorithms are based on modularity optimization algorithms (Cai *et al*., 2019; Gliwa *et al*., 2012; Evans, 2020; Qiao *et al*., 2018); there is a standard modularity resolution limit (Bachi *et al*., 2012); that is, some smaller communities will be merged in the calculation process in calculation errors.

To overcome the modularity resolution limitation and improve the accuracy of community discovery accuracy and convergence performance, this study uses differential evolution with a simple structure, easy to implement, fast convergence, strong robustness and the introduction of module encryption degree (Han *et al*., 2018) as a fitness function to address the modularity resolution limitation, a community development method combining improved differential evolution and module density is proposed. Improved differential evolution and modularity density community detection-IMDECD). From the standpoint of the kind of clever devices to monitor air quality for mobile providers phones or automobiles. As an illustration, MAQA (Cosenzo and Szymanski, 2013; Hawryszkiewycz, 2009; Ling-Han *et al*., 2013; Chatterjee and Saha, 2019; Singh *et al*., 2022; 2023; Almakki *et al*., 2022; Shi *et al*., 2018; Chen *et al*., 2017) has provided a gateway consisting of a Wi-Fi-enabled smartphone and a wearable CO_2 detector utilizes Bluetooth to read data from a wearable sensor (Hawryszkiewycz, 2009). Using a mobile phone as a gateway to combating both air pollution and using a smartphone's Bluetooth and Wi-Fi network. The process starts with first, the differential evolution was improved and a new mutation strategy was designed. and dynamic adaptive parameter tuning for the F and CR variables; then as a fitness function of differential evolution, module density is individual evaluations; finally modified based on known community structure; includes correction operations based on community variables and initialization and binomial modification of the crossover stage. Through comparative experiments, the proposed calculation in this study is a verified method compared to the agglomerative algorithm (GN), a density peak-based algorithm Overlapping Community Detection algorithm based on Density Peaks (OCDDP), a combination of Differential Evolution and modularity method (CCDECD) and Classification-based Differential Evolution algorithm for Modularity Optimization (CDEMO), The accuracy and convergence performance of community discovery have been improved to a certain extent. A social network is a network of connections made up of nodes and edges. Edges are certain forms of social networking such as social networks and scientific research cooperation networks, while nodes are individuals and reflect various social ties between persons (such as friendship and partnership). The foundation and center of research in personalized recommendation, putting together group features and social network analysis on the web of things is community discovery. However, when dealing with more complex social networks, conventional community-finding algorithms face challenges like as low accuracy, delayed convergence, modularity Resolution constraints and more.

In this study, the enhanced method Image Enhancement Method (IMDE), which is proposed, performs noticeably better than the other five mutation strategies and achieves the best result on 80% of the test functions. The community discovery algorithm's enhancement technique is studied in this research using module density and the previously improved differential evolution. Each node in the network is first represented by the community identification during the initialization stage, after which the fitness function and module density are set.

Differential Evolution

Its powerful global optimization capability is superior to its algorithm. Differential evolution (Ling-Han *et al.*, 2013) has been used because of its simplicity and effectiveness. It is widely used in data mining, pattern recognition and other fields. It integrates it into social Zone discovery with the following advantages: Neither complex binary encoding is required nor does it need to use a probability density function to adapt its individuals (Chatterjee and Saha, 2019), let alone any prior knowledge of community structure. The differential evolution package consists of three main steps: Mutation, crossover and selection.

Mutation Strategy

The mutation strategies used in differential evolution mainly include the following four species: DE/rand/1, DE/rand/2, DE/best/1 and DE/rand-to-best/1. As shown in formula (1):

$$V_{i}^{g} = x_{p1}^{g} + F\,x\left(x_{P2}^{g} - x_{P3}^{g}\right) \tag{1}$$

$$V_{i}^{g} = x_{p1}^{g} + F\,x\left(x_{P2}^{g} - x_{P3}^{g}\right) + F\,x\left(x_{P4}^{g} - x_{P5}^{g}\right) \tag{2}$$

$$V_{i}^{g} = x_{best}^{g} + F\,x\left(x_{P2}^{g} - x_{P3}^{g}\right) \tag{3}$$

$$V_{i}^{g} = x_{best}^{g} + F\,x\left(x_{best}^{g} - x_{P1}^{g}\right) + F\,x\left(x_{P2}^{g} - x_{P3}^{g}\right) \tag{4}$$

where, $i \in \{1, 2, \cdots, NP\}$, p_1, p_2 and p_3 are drawn from 1, 2, ⋯, NP; Randomly selected and satisfying the condition $p_1 \neq p_2 \neq p_3 \neq i$, g is the number of iterations number, is the target individual, is the variant individual and NP is the population size.

Crossover Operation

Commonly used crossover methods in differential evolution include binomial crossover and exponential crossover; cross the binomial crossover scheme performs each of the n components row crossover; the exponential crossover scheme selects a segment of variant individuals and the segment starts with a random integer k of arbitrary length that can be to include multiple components.

Since binomial crossover is easier to implement and has lower time complexity, in this study, we choose a binomial crossover pair for the variant individual and the target individual performs a crossover operation to generate the experimental individual. Binomial crossover (5) display:

$$u_{i,j}^{g} = \begin{cases} v_{i,j}^{g}, if\ \ Rand\ £\ CR \\ x_{i,j}^{g}, otherwise \end{cases} \tag{5}$$

where, $i \in \{1, 2, \cdots, NP\}$, $j \in \{1, 2, \cdots, n\}$, rand is between 0 and 1The uniformly distributed random number of are the i^{-th} target. The j^{-th} dimension components of individuals, variant individuals and experimental individuals.

Select Action

The fitness function value of the individual experimental and the individual target. For comparison, the fitness function value is more significant for the maximization problem. Finally, the individuals enter the next-generation population, as shown in formula (6):

$$x_{i}^{g+1} = \begin{cases} u_{i,j}^{g}, if\ \ f\left(u_{l}^{g}\right) 3 \left(x_{l}^{g}\right) \\ x_{i,j}^{g}, otherwise \end{cases} \tag{6}$$

Improvement of Differential Evolution

To improve the accuracy and convergence performance of community discovery, this study improves based on existing differential evolution, the main improvement measures. Including mutation strategy improvement and dynamic adaptive parameter adjustment.

Improvement of mutation strategy

Equations (1-2), the basis vector of Eqs. (1-2). The quantity is randomly selected from the population, which has a strong global search ability and is not easy to trap.

The advantage of entering the local optimum but the convergence speed is slow; the Eqs. (3-4). The base vector is the optimal individual in the current population, so the local search ability is strong. The convergence speed is fast, but it is easy to fall into the local optimum.

The above four mutation strategies all have the following problems: Performance in terms of accuracy or convergence. For example, formula (1) and formula (2) Have high accuracy, but slow convergence and Eqs. (3-4) are fast but easy to fall into optimal local solution, resulting in low accuracy. Therefore, this study will construct a mixed mutation strategy to balance accuracy and convergence speed, as shown in formula (7):

$$v_i^g = \begin{cases} x_{P1}^g + F\ x\left(x_{P2}^g - x_{P3}^g\right), if\ \ f_i > \overline{f} \\ x_{best}^g + F\ x\left(x_{best}^g - x_{P1}^g\right) + F\ x\left(x_{P2}^g - x_{P3}^g\right) otherwise \end{cases} \tag{7}$$

In formula (7), "DE/rand/1" is used to maintain the accuracy of the population; and randomly generate new individuals to prevent the people from falling into local optima. Then use "DE/rand-to-best/1" to use the best individual in the current population. The body's information generates new individuals to speed up the convergence rate. Where f_i is the fitness function value for the target individual, if Its fitness function value is greater than the average of all individuals in the current population If the fitness value is, it is classified as an excellent individual and DE/rand/1 is used to prevent stop falling into a local optimum; otherwise, it is classified as an eliminated individual and the DE/rand-to-best/1 uses the best individual to generate new individuals.

Dynamic Adaptive Parameter Adjustment

There are two essential parameters in differential evolution: The scaling factor F-value and the Crossover probability CR. After experimental observation and data research, it has been shown that the selection of parameter values should be adaptively adjusted (Chatterjee and Saha, 2019; Singh *et al*., 2022). An international optimization algorithm is differential evolution. It belongs to the category of evolutionary algorithms and is related to others like the genetic algorithm. It was created expressly to operate on vectors of real-valued values rather than bitstrings, in contrast to the genetic algorithm. An individual population is processed by an evolutionary algorithm called DE using dimensional vectors of real integers. A mutant is produced using a differential mutation operator in each iteration, for each parent.

The scaling factor *F* value in Eq. (7) is usually a constant, which plays the role of controlling the scaling range of the difference variable. When the *F* value is small, the population difference is reduced, the global search ability is reduced and local convergence is likely to be caused; when the *F* value is significant, although it is easy to jump out of the optimal local solution, the convergence speed will be slowed down. Therefore, Eq. (8) is used to adjust the value of *F* (Singh *et al*., 2022) dynamically:

$$F = F_{min} + \frac{F_{max} - F_{min}}{2} x\left[\left(2 - exp\left(\frac{g}{g_{max}} xln2\right)\right) + \frac{f_l - f_{min}}{f_{max} - f_{min}}\right] \tag{8}$$

where, *g* is the number of iterations, is the maximum number of iterations, = 0.3, = 0.9.

In formula (5), *CR* is the crossover probability. The larger the *CR*, the faster the convergence speed, but it is easy to mature prematurely and fall into the local optimum. Therefore, the dynamic adaptive parameter adjustment (Ling-Han *et al*., 2013; Chatterjee and Saha, 2019; Singh *et al*., 2022; 2023; Almakki *et al*., 2022; Shi *et al*., 2018; Chen *et al*., 2017; Fan *et al*., 2019; Xin *et al*., 2019; Cheng *et al*., 2018; 2016; Swietojanski *et al*., 2016) of the *CR* value is carried out by using Eq. CR with the best convergence, with 0.1 and 0.9 L:

$$\begin{gathered} CR = CR_{min} + \frac{CR_{max} - CR_{min}}{2} \\ x\left[\left(2 - exp\left(\frac{g}{g_{max}} xln2\right)\right) + \frac{f_{max} - f_l}{f_{max} - f_{min}}\right] \end{gathered} \tag{9}$$

Through dynamic adaptive parameter adjustment, the mutation rate of the population is adaptively determined so that the population maintains diversity in the initial stage and avoids premature convergence. Gradually reducing the mutation rate in the later stage avoids destroying the optimal solution and increases the search for the optimal global solution possibility of the optimal solution. Meta-heuristic algorithms must produce a varied variety of solutions employing diversification or exploration approaches in order to explore the search space globally. Based on past knowledge or fresh information discovered throughout the search process that a current good answer is found in this location, an intensification or exploitation strategy can direct the individual to search in a particular region (Chen *et al*., 2017; Fan *et al*., 2019; Xin *et al*., 2019; Cheng *et al*., 2018; 2016; Swietojanski *et al*., 2016; He *et al*., 2018; 2019; Zhang *et al*., 2020; Wang *et al*., 2019).

Experimental Verification

This study tests five commonly used standard benchmark functions to verify the effectiveness of the above improvement measures for differential evolution. Table 1 shows the five test functions' names, formulas, search spaces and optimal values. For example, peak functions, F3~F5 are multi-peak functions.

The improved strategies this study are combined, named Improved Differential Evolution (IMDE) and compared with the mutation strategies of Eqs. (1-4) (Singh *et al*., 2022; 2023; Almakki *et al*., 2022) in the mutation strategy DE_version 1 for comparison, in which the F value of each mutation strategy and the CR value in the crossover operation is calculated by Eqs. (8-9). The experimental results are shown in Table 2 and the results include the mean and standard deviation of 30 independent running tests (data in parentheses in Table 2 are standard deviations). In the experiments, the same experimental parameters as the mutation strategy DE version 1 are chosen: Population size NP = 100, dimension size D = 30, F ∈ [0.3, 0.9] and CR ∈ [0.1, 0.9]. The dynamic mutation method solves problems by automatically applying a number of mutation operators. To create the following generation, it employs more than one mutation operator. The SM-GA is more flexible and has a wider variety of constitutive properties. It also outperforms traditional genetic algorithms in terms of computing speed and convergence stability. These debates support and advance current intelligent optimization theory and techniques and they have a wide range of potential applications in the management of production, the optimization of complex systems and other areas.

Table 2 compares the six mutation strategies in accuracy and convergence performance. The average value reflects the accuracy and the standard deviation reflects the convergence performance. The optimal solution in each case is shown in bold. In this study, the proposed improved strategy IMDE significantly outperforms the other five mutation strategies and obtains the optimal solution on 80% of the test functions.

Table 1: Benchmark functions (Singh *et al*., 2023)

Function	Name	Formula	Search space	Optimal solution
F1	Sphere	$F_1(x) = \sum_{i=1}^{D} x_i^2$	[-100,100]	0
F2	Schwefel	$F_2(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	[-100,100]	0
F3	Rosenbrock	$F_2(x) = \sum_{i=1}^{D} 100\left(x_1^2 - x_{i+1}\right)^2 + \left(x_1 - 1\right)^2$	[-100,100]	0
F4	Rastrigin	$F_4(x) = \sum_{i=1}^{D} \left(x_i^2 - 10cos\left(2\pi x_i\right) + 10\right)^2$	[-5.12,5.12]	0
F5	Ackley	$F_5(x) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) -$ $\left(-0.2\sum_{i=1}^{D} cos\left(2\pi x_i\right)\right) + 20 + e$	[-32,32]	0

Table 2: Performance comparison of different mutation strategies

				De/rand		
Function	De/rand/1	De/rand/2	De/best/1	-to-best/1	De_version1	De_version1
F1	6.480e-37	3.689 4e-14	2.154e-105	4.543e-52	4.543e-52	1.300 7e-302
	(2.116e-37)	(1.367e-13)	(1.080e-103)	(1.237e-51)	(1.237e-51)	(0.000e+00)
F2	6.660e-37	6.749e-01	6.225e+00	1.027e-04	1.027e-04	1.696e-105
	(2.116e-37)	(1.97e-01)	(4.59e-01)	(3.26e-04)	(3.26e-04)	(6.159e-105)
F3	4.646e-01	5.680e-01	1.140e-15	5.198e-01	5.198e-01	3.214e-18
	(3.570e-00)	(9.040e-00)	(3.767e-12)	(3.120e-00)	(3.120e-00)	(1.394e-16)
F4	2.659e+01	7.319e+01	3.487e+01	9.714e+01	9.714e+01	7.242 6e-04
	(6.14e-00)	(8.84e-00)	(5.77e-00)	(3.83e-00)	(3.83e-00)	(3.22e-00)
F5	2.679e+01	-5.878e-06	-7.594e-06	-3.467e-06	-3.467e-06	-4.818e-06
	(6.14e-00)	(3.66e-06)	(1.44e-06)	(2.19e-06)	(2.19e-06)	(3.412e-06)

It can be seen from Table 2 that the standard deviation of IMDE and Eq. (1) DE/rand/1 are compared separately and the standard deviations of the four functions F1~F4 are significantly improved, which verifies that IMDE adopts "DE/rand-to-best/1” can speed up the convergence speed; compared with Eq. (4) DE/rand-to-best/1 separately, the accuracy of the solution on the four functions F1~F4 has been significantly improved, which verifies that IMDE adopts “DE/rand-to-best/1”. /rand/1” can prevent the population from falling into local extremes and improve the ability of the differential evolution algorithm to solve the global extremum.

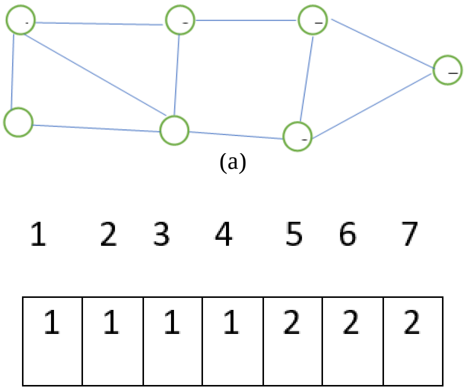
Initialization

For a complex network *G* = (*V*, *E*) with n nodes, the k^{-th} individual in the population consists of Eq. (10):

$$x_K = \{c_1, c_2, \ldots\ldots, c_n\} \qquad (10)$$

Among them, *ci* represents the community to which node *i* belongs, the community identifier. When IMDECD is initialized, the community to which each node belongs is randomly assigned, so there are at most *n* communities in *G*. The maximum value of the community identifier is *n*.

For example, Fig. 1(a) shows a network with seven nodes. According to the definition of community structure, the network is divided into two communities represented by nodes of different colors. Figure 1(b) is a vector representation based on community identifiers. Along with high resolution, vectors or feature vectors (sequences of integers) in a vector space (metric space) also offer multidimensional comparability. This is crucial because it enables high-resolution searching inside a huge data set. Additionally, by describing instances as vectors, it is possible to calculate their similarity automatically in terms of distance rather than having to expensively record this knowledge through symbolic representations.


(a)

1	2	3	4	5	6	7
1	1	1	1	2	2	2

(b)

Fig. 1: Vectorization representation of individuals

Fitness Function Settings

The fitness function usually has two types: Modularity and module density. The modularity *Q* function is a milestone in community discovery algorithm research history. It is widely used in community detection because of its ease of implementation. The algorithm based on modularity optimization has become a community discovery algorithm. A field of study in algorithms.

However, *Q* has several drawbacks: First, maximizing the *Q* value turns out to be an NP problem. Second, large *Q*-values do not always make sense and there are cases where random networks without community structure can have large *Q*-values. Finally, *Q* has a resolution limit and maximizing *Q* cannot find a community with a smaller community size. The expression of the *Q* value is shown in Eq. (11):

$$Q = \sum_{i=1}^{m}\left[\frac{L(V_l,V_f)}{L(V,V)} - \left(\frac{L(V_1,V)}{L(V,V)}\right)^2\right] \qquad (11)$$

To overcome the resolution limitation of modularity and improve the algorithm's accuracy, literature (Han *et al.*, 2018). Proposed a module density calculation method after a series of theoretical derivations and practical tests, as shown in Eq. (12). Furthermore, the experimental verification by literature. Bachi *et al.* (2012) confirm the effectiveness of the module density in solving the resolution limitation problem. Therefore, the module density is chosen as the fitness function in this study:

$$D_\lambda = \sum_{i=1}^{m}\left[\frac{2\lambda(V_i,V_i) - 2(1-\lambda)L(V_i\bar{V}_i)}{|V_i|}\right] \qquad (12)$$

Among them, *m* is the number of communities after the division is completed, *λ* is a number between 0 and 1, is the sum of degrees between nodes in the sub-network and is the number within. The sum of the degrees of the node and other nodes except the number of nodes.

Modification Based on Community Structure

Since there are differences in the function solution and community discovery of differential evolution, this study uses the known community structure to make the following modifications to improve the accuracy of the calculation when combining and improving differential evolution and module density.

Community Variable CV (*i*)

In the iterative process of community discovery, some nodes may be put into the wrong community and these errors will weaken the algorithm's ability to find the optimal solution and reduce the accuracy.

To solve the above problems, this study adopts the correction operation based on the community variable *CV* (*i*) proposed by literature (Shi *et al*., 2018) to reduce the situation that the nodes are assigned incorrectly. *CV* (*i*) is defined as the ratio of the number of node *i* and its neighbor nodes not in the same community to the degree of node *i*, as shown in Eq. (13), where *deg(i)* is the degree of the i^{-th} node, *ci* is the community containing the i^{-th} node:

$$CV(i)\frac{\sum(j)c_i{}^1c_j}{deg(i)} \tag{13}$$

The correction operation process is as follows: First, some nodes are randomly selected. Then, its community variable is calculated for each node and compared with a threshold, a predefined constant obtained after repeated experiments, which is taken as 0.3 in this experiment. If the community variable of the node is more significant than this threshold, the node and all its neighbors will be put into the same community and the new community will be the community that contains the most significant number of nodes among the neighbor nodes. Otherwise, the node will do nothing.

Through the correction operation based on community variables, every time a node is assigned, its neighbor nodes are considered, reducing node assignment errors and improving accuracy.

Initialization Stage (*IS*)

During initialization, each node is randomly assigned to a community; however, this may result in some unconnected nodes being given to the same community. To reduce the amount of computation, the initialization phase is modified.

The modified initialization process is as follows: Once an individual is generated, nodes in the individual are randomly selected, their community identifiers are assigned to their neighbor nodes, an initial population is generated and each node and its neighbor nodes are in the same community in the initialization phase. Through this operation, the space of possible solutions is limited, unnecessary iterations are eliminated and the convergence speed of the IMDECD algorithm is improved.

Binomial Crossover Stage (*BCS*)

Since each assignment of the community identifier is random, the following situation may arise: For two individuals {1,1,1,2,3} and {3,3,3, 2,1}, node *i* is assigned multiple times assigned to the same community, but the corresponding community identifiers are different each time, that is, there is no one-to-one correspondence between communities and community identifiers. If the community identifier counts which community node *i* is assigned to, an erroneous result will occur, resulting in node *i* being given to the wrong community. In this case, if you want to get the correct statistical results, the algorithm must judge whether different identifiers correspond to the same community according to the neighbor nodes and community structure, which will decrease the efficiency of searching for the optimal solution. Based on the above considerations, the following modifications are made to the crossover operation according to Chatterjee and Saha (2019).

Modification of the binomial crossover procedure: First, set the experimental individual for each $i \in \{1,2, \cdots NP\}$. Then, for each $j \in \{1,2, \cdots, n\}$ consider the j^{-th} component in and the variant individual. If rand ≤ *CR*, find all nodes whose community identifiers are *vi*, *j* and then assign the community identifiers of the corresponding nodes in as, to implement the correspondence; otherwise, no operation will be performed on.

The modified binomial cross enables each community to have a unique identifier corresponding to it, so there is no situation where nodes are assigned to the same community but correspond to different identifiers. Therefore, the statistical function can be completed only through the community identifier, thus improving the algorithm's efficiency in searching for the optimal solution.

Algorithm Description

Combined with the community structure, the specific execution steps of the IMDECD algorithm are as follows:

1. Initialize the relevant parameters NP, n and of the IMDECD algorithm; see Chapter 5 experimental parameter settings for details
2. According to the modified initialization process, NP n-dimensional vectors are generated to form an initial population = {, ⋯,}, where each is represented by Eq. (10). Calculate the module density value of each individual and save the current optimal individual and its corresponding module density value
3. If (*g* <) Then

1. The adaptive mixed mutation strategy is composed of Eqs. (7-8) performs mutation operation on each individual in the population to generate the corresponding mutant individual; performs the correction operation in Eq. (13) to correct in offset vector
2. The adaptive crossover strategy is composed of Eqs. (5 and 9) performs the modified binomial crossover operation on each individual in the population to generate the corresponding experimental individual; performs the correction operation in Eq. (13) and corrects the offset vector
3. Using the greedy selection strategy of formula (6), the target individuals in the population are compared with their corresponding experimental individuals respectively and the better individuals among them are selected to form the next generation population. The fitness function in Eq. (6) uses the module density given in Eq. (12) $g = g + 1$; At the end of the algorithm operation, the output of the complex network community structure is the optimal division result and the corresponding module density value

Time Complexity Analysis

Assume that the number of nodes in the community network is n, the number of edges is *t*, the number of communities after division is m, the population size is NP and the number of iterations is *g*. For a community network with n nodes, the time complexity of the module density function in this study is O (× NP × m). The time complexity of the mixed mutation strategy of Eq. (7) and the greedy selection strategy of Eq. (6) are both and the time complexity of the adaptive crossover strategy of formula (5) is O (× NP × n), so the time complexity of the algorithm in this study is O (× NP × (n + m), the time complexity comparison with other algorithms is shown in Table 3.

Table 3: Comparisons of time complexity

Algorithm	Time complexity
Gn	$V_l^g = x_{p1}^g + F\ x\left(x_{p_2}^g - x_{p_3}^g\right)$
Ocddp	$V_l^g = x_{p1}^g + F\ x\left(x_{p_2}^g - x_{p_3}^g\right)$ $+F\ x\left(x_{p_4}^g - x_{p_5}^g\right)$
Ccdecd	$V_l^g = x_{best}^g + F\ x\left(x_{p_2}^g - x_{p_3}^g\right)$
Cdemo	$V_l^g = x_{best}^g + F\ x\left(x_{best}^g - x_{P1}^g\right)$ $+F\ x\left(x_{P2}^g - x_{P3}^g\right)$
Imdecd	$CV(i) = \frac{\sum\left[j\left(ci\ ^1\ cj\right)\right]}{deg(i)}$

Materials and Methods

This study takes advantage of the above-improved differential evolution, combined with module density, to study the improvement strategy of the community discovery algorithm. First, in the initialization stage, each node in the network is represented by the community identifier; then, the fitness function is set and the module density is used to replace the module for the resolution limitation problem of the module degree as a fitness function; finally, using the known community structure, conduct modification operations, including those based on community variables and initial modification of initialization and binomial crossover stages.

Results

The experimental hardware environment in this study is Intel® Pentium, CPU 3.0 GHz, memory 8.0 GB and the simulation software is Windows 10 system and MATLAB R2017a. In this study, computer-generated networks and five real-world networks of different scales are selected to conduct experiments and the proposed algorithm is compared with GN (Rebehy and Chung, 2013), OCDDP (Singh *et al*., 2022), CCDECD (Gliwa *et al*., 2012) and CDEMO (Ling-Han *et al*., 2013) algorithms. The algorithm was independently run 30 times on each dataset to reduce statistical error. The mean value was selected for comparison and all algorithms' experimental parameter settings and environment configurations were the same.

Set the parameters according to the size of the dataset. In the Karate network, set the population size NP to 20 and the maximum number of iterations to 50; in the football network, set the NP to 30 and the maximum number of iterations to 300; in the jazz and US air-lines networks, set the NP to is 30 and the maximum number of iterations is 600; in the Polblogs network, NP is set to 60 and the maximum number of iterations is 600.

In this study, the modularity Q value shown in Eq. (11) and the mutual information value *NMI* shown in Eq. (14) (Ling-Han *et al*., 2013; Chatterjee and Saha, 2019; Singh *et al*., 2022; 2023; Almakki *et al*., 2022) are selected as the evaluation function:

$$NBI\left(A,B\right) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_A} c_{ij}\, lb\left(\frac{c_{ij^N}}{c_i c_j}\right)}{\sum_{i=1}^{c_A} c_i\, lb\left(\frac{c_i}{N}\right) + \sum_{j=1}^{c_B} c_j\, lb\left(\frac{c_j}{N}\right)} \tag{14}$$

Computer Generated Network

This study uses the artificial computer-generated network literature (Singh *et al.*, 2023) proposed by Lanc Chinetti to verify the performance of the IMDECD algorithm to detect network community structure. There are 128 nodes in the GN benchmark network and they are divided into 4 communities with 32 nodes in each community.

The mixing parameter μ in the above network is mainly used to determine the shared edge between any node in a community and the nodes of other communities. The mixing parameter μ in the above network is mainly used to determine the relationship between any node in a community and other communities. Edges are shared between nodes.

Figure 2 shows the change of *NMI* between the IMDECD algorithm and other comparison algorithms on the GN benchmark network as the value of μ increases.

As can be seen from Fig. 2, compared with the other four algorithms, with the continuous increase of μ, the performance advantage of the IMDECD algorithm becomes more and more apparent. When the mixing parameter μ is equal to 0.7, the decline begins to appear and the results verify the effectiveness of the IMDECD algorithm.

Real World Networks

This section uses five representative real network datasets of different sizes for testing. The details of each dataset are shown in Table 5, including the Karate Club network (Almakki *et al.*, 2022) (Karate), American college students' 2000 season American football league network (Rebehy and Chung, 2013) (football), jazz musicians cooperative network (Shi *et al.*, 2018) (jazz), American airlines network (Chen *et al.*, 2017) (US Airlines) and the 2004 US election political blog network (Fan *et al.*, 2019) (Polb-logs). Table 4 illustrates the scale of the dataset: The number of nodes and edges.

Regarding the module density, λ takes the value λ ∈ (0,1). To analyze the influence of λ on the community discovery results of the algorithm in this study, we take the influence of the modularity *Q* value and the mutual information value NMI of the community discovery results under different values of λ. Figure 3 shows the Karate network's community discovery results as a function of the λ value.

As can be seen from Fig. 3, in the Karate network, when λ = 0.3, NMI = 1, Q = 0.3728, the result of the community division is the same as the real network community, but the Q value is lower; when λ = At 0.6, 0.7 and 0.8, the Q values are all 0.419 8, but the NMI values are 0.735 3, 0.6388 and 0.655, respectively. Therefore, in the Karate network, when λ = 0.6, NMI = 0.735 3, Q = 0.4178, both are higher values.

Figure 4 shows that the community discovery results of the Football network vary with the value of the λ result.

Figure 4, in the football network, when λ = 0.4~0.8, the Q value does not change much, all-around 0.6056 and the NMI value changes greatly. When λ = 0.6, the NMI value is the largest at 0.9354. Therefore, in the Football network, when λ = 0.6, NMI = 0.9354 Q = 0.6066, both are higher values.

Since Karate and football networks have known community structures, the results of community division can be measured by the modularity Q value and mutual information value NMI; while Jazz, US airlines and Polblogs networks have no known community structure; therefore, it can only be measured by the modularity Q value. Figure 5 shows the community discovery results of the other three networks as a function of the λ value. It can be seen that when λ = 0.6, the Q value is higher, so the λ in the module density, takes the value of 0.6.

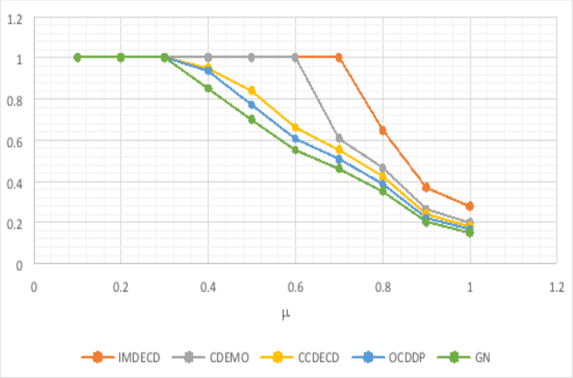


Fig. 2: Average NMI of IMDECD and other algorithms

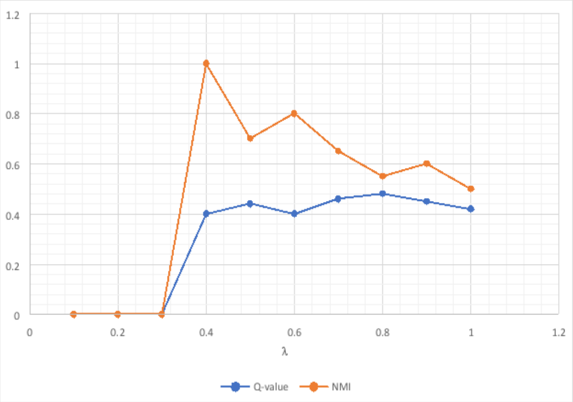


Fig. 3: Effect of λ on Q and NMI in Karate network

Table 4: Average NMI of IMDECD and other algorithms

Mutual Information NMI	IMDECD	CDEMO	CCDECD	OCDDP	GN
0.1	1.0000	1.0000	1.000	1.000	1.00
0.2	1.0000	1.0000	1.000	1.000	1.00
0.3	1.0000	1.0000	1.000	1.000	1.00
0.4	1.0000	1.0000	0.095	0.935	0.85
0.5	1.0000	1.0000	0.084	0.077	0.07
0.6	1.0000	1.0000	0.066	0.605	0.55
0.7	1.0000	0.6072	0.552	0.506	0.46
0.8	0.6468	0.0462	0.042	0.385	0.35
0.9	0.3696	0.0264	0.024	0.022	0.02
1.0	0.2772	0.0198	0.018	0.165	0.15

Table 5: Real number dataset

Data set	Node	Number of sides
Karate	35	75
Football	116	615
Jazz	199	2745
US Airlines	335	2125
Polblogs	1498	19095

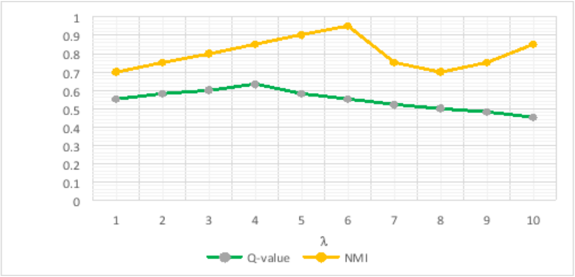


Fig. 4: Effect of λ on Q and NMI in football networks

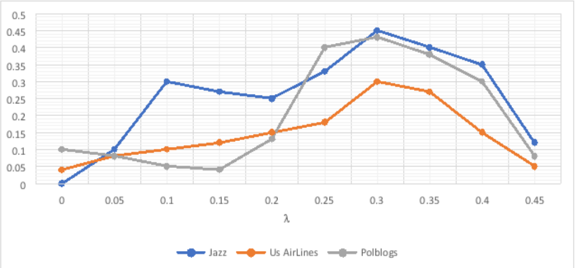


Fig. 5: Effect of λ on Q in other networks

Table 6: Effect of λ on Q and NMI in the Karate network

Modularity Q-value and Mutual Information NMI	Q-value	NMI
0.1	0.00	0.00
0.2	0.00	0.00
0.3	0.00	0.00
0.4	0.04	0.01
0.5	0.44	0.07
0.6	0.04	0.08
0.7	0.46	0.65
0.8	0.48	0.55
0.9	0.45	0.06
1.0	0.42	0.05

Table 7: Effect of λ on Q and NMI in football networks

Modularity Q-value and mutual information NMI	Q-value	NMI
0.45	0.55	0.07
0.50	0.58	0.75
0.55	0.06	0.08
0.60	0.63	0.85
0.65	0.58	0.09
0.70	0.55	0.95
0.75	0.52	0.75
0.80	0.05	0.07
0.85	0.48	0.75
0.90	0.45	0.85

Table 8: Effect of λ on Q in other networks

Modularity Q-value	Jazz	Us air-lines	Polblogs
0.00	0.00	0.04	0.01
0.05	0.01	0.08	0.08
0.01	0.03	0.01	0.05
0.15	0.27	0.12	0.04
0.02	0.25	0.15	0.13
0.25	0.33	0.18	0.04
0.03	0.45	0.03	0.43
0.35	0.04	0.27	0.38
0.04	0.35	0.15	0.03
0.45	0.12	0.05	0.08

Table 6 we have to mention value parameter in the NMI is karate network and the Q value and standard deviation are low, which indicates that the division result of IMDECD on the karate network than table Football network and Table 7 is the closest to the football network Effect of λ on Q and NMI in football networks. However, the lower value also shows that the IMDECD algorithm than Table 8 Effect of λ on Q in other networks division result of IMDECD on the network.

Analysis of Results

According to the above research, Tables 9-10 are the experimental results after each algorithm is independently run 30 times on 5 data sets; the table includes the average value of modularity Q, the value of mutual information NMI the standard deviation and the number of communities. Description of the data format in the table: 0.419 8/4, where 0.419 8 represents the degree of modularity and 4 represents the number of communities; (0.00E-00) represents the standard deviation of 30 runs.

Table 9: Comparison of NMI and modularity between Karate and football network

Data set	Algorithm	Modularity Q	Mutual information NMI
Karate	IMDECD	0.429 8/4 (0.00E-00)	0.725 4
	CDEMO	0.429 8/4 (0.00E-00)	0.698 3
	CCDECD	0.429 8/4 (0.00E-00)	0.687 1
	OCDDP	0.416 3/4 (2.61E-04)	0.673 3
	GN	0.407 3/2	0.840 1
Football	IMDECD	0.614 0/10 (2.8E-04)	0.935 5
	CDEMO	0.614 6/10 (0.00E-00)	0.918 7
	CCDECD	0.614 6/10 (8.90E-04)	0.917 2
	OCDDP	0.610 4/12 (1.86E-03)	0.910 4
	GN	0.607 3/12	0.930 1

Table 10: Comparison of modularity of other networks

Algorithm	Jazz	US Airlines	Polblogs
IMDECD	0.443 7/7 (3.03E-03)	0.306 0/16 (3.67E-03)	0.306 0/16 (3.67E-03)
CDEMO	0.443 7/5 (9.57E-03)	0.293 3/11 (2.63E-03)	0.293 3/11 (2.63E-03)
CCDECD	0.439 2/4 (6.32E-03)	0.299 1/11 (5.97E-03)	0.299 1/11 (5.97E-03)
OCDDP	0.384 8/21 (1.24E-02)	0.176 4/12 (8.45E-02)	0.176 4/12 (8.45E-02)
Proposed	0.405 0/39	0.140 0/124	0.140 0/124

As can be seen from Tables 9-10, compared to other algorithms, IMDECD has promising results on five real network datasets. Table 9, the Q-values and criteria of IMDECD in the Karate network. The difference is optimal with CCDECD and CDEMO at the same time and the Q value is increased by 3.3% compared with OCDDP and 4.6% compared with GN; the mutual information value NMI is not optimal because the GN algorithm is more suitable for small-scale networks; In the Football network, the NMI is optimal and the Q value and standard deviation are low, which indicates that the division result of IMDECD on the Football network is the closest to the actual network. However, the lower value also shows that the IMDECD algorithm is unsuitable for smaller networks.

Table 10, the Q value and standard deviation of IMDECD in the three networks of Jazz, US Air-lines and Polblogs are the best, which shows that the IMDECD algorithm proposed in this study is more suitable for more extensive networks. Therefore, compared with the other four algorithms, the algorithm proposed in this study has higher accuracy and better convergence performance in community discovery.

Regarding the resolution limit, among the four comparison algorithms, the CCDECD and CDEMO algorithms are based on modularity optimization. In the Football, Jazz, US Air-Lines and Polblogs networks, the number of communities divided by the IMDECD algorithm is more significant than that of the CCDECD and CDEMO algorithms, indicating that the IMDECD algorithm using the module density can divide more communities; that is, it can distinguish smaller Community.

Discussion

According to the above research, Tables 9-10 are the experimental results after each algorithm is independently run 30 times on 5 data sets; the table includes the average value of modularity Q, the value of mutual information NMI the standard deviation and the number of communities. Description of the data format in the table: 0.419 8/4, where 0.419 8 represents the degree of modularity and 4 represents the number of communities; (0.00E-00) represents the standard deviation of 30 runs.

Conclusion

To overcome the limitation of modularity resolution and improve community discovery's accuracy and convergence performance, this study proposes a community discovery algorithm that combines improved differential evolution and module density. By Using the algorithm first adjusts the differential evolution mutation strategy and dynamically adapts the F and CR variables. Then second given the resolution limitation of modularity, the module density is used as the fitness function of differential evolution to evaluate the individuals in the population; finally, it is modified based on the known community structure, including the correction operation based on community variables and corrections in the initialization and binomial crossover stages. Comparative experiments show that the IMDECD algorithm proposed in this study has higher community discovery accuracy, better convergence performance, more vital optimization ability and robustness and can effectively detect community structures in real-world networks.

Acknowledgment

To our families and all those who matter in our lives.

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Vinod Patidar: Designed the research plan supervised this study and approved the final manuscript.

Jayashree A. Patil: Correction of the translation of the manuscript in English, experimental monitoring, member of the laboratory/implementation and approved the field data.

Sonika Thapak: Member of the laboratory preparation of the nursery, implementation monitoring and collection of data.

Manmohan Singh and Vikas Prasad: Carrying out the experiment, collected and verified the analyzed data; prepared the draft of the manuscript and approved the final manuscript.

Shaheen Ayyub and Dharmendra Sharma: Designed the research plan and supervised this study and approved the final manuscript.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

Reference

Almakki, R., AlSuwaidan, L., Khan, S., Baig, A. R., Baseer, S., & Singh, M. (2022). Fault Tolerance Byzantine Algorithm for Lower Overhead Blockchain. *Security and Communication Networks*, *2023*(1), 1–9. https://doi.org/10.1155/2022/1855238

Bachi, G., Coscia, M., Monreale, A., & Giannotti, F. (2012). Classifying Trust/Distrust Relationships in Online Social Networks. *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, 552–557. https://doi.org/10.1109/socialcom-passat.2012.115

Bettencourt, L. M. A. (2014). Impact of Changing Technology on the Evolution of Complex Informational Networks. *Proceedings of the IEEE*, *102*(12), 1878-1891. https://doi.org/10.1109/jproc.2014.2367132

Cai, C., Li, L., Zeng, D., & Ma, H. (2019). Exploring Writing Pattern with Pop Culture Ingredients for Social User Modeling. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1-8. https://doi.org/10.1109/ijcnn.2019.8852187

Chatterjee, B., & Saha, H. N. (2019). Detection of Communities in Large Scale Networks. *IEEE Xplore*, 1051-1060. https://doi.org/10.1109/iemcon.2019.8936304

Chen, J., Zhang, H., He, X., Nie, L., Liu, W., & Chua, T.-S. (2017). Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 335-344. https://doi.org/10.1145/3077136.3080797

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H. anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). Wide & Deep Learning for Recommender Systems. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 7-10. https://doi.org/10.1145/2988450.2988454

Cheng, Z., Ding, Y., Zhu, L., & Kankanhalli, M. (2018). Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. *Proceedings of the 2018 World Wide Web Conference*, 639-648. https://doi.org/10.1145/3178876.3186145

Cosenzo, K. A., & Szymanski, B. K. (2013). Challenges of social cognitive network science: Network science collaborative technology alliance perspective. *2013 IEEE 2nd Network Science Workshop (NSW)*, 122–125. https://doi.org/10.1109/nsw.2013.6609207

DeOrio, A., & Bertacco, V. (2010). Electronic design automation for social networks. *DAC '10: Proceedings of the 47th Design Automation Conference*, 621-622. https://doi.org/10.1145/1837274.1837429

Evans, J. (2020). Social Computing Unhinged. *Journal of Social Computing*, *1*(1), 1-13. https://doi.org/10.23919/JSC.2020.0002

Fan, W., Ma, Y., Yin, D., Wang, J., Tang, J., & Li, Q. (2019). Deep social collaborative filtering. *Proceedings of the 13th ACM Conference on Recommender Systems*, 305-313. https://doi.org/10.1145/3298689.3347011

Gliwa, B., Zygmunt, A., & Byrski, A. (2012). Graphical analysis of social group dynamics. *IEEE Xplore*, 41-46. https://doi.org/10.1109/cason.2012.6412375

Han, X., Liu, Y., & Wang, J. (2018). Modeling and analyzing privacy-awareness social behavior network. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 7-12. https://doi.org/10.1109/infcomw.2018.8406830

Hawryszkiewycz, I. (2009). Workspace requirements for complex adaptive systems. *2009 International Symposium on Collaborative Technologies and Systems*, 342–347. https://doi.org/10.1109/cts.2009.5067499

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 639-648. https://doi.org/10.1145/3397271.3401063

He, X., He, Z., Du, X., & Chua, T.-S. (2018). Adversarial Personalized Ranking for Recommendation. *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 355-364. https://doi.org/10.1145/3209978.3209981

Hima, K. S., & Singh, A. K. (2015). Allocation and sharing infrastructure resources via social networks using preference based on trust method. *IEEE Xplore*, 1-4. https://doi.org/10.1109/iciiecs.2015.7193045

Kolli, N., & Narayanaswamy, B. (2019). Influence Maximization from Cascade Information Traces in Complex Networks in the Absence of Network Structure. *IEEE Transactions on Computational Social Systems*, *6*(6), 1147-1155. https://doi.org/10.1109/tcss.2019.2939841

Ling-Han, L., Jia-Yin, Q., & Chao-Yang, Y. (2013). How does praise and rumor spread online? Analyzing user relationships on social networks: A case study of Haidilao. *2013 International Conference on Management Science and Engineering 20th Annual Conference Proceedings*, 57–65. https://doi.org/10.1109/icmse.2013.6586262

Liu, L., Wang, L., Wu, W., Jia, H., & Zhang, Y. (2019). A Novel Hybrid-Jump-Based Sampling Method for Complex Social Networks. *IEEE Transactions on Computational Social Systems*, *6*(2), 241-249. https://doi.org/10.1109/tcss.2019.2893889

Liu, S., Hu, X., Wang, S.-H., Zhang, Y.-D., Fang, X., & Jiang, C. (2021). Mixing Patterns in Social Trust Networks: A Social Identity Theory Perspective. *IEEE Transactions on Computational Social Systems*, *8*(5), 1249-1261. https://doi.org/10.1109/tcss.2020.3021179

Qiao, S., Han, N., Gao, Y., Li, R.-H., Huang, J., Guo, J., Gutierrez, L. A., & Wu, X. (2018). A Fast Parallel Community Discovery Model on Complex Networks Through Approximate Optimization. *IEEE Transactions on Knowledge and Data Engineering*, *30*(9), 1638-1651. https://doi.org/10.1109/tkde.2018.2803818

Rebehy, S., & Chung, K. S. K. (2013). Measuring performance & interrelatedness in social networks of knowledge workers. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 910-915. https://doi.org/10.1145/2492517.2500330

Shi, S., Zhang, M., Liu, Y., & Ma, S. (2018). Attention-based Adaptive Model to Unify Warm and Cold Starts Recommendation. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 127-136. https://doi.org/10.1145/3269206.3271710

Singh, M., Mewada, H., Tahilyani, M., Malviya, J., Sharma, R., & Shrivastava, S. S. (2022). RRDTool: A Round Robin Database for Network Monitoring. *Journal of Computer Science*, *18*(8), 770-776. https://doi.org/10.3844/jcssp.2022.770.776

Singh, M., Patidar, V., Ayyub, S., Soni, A., Vyas, M., Sharma, D., & Ranadive, A. (2023). An Analytical Survey of Difficulty Faced in an Online Lecture During COVID-19 Pandemic Using CRISP-DM. *Journal of Computer Science*, *19*(2), 242-250. https://doi.org/10.3844/jcssp.2023.242.250

Swietojanski, P., Li, J., & Renals, S. (2016). Learning Hidden Unit Contributions for Unsupervised Acoustic Model Adaptation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *24*(8), 1450-1463. https://doi.org/10.1109/taslp.2016.2560534

Trier, M., & Bobrik, A. (2009). Social Search: Exploring and Searching Social Architectures in Digital Networks. *IEEE Internet Computing*, *13*(2), 51-59. https://doi.org/10.1109/mic.2009.44

Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural Graph Collaborative Filtering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165-174. https://doi.org/10.1145/3331184.3331267

Xin, X., He, X., Zhang, Y., Zhang, Y., & Jose, J. (2019). Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 125-134. https://doi.org/10.1145/3331184.3331188

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, *52*(1), 1-38. https://doi.org/10.1145/3285029