International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

Detecting communities in social networks using reinforcement learning

[1]Bahareh Radi*- [2]Atiyeh Haghshenas
[1]University Professor .st_b_radi@azad.ac.ir
[2]Student .haghshenasatiyeh@gmail.com

Abstract
Community detection is a fundamental optimization challenge that investigates the identification of communities within graph-structured networks. Although numerous algorithms have been proposed for this problem, many of them are not scalable to large-scale networks and suffer from high computational costs. In this paper, we propose a multi-agent reinforcement learning (MARL) algorithm for community detection in complex networks, which demonstrates superior performance compared to several well-known baseline methods. The proposed approach is evaluated using multiple performance metrics, including majority accuracy and Nautical mile (NMI), and the results indicate strong and competitive performance. Interactive network-based methods are widely applied across various scientific domains, including social sciences and health informatics, where they facilitate the analysis of behaviors and structural patterns. Furthermore, community detection in dynamic networks can benefit from reinforcement learning and local optimization techniques to effectively manage evolving entities. This type of analysis provides a more efficient framework for examining continuously growing and evolving networks.

Keywords: Complex networks, Community detection, Multi-agent systems, Reinforcement learning, Majority accuracy

Overview
Recent studies have shown that various systems across different domains can be represented as complex networks. These networks are typically sparse at the global scale while being dense at the local scale. Social networks and physical connections among routers are examples of such complex structures. Community detection in these networks is of significant importance, and a variety of optimization-based methods have been proposed to address this problem. A novel approach based on multi-agent reinforcement learning for identifying communities in complex networks has recently been introduced, demonstrating strong capability in community detection. Experimental results on different networks show that this approach is able to detect communities with high accuracy and stability, and it is competitive with existing methods. This paper covers a review of related work, an introduction to multi-agent reinforcement learning, a detailed description of the proposed method, experimental evaluations, and concluding remarks.

Introduction
Since ancient times, human civilization has sought to discover new and unknown materials, such as metals and alloys, all of which can play a key role in improving overall quality of life. Since the Bronze Age, alloys have traditionally been produced according to a “base element” paradigm, in which a primary element is combined with various additional elements to enhance selected properties. In recent decades, a new approach to alloy design has been introduced that involves mixing typically five or more elements to produce compositionally balanced alloys known as high-entropy alloys (HEAs). Due to their attractive properties—such as high thermal and electrical conductivity, excellent corrosion resistance, and high strength combined with good ductility—HEAs have been extensively investigated. The aim of the present study is to introduce a new model for a lightweight H-beam interaction network constructed based on descriptors of lightweight H-beams. This model forms a network of interactions based on similarity measures, quantifying similarities among lightweight H-beam descriptors. Communities are extracted from the interaction network such that each community contains similar lightweight H-beam compositions. Network analysis is a powerful tool that seeks to evaluate target networks from various perspectives, including node classification, link prediction, and community detection (CD). The challenges and opportunities associated with classical community detection methods—such as spectral clustering and statistical inference—are increasingly being

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

addressed by deep learning techniques with enhanced capacity to handle high-dimensional graph-structured data, achieving remarkable performance as reported in recent studies. In the context of community detection, communities are typically extracted from a given network by selecting a scoring function (e.g., modularity) that captures the intuition of communities as groups of densely connected nodes. Subsequently, a method is applied to identify sets of nodes that maximize the value of the chosen scoring function. This process can follow two main directions: agglomerative approaches, in which groups of nodes are merged, or divisive approaches, in which edges are removed from the network and the scoring function is recalculated. Within this challenging framework, the central hypothesis of this work is that reinforcement learning (RL) can serve as an effective approach for optimizing modularity in community detection solutions applied to dynamic social networks.

Literature Review

Lightweight H-beams composed of at least five elements with equal or near-equal atomic percentages exhibit high strength, primarily due to four distinct core effects, the understanding of which contributes to a deeper insight into the physical properties of these alloys. Recent studies have demonstrated that machine learning (ML) techniques can facilitate the design and prediction of material properties in lightweight H-beam systems. In addition, community detection and network analysis methods have been employed to investigate the complex characteristics of lightweight H-beam compositions. These approaches have shown significant potential in identifying structural patterns and key features, reflecting important advancements in this field.

Related Work

Community detection in complex networks has attracted significant attention from researchers. Community detection methods can generally be classified into five categories: conventional algorithms, hierarchical algorithms, majority-based algorithms, spectral algorithms, and dynamic algorithms. Among these, hierarchical algorithms are more widely used. In divisive methods, edges that connect different communities are identified and removed. In contrast, agglomerative methods employ a bottom-up strategy to identify communities by progressively merging nodes or groups of nodes. Several other algorithms have also been proposed for community detection, each with its own advantages and limitations. The use of majority-based concepts has produced favorable results in community detection by improving algorithmic efficiency. In addition, genetic algorithms and other artificial intelligence techniques have been applied for this purpose. However, the algorithms employed must carefully consider network topology and specific environmental conditions in order to produce accurate results. Recent studies have enhanced these approaches through the use of reinforcement learning and neural networks.

Preliminaries

This section presents some fundamental aspects related to community detection, along with key concepts and artificial intelligence techniques relevant to this field.

Community Detection Concepts

In a graph consisting of nodes and edges, communities are formed by nodes that are densely connected to each other while maintaining sparser connections with the rest of the graph. Identifying communities within a graph helps reveal its internal structure and provides a means to describe the entities that constitute these communities. In a dynamic network, the network (G) can be examined at each time point. Modularity is a widely used criterion for evaluating the quality of community detection in a network. Using a specific formulation, the degree of organization and the internal connectivity of communities within a graph can be quantified. The measure (Q_{ds}) has been introduced to extend this concept and to address the limitations of the traditional modularity measure (Q).

Proposed Solutions

1.

The first proposed solution employs a reinforcement learning–based approach to optimize both the algorithms and their parameters for modularity-based community detection in a dynamic network. This approach builds upon

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

existing modularity-based community detection methods, such as eigenvector-based techniques, random walk methods, label propagation, and multilevel algorithms. In the proposed reinforcement learning framework, the Q-learning algorithm is used to store combinations of algorithms and their corresponding parameters. The improvement in the network architecture score is utilized as a reward signal and incorporated into the Q-matrix. Within this framework, iterative reinforcement learning agents and their associated policies are implemented to enhance modularity-based community detection. The quality measure (Q(G, c)) is selected for comparison with modularity density. The discount factor (\gamma) plays a critical role in improving the accumulated reward, while an RL-greedy policy determines whether a selected action maximizes the improvement of the current community state (C).

2.

In the second approach, a community detection framework based on interactive networks of high-entropy alloys is proposed by integrating Louvain-based concepts with modified Particle Swarm Optimization (PSO) algorithms. In this method, lightweight H-beam community members are first selected based on chosen descriptors, and after preprocessing, communities are identified using the Louvain method and modified PSO techniques. The proposed framework consists of five main stages:

1. Preprocessing the dataset to enable the application of machine learning algorithms.
2. Computing content-based similarity among descriptors.

[1]

3. Constructing the interactive network of high-entropy alloys.
4. Calculating structural similarity between descriptors.
5. Extracting communities by optimizing the modularity objective function.

The dataset used in this study consists of 90 lightweight H-beam alloys, from which communities are identified using the listed algorithms. A specific descriptor is selected for the lightweight H-beams, after which the proposed algorithms are applied to detect communities. As an example, the overall workflow comprises three main stages: data preparation, construction of the lightweight H-beam interaction network, and application of machine learning algorithms for community extraction. Finally, modularity is employed as a quantitative measure to evaluate the quality of the detected communities.

Data Normalization

Normalization is applied when the input data values do not lie within a common range and have different scales, in order to prevent features or descriptors with large numerical values from dominating the overall system performance. In addition, normalization can reduce the effect of out-of-range scales and ensure that all inputs remain within a unified interval.

In this study, min–max normalization is used to map feature values into the range [0, 1] using Equation (2). In this formulation, minA and maxA represent the minimum and maximum values of the features in set A, respectively. The original and normalized feature values are denoted by v and v′. As indicated in Equation (1), the minimum and maximum normalized values are 0 and 1.

$$v' = \frac{v - \min_A}{\max_A - \min_A}$$

Content-Based Cosine Similarity Metrics

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

Content-based cosine similarity measures the angle between two vectors and determines whether the selected vectors are considered aligned. As shown in the dataset in Appendix A, each feature of a single composition can be analyzed and compared with other compositions. The content-based cosine similarity between two compositions is calculated using the sum of the products of corresponding features divided by the product of the square roots of the sums of squares of each feature vector. In this formula, **xi** represents the i-th feature of the first composition, and **yi** represents the i-th feature of the second composition.

[2]

$$\text{content cosine similarity } (x, y) = \frac{\sum_i (x_i y_i)}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}}$$

Jaccard Structural Similarity Metrics

The Jaccard index is primarily used to compare the structural similarity of a dataset. The Jaccard similarity coefficient between two datasets is typically calculated as the number of shared features divided by the total number of features present in both sets.

Since a graphical network representation of the interaction network is required to compute the Jaccard metric, the matrix obtained from content-based cosine similarity must first be examined with various thresholds to identify a suitable value. This ensures the creation of a proper network graph, allowing structural similarities to be measured based on the resulting visualization. In the present study, a threshold of **0.98** was selected to construct the network for analyzing content-based cosine similarity.

The calculation of the Jaccard structural similarity is shown in Equation 3.

[3]

$$\text{structural Jaccard similarity } (v_i, v_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

Jaccard Structural Similarity

In Equation 3, vi and vj represent two nodes corresponding to lightweight H-beam compositions. The term $|N_i \cap N_j|$ denotes the number of shared features between compositions vi and vj, while $|N_i \cup N_j|$ represents the total number of features in both vi and vj. It is important to note that this metric can be applied to all pairs of shared features across the dataset.

Alpha Coefficient (α)

The calculation of content-based and structural similarity parameters results in two separate similarity matrices. To identify communities, a combined similarity matrix is required as input, which incorporates both types of similarities. The alpha coefficient (α) determines the relative influence of each similarity measure. Specifically, α controls the contribution of content-based similarities as well as the structural Jaccard similarity. The output of this phase is a

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

combined similarity matrix, which serves as input for the community detection algorithm.

Community Detection

Each community in the interaction network represents a group of alloys that are interrelated and exhibit similar behaviors or properties. Identifying these communities allows the analysis of functional relationships within the network and the grouping of similar compositions into coherent clusters.

Theoretical Notation Definitions

A complex network can be represented as a graph $G(V, E)$, where $V$is the set of nodes and $E$is the set of edges. A network $C(v, e)$is considered a subnetwork if $v \subseteq V$and $e \subseteq E$.

Let Abe the adjacency matrix. Two nodes are considered adjacent if an edge exists between them. Specifically, if a link exists between node $i$and node j, then $A_{ij} = 1$; otherwise, $A_{ij} = 0$. A weighted network assigns a weight $w$to its edges, where $w$is a real number. Communities in networks are groups of nodes that are more densely connected to each other than to the rest of the network. Community detection is a key feature that can be used to extract valuable information from networks.

Louvain Algorithm

In scientific studies, content and their relationships are often represented as complex networks, in which the topology of the nodes is interconnected and organized either structurally or randomly. The Louvain algorithm is a metaheuristic method used to identify communities and groups within a graph. Each extracted community represents a group, and this algorithm is considered a **bottom-up clustering method**. To evaluate the quality of the detected communities, the modularity parameter is employed, and maximizing this parameter is of critical importance.

The Louvain algorithm is regarded as one of the fastest and most effective methods for community detection, aiming to achieve maximum modularity over time. The algorithm operates in two phases, which are iteratively repeated and include the following steps:

1. Assign each node to an initial community based on the current network structure.
2. Merge potential neighboring nodes and evaluate the modularity gain for this transfer.
3. Relocate nodes to communities based on the neighboring community that maximizes modularity.
4. Repeat this process for all nodes until a stable state is reached.

The Louvain algorithm operates efficiently, quickly computing modularity and minimizing the number of communities. The overall performance of the algorithm can be assessed using the modularity gain ΔQ, which is calculated based on node transfers from one community to another. The algorithm continuously applies iterative procedures to create and merge communities to achieve maximum network modularity. This iterative process is repeated several times to ensure all nodes are assigned to suitable communities, and the alternating phases continue until local modularity reaches its maximum and stable results are obtained.

Community Detection Based on Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was introduced by Kennedy and Eberhart in 1995, inspired by the collective behavior of birds. PSO is considered one of the most important swarm intelligence algorithms, often capable of providing near-optimal solutions. In this method, the movement of particles—represented as an array of nodes—is used to update each particle and detect communities within the network. The optimization process facilitates rapid convergence and reduces the reliance on the fitness function. During each update, the **pbest** (personal best) and **gbest** (global best) values are determined, and both the social interaction and learning of particles are considered. The algorithm also defines parameters such as t, w, c_1, c_2, rand_1, rand_2, and $\rho$to control particle movement and community detection.

[4.5.6]

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

$$V_i(t+1) = wV_i(t) + c_1 + rand_1[pbest_i - X_i(t)] + c_2 + rand_2[gbest - X_i(t)]$$

$$X_{ij}(t+1) = \begin{cases} 1, & \text{if } \rho < sig(V_{ij}(t+1)) \\ 0, & \text{otherwise} \end{cases}$$

$$sig(V_{ij}(t+1)) = 1 / [1 + \exp(V_{ij}(t+1))]$$

Optimized PSO Algorithm and Group Learning

Since communities are obtained independently of the ordering of lightweight H-beam compositions in the material interaction network $L(G)$, the resulting communities are optimized and smaller subsets of G. Identifying independent communities within a network requires discovering communities that are structurally independent. A linear graph corresponding to the developed Particle Swarm Optimization (PSO) algorithm, combined with group learning techniques—referred to as **LEPSO**—is employed to optimize the results obtained from linear graph partitioning.

Community Detection Using Optimized PSO

The linear graph for the chemical compositions of alloys is represented as $L(G) = \langle N, E \rangle$, where $N = \{n_1, n_2, \ldots, n_k\}$is a subset of $L(G)$and each node is described as $X_i = (X_{i1}, X_{i2}, \ldots, X_{id})$with $k = \mid N \mid$. If the initial value $X_{ij} = m$is assigned, the results may indicate a relationship between two compositions $e = \langle n_j, n_m \rangle$and the particles X_i, particularly when $n_j$and $n_m$belong to the same community in $L(G)$.

To determine the initial community as an optimal type, each particle in the PSO is considered as an array of alloy compositions. In this framework, the adjacency matrix of the initial interaction network is used to represent connections between materials through connected nodes. Some potential challenges of this approach include random particle initialization and repeated updates of particle positions. Moreover, this issue may lead to particles representing links that did not previously exist. To address these problems, particles are recommended to be represented within a list of typical neighbors.

The core idea of this approach is to use the distribution of neighbors for each node as a representative of an alloy composition, ensuring that newly introduced particles in the transfer or initialization process are valid. Additionally, the elimination of invalid particles and the prevention of locally suboptimal communities are achieved through iterative bipartitioning and automated community detection, representing key advantages of this PSO-based optimization method.

Particle Fitness Function in Optimized PSO

A well-defined community enables researchers to propose new and diverse quality indices for evaluating the potential benefits of a partition. The underlying assumption behind modularity is that the edge density of a cluster should be higher than the expected density of the subgraph, allowing nodes to be randomly connected.

To complete the discretization process of the proposed algorithm, each node and its relationship with other nodes are analyzed

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

individually. First, the connections between the initial composition and other compositions are obtained, followed by the construction of the adjacency matrix. Finally, the particle fitness function, which determines the quality of communities in the final phase, is defined. This function is also referred to as **modularity**, as shown in Equation 7.
[7]

$$fit(P_i) = Q(C) = \sum_{c=1}^{m}\left[\frac{l_c}{|E|} - \left(\frac{d_c}{2|E|}\right)^2\right]$$

Particle Fitness and Partition Parameters
In this equation, fit(Pi) represents the fitness value of particle P_i, and $m$is the number of communities found in the network partition $C$of graph $G = \langle N, E\rangle$. Here, $l_c$denotes the number of edges connecting nodes within community $c \in C$, $d_c$is the sum of degrees of nodes within community C, and $|E|$is the total number of edges in G.

Particle Velocity and Position Update

• Particle Velocity Update

An optimized particle velocity update algorithm, called GbestGenerator, is employed to avoid local optima. This method leverages a voting-based clustering strategy to fully utilize the valuable hidden community patterns found in less efficient particles and in the gbest values.

$$w_t = (w_{\max} - w_{\min})\frac{t_{\max} - t}{t_{\max}} + w_{\min}$$

If the gbest value does not improve over consecutive iterations T_{max}, indicating that the swarm is trapped in a local optimum, clusters of particles within the MPS (Most Promising Swarm) are created. This is achieved by selecting all gbest particles from the last T_{max}iterations, as well as the corresponding consecutive particles, to generate a suitable combination of particles for producing a new gbest.
Accordingly, each particle can potentially have both a minimum and a maximum velocity. Equation 4 illustrates this concept, and the inertia coefficient, denoted as w, is considered crucial in the execution of particle velocity updates. The strategy for adjusting $w$can be effectively expressed using Equation 8, as described below.

Inertia Coefficient Adjustment
In this context, $w_{max}$and w_{min}represent the initial and final inertia coefficients, respectively. T_{max}denotes the maximum number of iterations, and $t$indicates the current iteration. As shown in Equation 10, at the initial stage ($t = 0$), both $w_{max}$and $w_t$are considered equal. As $t$approaches T_{max}, $w_t$gradually decreases toward w_{min}.
Furthermore, due to the convergence behavior of the algorithm in the early stages, larger inertia coefficients are required to allow faster particle movement. In later stages, smaller coefficients are assigned to particles to gradually increase their overall stability.

Particle Position Update
Based on Equation 5, the components of the position vector were initially assigned values of 0 or 1, which is not ideal for representing particles relative to their neighbors. Accordingly, the previous positions of particles correspond to the prior community, while the new positions can represent the final community assignments.

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

Thus, the value of X_{ij} for particle i is obtained as an integer within the range 1to $\deg(n_i)$, meaning $X_{ij} \in \{1,2,\ldots,\deg(n_i)\}$. This adjustment fundamentally improves the PSO algorithm and enhances the system's search capabilities.
Particle position updates are carried out according to Equations 9 and 10, which provide a detailed description of the update mechanism.

$$X_{ij}(t+1) = \begin{cases} k, & \text{if } [\rho < sig(V_{ij}(t+1))] \wedge [deg(n_j) > 1] \\ X_{ij}(t), & \text{otherwise} \end{cases}$$

$$sig(V_{ij}) = \left| \frac{1 - \exp(-V_{ij})}{1 + \exp(-V_{ij})} \right|$$

Particle Position Update Based on Neighborhood Distribution

In this context, $k = \text{rand} \times \deg(n_j)$, where $k \neq X_{ij}(t)$, $\deg(n_j)$represents the degree of node n_j, and $\rho$is a user-defined threshold. It is important to note that the generated position values are based on the degree distribution, indicating that if the value of node $v_j$among its neighbors is higher, $\text{sig}(V_{ij}(t+1))$will always be greater than ρ. Consequently, neighbors of nodes should be moved to the currently selected neighbors. Therefore, the function sig()in Equation 11 has been modified to address this issue.

Particle positions are likely to change gradually through the particle velocity reduction method, allowing the PSO algorithm to progressively converge toward a global optimum.

Artificial Intelligence Techniques

Reinforcement Learning (RL) is a subfield of artificial intelligence that addresses goal-directed agent problems in uncertain environments. In the RL model, there are two main components: the agent and the environment. The agent observes the environment, performs actions to modify it, and receives rewards. The ultimate objective of the agent is to maximize the cumulative reward.

In this study, a Q-Learning approach is employed. The experiments include four implementations of community detection algorithms, each based on different detection strategies. Other implementations can also be integrated with the proposed solution. These include the Leading Eigenvector Newman algorithm, Walktrap, Label Propagation, and Multilevel algorithm, which are used for eigenvalue computation, community identification, and network modularity optimization.

Reinforcement Learning and Learning Agents

Reinforcement learning is an AI method in which an agent performs actions in the environment and receives numerical rewards. The agent observes the environment, selects actions that lead the environment to a new state, and aims to maximize the total reward. This process relies on systematic trial-and-error and learning algorithms.

One RL approach, temporal-difference learning, gradually gathers information about the best action in each state. Various strategies can be used to select actions. With online learning capability, agent performance in multi-agent systems improves, referred to as multi-agent reinforcement learning (MARL). In complex environments, designing agents from scratch may

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

be impossible or difficult; therefore, multiple RL agents interact with the dynamic environment to learn. At the end of each iteration, the environment evaluates the outcomes and compares agent performances, then agents update their actions based on these results.

Multi-Agent Reinforcement Learning Community Detection (MARLCD)

The MARLCD algorithm is designed to detect communities in complex networks. Here, RL agents iteratively attempt to find connected communities over MNLI repetitions. The algorithm involves maindependent agents, each searching for communities in every round. Based on the evaluation of results, agents update their action probability vectors.

The community detection process includes:

1. Selecting unvisited nodes.
2. Creating paths for exploration.
3. Performing actions until predefined conditions are met.
4. Updating action probability vectors.
5. Removing nodes from the current set.

Finally, the detected communities are evaluated. Their quality is computed using a **normalized cut-based objective function**, which assesses community quality based on intra-community similarity and inter-community differences, aiming to identify the best set of communities.

Below, the pseudocode of the **MARLCD algorithm** for community detection in complex networks is presented.

1. For t = 1 to MNLI
2. $G' \leftarrow G$
3. K ← 1
4. Repeat // making k community
5. $v_i \leftarrow a\ random\ vertex\ of G'$
6. $c_k \leftarrow v_i$
7. $L^t \leftarrow v_i$
8. $While\ (d_{in}(c_k) > d_{out}(c_k) AND\ |a_i| \neq 0) Do$
9. //Finding k^{th} community
10. a_j ←an action selected by agent using p_i
11. v_i ← vertex correspond to a_j
12. IF$(d_{in}(c_k\ \cup v_i) > d_{in}(c_k)\ AND\ d_{out}(c_k\ \cup v_i) <\ d_{out}(c_k) Then$
13. $c_k \leftarrow c_k \cup v_i$
14. $L^t \leftarrow L^t \cup v_i$
15. $v_i \leftarrow v_j$

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

16. $Else$
17. a_j of agent is deactivated until next round
18. End If
19. End While
20. $G' \leftarrow G' \backslash C_k$
21. $k \leftarrow k + 1$
22. Until ($|G'| \neq 0$)// making k community
23. $NC(C^t) = \frac{1}{k} \Sigma_{i=1}^{k} \frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$
24. If ($NC(C^t) < NC(C^{t-1})$)
25. reward the selected actions along the path L^t
26. End If
27. Next // end of each round
28. Return C^t

Community Reward and Agent Evaluation

If the **normalized cut value (NCC)** in the current round is less than or equal to the normalized cut value in the previous round (line 25), the actions selected along the path $L$by the agent receive a reward according to the reinforcement learning algorithm described in the previous section (Equation 2, line 26).

After each of the maagents executes a round of the algorithm and evaluates the results, if the normalized cut value of a given agent in the current round is lower than the normalized cut values of all agents up to the current round, this agent is considered a **successful agent**. All actions selected by this successful agent along the path are rewarded in other agents as well, according to the reinforcement learning scheme (Equation 2).

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

The proposed algorithm is executed uniformly for **MNLI** rounds for each agent.

Table 1. Information on Networks Used in Experimental Studies

Network	Number of Nodes	Number of Edges	Description
Karate	۳۴	۷۸	Karate Club Network
Dolphins	۶۲	۱۵۹	Dolphins Network [50]
Books	۱۰۵	۴۴۱	U.S. Political Books Network [51]
Football	۱۱۵	۶۱۵	U.S. College Football Teams Network [17]
LFRI	۱۰۰۰	۳۸۱۶۰	LFR Synthetic Network [32]
LFR2	۵۰۰۰	۲۵۰۰۰۰	LFR Synthetic Network

Evaluation Metrics

To evaluate the performance of **MARLCD** and other community detection algorithms used for comparison, two commonly employed metrics for assessing the quality of detected community sets in networks are utilized. These metrics are defined as follows:

$$Q = \frac{1}{2m}\Sigma_{c\in p}\Sigma_{V_i,V_i\in C}[A_{i,j} - \frac{k_i k_j}{2m}$$

Where **A** is the adjacency matrix, such that $A_{i,j} = 1$if there is an edge between vertices $i$and j, and 0 otherwise. $k_i = \sum_j A_{i,j}$denotes the degree of vertex i, and $m$is the total number of edges in the network. The summation is performed over all pairs of vertices belonging to the community $C$of the partition P.

$$MI(A,B) = \frac{-2\Sigma_{i=1}^{c_a}\Sigma_{j=1}^{c_b}D_{i,j}\log\left(\frac{D_{i,j}\mathrm{n}}{D_{i,j}}\right)}{\Sigma_{i=1}^{c_a}D_i\log\left(\frac{D_i}{n}\right)\Sigma_{j=1}^{c_b}D_j\log\left(\frac{D_j}{n}\right)}$$

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

Normalized Mutual Information (NMI) Normalized Mutual Information (NMI) is also one of the important metrics for evaluating the results of community detection algorithms. It is used to estimate the similarity between the communities detected by an algorithm and the actual communities. Suppose $A$and $B$are two partitions of a network, representing the number of communities in $A$and B, respectively. Let Cdenote the number of communities in B. $D$is the confusion matrix, and D_{ij}indicates the number of nodes in community $i$of Athat also appear in community $j$of B. $D_{i.}$is the sum of the i-th row of D, and $D_{.j}$is the sum of the j-th column of D. The definition of NMI (A, B) is as follows:

NMI takes a value between 0 and 1, where higher values indicate that communities $A$and $B$are more similar. This metric is commonly used for networks for which we have prior knowledge of the actual communities.

Parameter Settings of MARLCD The parameter values for MARLCD are determined based on a series of experiments. The chosen values are such that the best results are obtained in terms of both solution quality and computation time. Table 2 shows the parameter settings of the proposed approach. Nevertheless, below, the reasons for selecting the values of some key parameters are examined based on the results of certain experiments.

When developing algorithms based on intelligent agents, selecting appropriate learning parameters has a significant impact on the algorithm's performance. For example, the learning rate parameter is one of them. In other words, successful learning in reinforcement learning algorithms strongly depends on the precise tuning of learning parameters. Achieving high-quality results and good performance is critically related to proper parameter adjustment.

Figure 3 shows the results of executing the MARLCD algorithm with different learning rate values. In this figure, modularity values for the Karate and Dolphins networks are illustrated with learning rates varying from 1 to 100. From the chart, it can be concluded that the performance of the MARLCD algorithm fully depends on the choice of learning rate. Typically, lower learning rates lead to better algorithm performance in the networks under study, while higher values result in poorer performance. According to the obtained results, within a certain range of values (0.15–0.35), the algorithm shows better performance. Consequently, the learning rate for agents in the MARLCD algorithm is set within the range [0.15, 0.35), with each agent independently selecting a value within this range.

Table (2). Parameter settings for the MARLCD approach

Concept	Value	Parameter
Reward	[0.8, 1.3]	r_d
Learning rate	[0.15, 0.35]	lr
Number of agents	5	m_a
Maximum number of learning iterations	2000	$MNLI$

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

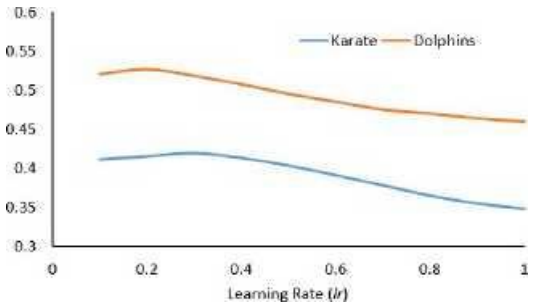

Figure 1: Results of executing MARLCD on the Karate and Dolphins networks with different learning rates based on value and modularity.

In **Figure 4**, when the reward parameter $rd$is around 1, better results are observed. However, by decreasing it below 1 and toward 0, the algorithm's performance drops significantly. Similarly, increasing it to values above 1 has an adverse effect on the algorithm's performance. Therefore, the reward parameter $rd$for agents in MARLCD is set within the range [0.8, 1.3), with each agent independently selecting a value within this range.

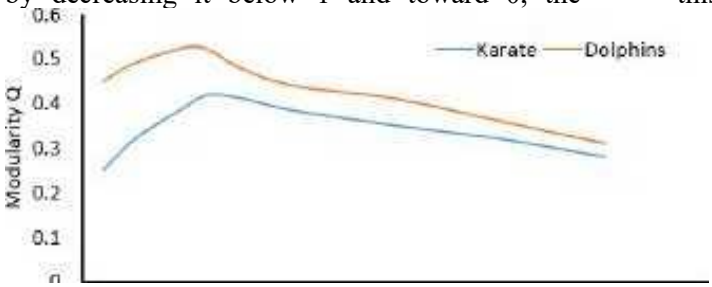

Figure 2: Results of MARLCD on the Karate and Dolphins networks with different reward parameters based on value and modularity Q.

Experiments on Real Networks

Figure 3 shows the best result obtained by MARLCD on the Dolphin network. According to Figure 3, the number of clusters found by the algorithm is 2, which matches the actual structure of this network.

In this section, the results of the proposed algorithm are compared with several well-known community detection methods, including GA-Net by Pizzuti, Meme Net by Gong et al., LPA by Liu and Murata, and MAGA-Net by Li and Liu. The comparison is based on modularity $Q$and NMI metrics on real-world networks. The results obtained by different methods according to the majority criterion are presented in Table 3.

According to Table 3, for the Books, Karate, and Football networks, the proposed algorithm shows better or at least comparable performance compared to other methods based on the average modularity Q_{ars}. For the Dolphins network, the results of MARLCD outperform Meme Net, GA-Net, and LPA, while slightly lower than MAGA-Net, indicating that the results are in a similar range. The average Q_{ars}obtained by MARLCD is approximately 12.33% higher than the average Q_{ars}obtained by the three other algorithms.

From the perspective of the maximum modularity Q_{max}, the proposed method achieves higher quality in all four networks compared to the other methods and produces the best results. This indicates that MARLCD is capable of detecting appropriate communities.

Furthermore, the computation time of the proposed algorithm is lower in all four networks (Football, Books, Dolphins, Karate) compared to LPA and MAGA-Net. The computation times for GA-Net and Meme Net were not reported.

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

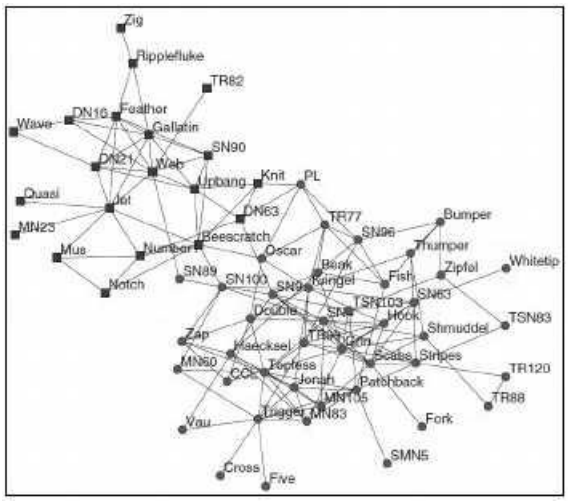

Figure 3: Two communities detected by MARLCD in the Dolphins network.

Table 3. MARLCD and GA-Net Algorithms

روش / شبکه		Karate	Dolphins	Books	Football	Avg.
GA-Net	Q_{avg}	0.374	0.492	0.487	0.502	0.464
	Q_{max}	0.419	0.522	0.521	0.556	0.505
	Q_{std}	0.076	0.011	0.036	0.023	0.037
	Time(s)					
Meme-Net	Q_{avg}	0.408	0.427	0.443	0.490	0.442
	Q_{max}	0.419	0.502	0.513	0.549	0.496
	Q_{std}	0.013	0.305	0.021	0.023	0.091
	Time(s)					
LPA	Q_{avg}	0.352	0.495	0.493	0.579	0.48
	Q_{max}	0.399	0.516	0.522	0.604	0.51
	Q_{std}	0.028	0.008	0.020	0.018	0.019
	Time(s)	0.009	0.019	0.048	0.049	0.031
MAGA-Net	Q_{avg}	0.419	0.527	0.527	0.602	0.519
	Q_{max}	0.419	0.528	0.527	0.604	0.52
	Q_{std}	0.002	0.001	0.001	0.003	0.002
	Time(s)	0.021	0.073	0.268	0.378	0.185
MARLCD	Q_{avg}	0.419	0.526	0.527	0.603	0.519
	Q_{max}	0.420	0.528	0.527	0.607	0.521
	Q_{std}	0.001	0.001	0.001	0.002	0.001
	Time(s)	0.016	0.052	0.143	0.309	0.13

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

Based on the majority criterion, the results obtained from executing the MARLCD algorithm and GA-Net, Meme-Net, and MAGA-Net on real networks according to the NMI metric are reported in Figure 4. This metric is commonly used to measure the similarity between the communities detected by the algorithms and the actual communities. As shown in Figure 4, the communities detected by MARLCD have higher similarity to the actual communities compared to the three other algorithms, except for the result obtained by MAGA-Net on the Dolphins network. Across the other networks, MARLCD shows the closest match to the real communities. The average NMI obtained by MARLCD is approximately 9.85% higher than the average NMI obtained by the three other algorithms. In other words, the communities detected by the proposed algorithm are, on average, 9.85% more similar to the actual communities compared to Meme-Net, GA-Net, and MAGA-Net.

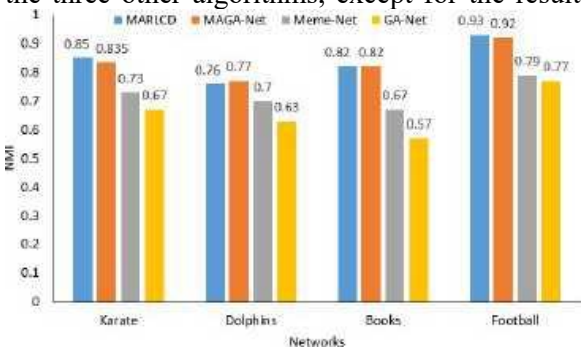

Figure 4: Comparison of the average NMI values obtained by MAGA-Net, Meme-Net, GA-Net, and MARLCD across four networks: Football, Books, Dolphins, and Karate.

Experiments on Synthetic Networks

To further evaluate the performance of the MARLCD algorithm, synthetic networks from the LFK (LFR benchmark) dataset were used. We generated networks of 1,000 and 5,000 nodes. The LFR synthetic networks, proposed by Lancichinetti et al., are widely used for systematically assessing the quality of community detection algorithms. In the experiments, six networks of 1,000 nodes (LFR1) were generated with mixing parameter values of 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, and six networks of 5,000 nodes (LFR5) were generated with 12 similar mixing parameter values. The mixing parameter μrepresents the fraction of edges a node shares outside its community versus within its own community. As the mixing parameter increases, community detection becomes more difficult. Other network parameters for the generated LFR networks were set as follows: average node degree $K = 15$, maximum node degree $max = 50$, minimum community size $min = 10$, and maximum community size $max = 50$.

Figures 7 and 8 show the average NMI values obtained from running MARLCD, MAGA-Net, LPA, Meme-Net, and GA-Net on these networks. As shown in Figure 5, for networks with small mixing parameter values, all algorithms produced similar and high NMI values, indicating acceptable performance in this category of networks. However, as the mixing parameter increases, leading to more complex networks, the NMI values for all algorithms decrease. The figure also shows that MARLCD, LPA, and MAGA-Net perform well on 1,000-node LFR networks, with NMI values ranging from approximately 1 to 0.85 for mixing parameters from 1 to 16, demonstrating the high similarity between the detected communities and the true communities. In contrast, GA-Net and Meme-Net show weaker performance, especially at higher mixing parameter values. Overall, the average NMI obtained by MARLCD is approximately 21.71% higher than the average NMI obtained by the other four algorithms. These results highlight the weakness of GA-Net and Meme-Net in large-scale networks, particularly for high mixing

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

parameter values, while LPA, MARLCD, and MAGA-Net can produce reliable results even for very large networks. As mentioned earlier, increasing the mixing parameter from 0.1 to 16 increases the difficulty of community detection. Despite this, MARLCD maintains strong performance even at high mixing parameter values, confirming the high quality of the proposed approach. The average NMI obtained by MARLCD is approximately 21.89% higher than that of the other four algorithms.

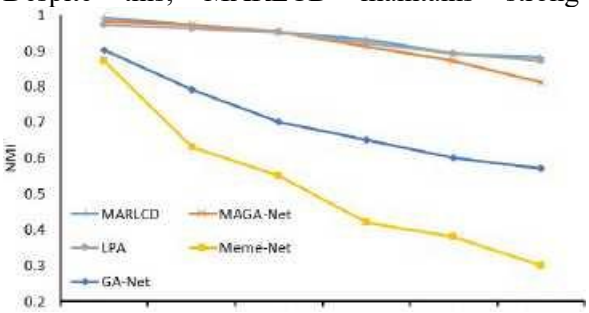

Figure (5) illustrates the comparison of average NMI values obtained by applying LPA, MemeNet, GA-Net, and MARLCD/MAGA-Net algorithms on synthetic LFR networks with 1000 nodes (LFR1), for combined parameter values ranging from 0.1 to 0.16.

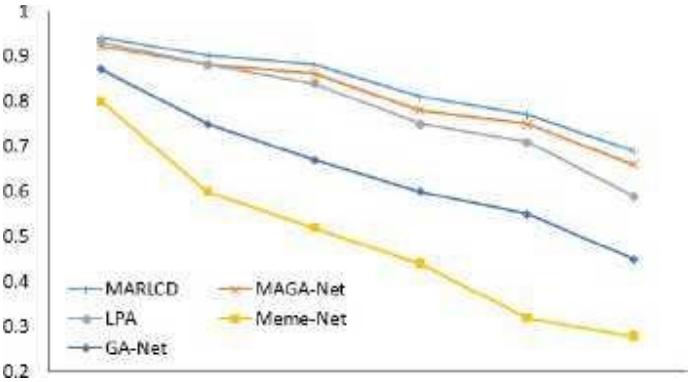

Figure (6) presents a comparison of the average NMI values obtained by applying LPA, Meme-Net, GA-Net, and MARLCD/MAGA-Net algorithms on synthetic LFR networks with 5000 nodes (LFRI), with a combined parameter value of 2 in the range 0.1–0.16.

Performance Evaluation of MARLCD

Based on the evaluations and results reported in this section, MARLCD demonstrates better and more stable performance compared to other well-known community detection algorithms. It converges quickly, produces consistent results for both small and large-scale networks, and can manage networks with up to 5,000 nodes.

Experiments and Discussion

To validate our approach, several experiments were conducted. The algorithms mentioned above, each implementing a different community detection strategy, were used to implement the reinforcement learning (RL) framework. The experiments utilized both a synthetic network generated using the Erdős–Rényi model for

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

validating the RL implementation and a real-world dataset from the High Energy Physics Theory citation network available in the SNAP Project. Sosa and Zhao evaluated existing community detection solutions using the iGraph package across various scenarios. Inspired by this evaluation, the reinforcement learning framework proposed in this study uses iGraph to implement different community detection algorithms. To assess the quality of the community structure, modularity scores were used, as described in Section 3. The OpenAI Gym library was used to implement the RL environment, while iGraph managed network data. A null model was first used as the baseline, where a policy selected completely random actions ($\epsilon = 1$). This policy is purely random, and the results are shown in Figure 3. This null model serves as a reference for comparison. Subsequently, a greedy policy was adopted to balance exploration and exploitation, and SARSA was used to update the policy, primarily due to the unpredictable nature of future states in a dynamic network. Executing the RL implementation on the High Energy Physics Theory citation network produced a reward accumulation graph shown in Figure 2, illustrating the evolution of cumulative rewards over episodes.

Application to Lightweight H-Beam Alloy Interaction Networks

To further assess the algorithm, MARLCD was applied to a weighted interaction network of lightweight H-beam alloys. The network considers feature similarity between alloys, where link weights represent the degree of similarity. Initially, the interaction network was constructed based on compositional and structural descriptor similarity. All nodes were connected to form a fully connected graph with 90 nodes representing different alloy compositions, as detailed in Appendix A. Each node was connected to every other node with 3,968 edges, forming an undirected interaction network. The degree distribution of the network, representing the probability distribution of node degrees, is shown in Figure 3, with an average degree of 14.20. Next, as shown in Figure 4, an α coefficient of 0.9 and a threshold of 0.6 were applied to the network to remove weak and less similar connections. The resulting interaction network contained 632 edges while preserving relationships between the 90 nodes. Node sizes in Figure 4 reflect the degree of each node.

Community Detection Results

Applying the Louvain algorithm to the lightweight H-beam alloy network, as shown in Figure 5, resulted in 13 communities with an overall modularity of approximately 0.71. Each community is represented by a unique color, and all compositions within each community are fully connected. The Particle Swarm Optimization (PSO) algorithm was also applied to the same network over 100 iterations, extracting 13 optimized communities with improved modularity of approximately 0.89, as shown in Figure 6. Since nodes in each community are not connected to nodes in other communities, the PSO-optimized communities have higher-quality modularity. Analysis of each community shows that neighbors of each alloy composition share the same phase label and similar elements. In this study, modularity was used as the main metric to evaluate community quality. If the number of edges within a community is no greater than a random graph, the modularity is effectively zero. Maximum modularity occurs when all internal nodes are fully connected with no external edges. Modularity also allows comparison of communities across different algorithms. Since other algorithms may not produce identical results, many metrics cannot evaluate community quality consistently. Using the hierarchical Louvain method, modularity trends during the community splitting or merging process can be analyzed, with the maximum value representing the best outcome. Modularity ranges from -1 to 1, measuring the density of internal connections relative to inter-community connections. A modularity between 0.3 and 0.7 indicates a strong community; values closer to 1 denote very high-quality communities. Empirical results show that both algorithms achieve modularity above 0.7.

Community Analysis and Practical Implications

The extracted communities demonstrate high quality and accuracy, confirmed via modularity metrics. Table 1 summarizes community quality results obtained using the Louvain and PSO-

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

optimized algorithms. In Louvain, modularity remained constant at 0.71, while in PSO, modularity increased from 0.87 to 0.89 over 30 iterations, remaining stable after 150 iterations.

Community detection in lightweight H-beam networks provides practical insights: similar-phase alloys within the same community exhibit similar behaviors and properties. Once communities are identified, properties such as the maximum number of elements in an alloy can be predicted. Phase prediction using machine learning techniques is also possible; unknown-phase compositions can be inferred from other compositions in the same community. Table 2 shows phase counts per community and prediction accuracy, with Louvain and PSO achieving approximately 88% and 93% accuracy, respectively.

Related Work

Several studies have explored deep reinforcement learning in recommender systems. Table 4 provides a summary and analysis of key works in this domain.

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

Table 4. Analysis of Related Works

Row	Year	Authors	Article Title	Findings	Advantages & Limitations
1	2021	Chen, Y. Y.; Hong, U. T.; Niwattanakul, S.	Machine Learning Approach for Community Detection in a High-Entropy Alloy Interaction Network	Using community detection and optimization algorithms, the communities of high-entropy alloys (HEAs) were identified with greater accuracy.	This study applies modern community detection and machine learning methods to HEAs using **Louvain** and **PSO** algorithms. The approach can improve prediction and analysis of new alloy properties and non-laboratory scenarios. Limitations: requires specific data and conditions; further analysis and validation are needed for industrial and material research applications.
2	2024	Wasserman, S.; Faust, K.; Csardi, G.; Nepusz, T.	Towards Modularity Optimization Using Reinforcement Learning for Community Detection in Dynamic Social Networks	Reinforcement learning was applied to optimize modularity in dynamic social network community detection. Comparative results show the approach is acceptable and effective.	Advantages: Demonstrates success of RL-based community detection and comparable evaluation metrics. Limitations: May have limited adaptability; RL implementation requires careful design to improve results.

Row	Year	Authors	Article Title	Findings	Advantages & Limitations
3	2024	Mir Mohammad Alipour & Mohsen Abdolhosseinzadeh	A Novel Algorithm for Community Detection Using Multi-Agent Reinforcement Learning	A new algorithm was proposed and evaluated for community detection. Performance, speed, and stability were assessed, showing an improvement of 21% over similar algorithms.	Advantages: High accuracy and fast detection of communities. Limitations: Handling incomplete information and constraints may negatively impact performance.

Conclusion

In this paper, a novel algorithm for community detection in complex networks using multi-agent reinforcement learning (MARL) has been proposed. The algorithm, leveraging the distinct characteristics of agents and their interactions, is capable of identifying highly dense local communities. Empirical studies were conducted on four real-world networks as well as a set of LFR synthetic networks, demonstrating that the proposed algorithm significantly outperforms the compared methods. This approach efficiently identifies optimal partitions with high speed, accuracy, and stability. The study specifically applied this method to detect communities in lightweight H-beam alloy compositions (HEAs without floating elements) and showed that the algorithm can be extended to other areas in materials science. This approach can assist in identifying useful alloy compositions for industrial applications and enable accurate predictions of future phases. Finally, the use of reinforcement learning to optimize community detection solutions in dynamic social networks has been shown to be effective and highlights the flexibility of reinforcement learning for addressing more general problem-solving tasks in community detection.

References

[١]DJ Watts and S. H. Strogatz, "Collective dynamics of small-world networks, nature, vol. 393, no 6684, pp 440, 1998 .

[٢]S. Boccaleth, V Latora, Y Moreno, M Chavez, and D.-U. Hwang, "Complex networks. Structure and dynamics, Physics reports, vol 424, no. 4-5, pp. 175-308, 2006 .

[٣]M. E Newman, "The structure and function of complex networks, SIAM review, vol 45, no 2, pp 167-256, 2003

[٤]S. Wasserman and K. Faust, Social network analysis. Methods and applications Cambridge university press, 1994 .

[٥]R. Milo, S. Shen-Orr, S. Itzkovitz, N Kashtan, D Chklovski, and U Alon, "Network motifs simple building blocks of complex networks," Science, vol 298, no. 5594, pp 824-827,2002

[٦]U Brandes et al. On modularity clustering," IEEE transactions on knowledge and data engineering, vol. 20, no 2, pp. 172-188, 2007

[٧]J. Liu. W Zhong, and L. Jiao, 'A multuagent evolutionary algorithm for combinatorial Systems, Man, and Cybernetics, Part B (Cybernetica), vol. 40, no 1, pp. 329-340, 2009.

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

[٨]J Lau, W. Zhong. and L. Jiao, "A multiagent evolutionary algorithm satisfaction problems," IEEE Transactions ON Systems, Man, and Cybernetics, Part B (Cybernetics), vol 36, no. 1, pp. 54-73, 2006 .

[٩]M M. Alipour and S. N. Razavi, "A new multiagent reinforcement learning algorithm to solve the symmetric traveling salesman problem, Multiagent and Grid Systems, vol. 11, no. 2, pp 107-119, 2015 .

[١٠]MM Alipour, SN Razavi, MRF Derakhshi, and M A Balafar, "A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem," Neural Computing and Applications, pp 1-17, 2017

[١١]M. M. Alipour and M. Abdolhosseinzadeh, "A multiagent reinforcement learning algorithm to solve the maximum independent set problem, Multiagent and Grid Systems, vol 16, no. 1, pp 101-115, 2020 .

[١٣]ME Newman, "Modulanty and community. structure in networks." Proceedings of the national academy of sciences, vel 103, no 23. pp. 8577-8582, 2006 .

[١٣]A Clauset, M. E. Newman, and C Moore, "Finding community structure in very large networks, Physical review E. vol 70, no 6, p. 066111, 2004

[١٤]J M. Kumpula, J. Saramakı, K. Kaskı, and J Kertész, 'Limited multiresolution methods in complex network community detection, Fluctuation and Noise Letters, vol. 7, no 03, pp. 1209-L214, 2007

[١٥]S. Fortunato, Community detection in graphs. Physica reporta, vol 486, no. 3-5, pp. 75-174, 2010 .

[١٦]L. Donetti and M A Munoz, 'Detecting network communities a new systematic and efficient algorithm. Journal of Statistical Mechanics: Theory and Experiment, vol 2004, no. 10. pp. P10013, 2004 .

[١٧]M. Girvan and M. E Newman, Community structure social and biological networks, Proceedings of the national academy of sciences, vol 99, no. 12, pp. 7821-7826, 2002

[١٨]M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical review E", vol 69, no 2, pp. 026113, 2004 .

[١٩]F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisı, "Defining and identifying communities in networks ",Proceedings of the national academy of sciences, vol 101, no 9, pp. 2658-2663, 2004

[٢٠]L. Hagen and A B. Kahng, "A new approach to effective circuit clustering, in ICCAD, P. De Meo, E. Ferrara, G. Fiumara, and A Provetti, "Mixing local and global information . community detection in complex networks using multiobjective evolutionary algorithms." IEEE Congress on Evolutionary Computation, 2010 IEEE, pp. 1-7

[٢١]J. Cao, Z. Bu, Y. Wang, H. Yang, J. Jiang, and H.-J. Li, Detecting prosumer-community groups in smart grids from the multiagent perspective, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 8, pp. 1652-1664, 2019 .

[٢٢]C.-K. Han, S.-F. Cheng, and P. Varakantham, "A Homophily-Free Community Detection Framework for Trajectories with Delayed Responses, in Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2019. International Foundation for Autonomous. Agents and Multiagent Systems, pp 2003- 2005

[٢٣]X Feng and X Yang. "Fast convergent average consensus of multiagent systems based on community detection algorithm," Advances in Difference Equations, vol. 2018, no. 1, pp. 1-13, 2018 .

[٢٤]P Pons and M. Latapy, "Computing communities in large networks using random walks, J. Graph Algorithms Appl., vol 10, no. 2, pp. 191-218, 2006 .

[٢٥]P Ronhovde and "Multiresolution community detection for megascale networks by information-based replica correlations, Physical Review E, vol 80, no. 1, p. 016109, 2009

[٢٦]M. Zhou and J. Liu, "A memetic algorithm for enhancing the robustness of scale-free networks against malicious attacks," Physical A. Statistical Mechanics and its Applications, vol. 410, pp 131-143, 2014 .

[٢٧]T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning, IEEE

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

Transactions on computers, vol. 45, no 7, pp. 841-855, 1996

[۲۸]M. Tasgin, A. Herdagdelen, and H. Bingol, "Community detection in complex networks using genetic algorithms, arXiv preprint arXiv: 0711.0491, 2007 .

[۲۹]C Pizzuti, "Ga-net A genetic algorithm for community detection in social networks," in International conference on parallel problem solving from nature, 2008 Springer, pp. 1081-1090

[۳۰]A Gog. D. Dumitrescu, and B. Hirsbrunner, *Community detection in complex networks using collaborative evolutionary algorithms," in European Conference on Artificial Life, 2007. Springer, pp 886-894 .

[۳۱]M. Gong. B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks, Physical Review E, vol 84, no. 5, p. 056101, 2011 .

[۳۲] J Liu, W Zhong, HA Abbass, and D. G. "Separated and overlapping]Downloaded from jsúp rasp ac ir on 2024-05-30 community detection in complex networks using multiobjective evolutionary algorithms." IEEE Congress on Evolutionary Computation, 2010 IEEE, pp. 1-7

[۳۳]X. Liu and T. Murata, "Advanced modularity- specialized label propagation algorithm for detecting communities in networks," Physica A: Statistical Mechanics and its Applications, vol. 389, no. 7, pp. 1493-1500, 2010 .

[۳۴]J. Duch and A. Arenas, "Community detection complex networks using extremal optimization, Physical review E, vol. 72, no 2. p 027104, 2005

[۳۵]B Yang and DY Lau, "Force-based incremental algonthm for mining community structure in dynamic network," Journal of Computer Science and Technology, vol. 21. no 3. pp. 393-400, 2006 .

[۳۶]I Gunes and H Bingol, "Community detection in complex networks using agents," arXiv preprint cs/۰۶۱۰۱۲۹, 2006

[۳۷]Lt and J Liu, "A multi-agent genetic algorithm for commumty detection in complex networks, Physica A Statistical Mechames and its Applications, vol. 449, pp. 336-347,2016 .

[۳۸]J. Huang, B. Yang, D. Jin, and Y. Yang. "Decentralized mining social network communities with agents. Mathematical and Computer Modelling, vol 57, no. 11-12, pp. 2998-3008, 2013

[۳۹]G. Palla, I Derényı, I Farkas, and T. Vicsek, "Uncovering the overlapping community. structure of complex networks in nature and society, nature, vol 435, no. 7043, p. 814. 2005 .

[۴۰]S. J Russell and P Norvig, Artificial intelligence a modern approach Malaysia, Pearson Education Limited, 2016 .

[۴۱]R. S. Sutton and A. G. Barto, Reinforcement learning An introduction. Cambridge MIT press, 1998 .

[۴۲]L. P. Kaelbling, M L Littman, and A. W Moore, "Reinforcement learning A survey," Journal of Artificial Intelligence Research, vol. 4, pp. 237-285, 1996

[۴۳]P. Stone and M Veloso, Multuagent systems A survey from the machine learning perspective, Autonomous Robots, vol 8, no 3. pp. 345-383, 2000 .

[۴۴]S. Sen and G. Weiss, "Learning in multiagent systems, in Multragent Systems A Modern Distributed Intelligence, G Weiss Ed MIT Press, 1999, ch. 6, pp. 259-298

[۴۵]R. S Sutton and A. G. Barto, Reinforcement Learning An Introduction MIT Press, 1998

[۴۶]IS Dhillon, Y. Guan, and B Kulis. "Kernel k-means spectral clustering and normalized for community detection in large networks," Journal of Computer and System Sciences, vol. 80, no. 1, pp 72-87, 2014

[۴۷]J. Shi and J Malik, "Normalized cuts and image segmentation, Departmental Papers (CIS), pp. 107, 2000

[۴۸]W W Zachary, "An information flow model for conflict and fission in small groups." Journal of anthropological research, vol 33. no. 4, pp. 452-473, 1977 .

[۴۹]D Lusseau, K. Schneider, O J. Borsseau, P. Haase, E Slooten, and S. M Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long- lasting associations, Behavioral Ecology and Sociobiology, vol 54, no. 4, pp. 396-405, 2003

[۵۰]V Krebs, "Books about us politics," unpublished, http//:www.orgnet.com, 2004 .

International Journal of Innovative Research In Humanities
Vol.5, NO.2 , P:185-207
Received: 10 January 2024
Accepted: 18 January 2026

[٥١]A Lancichinetti, S. Fortunato, and F Radicchi, "Benchmark graphs for testing community detection algorithms," Physical review E, vol. 78, no 4, p. 046110, 2008 .

[٥٢]L. Danon, A Diaz-Guilera, J. Duch, and A Arenas, "Comparing community structure identification," Journal of Mechanics: Theory and Experiment, vol 2005 ,ne 09, pp. P09008, 2005 .

[٥٣]M. Tokic, F. Schwenker, and G. Palm, "Meta- learning of exploration and exploitation parameters with replacing eligibility traces," presented at the In IAPR International Workshop on Partially Supervised Learning (pp. 68-79). Springer Berlin Heidelberg, 2013. May

[٥٤]K Kobayashi, H. Mizoue, T. Kuremoto, and M. Obayashi, "A meta-learning method based on temporal difference error," presented at the In International Conference on Neural Information Processing (pp. 530-537) Springer Berlin Heidelberg, 2009, December .