applied sciences

MDPI

Article

Distributed Genetic Algorithm for Community Detection in Large Graphs with a Parallel Fuzzy Cognitive Map for Focal Node Identification

Haritha K. [1], Judy M. V. [1], Konstantinos Papageorgiou [2] and Elpiniki Papageorgiou [2,*]

[1] Department of Computer Applications, Cochin University of Science and Technology, Kochi 682022, India; harithak@cusat.ac.in (H.K.); judynair@cusat.ac.in (J.M.V.)

[2] Department of Energy Systems, University of Thessaly, Gaiopolis Campus, 41500 Larissa, Greece; konpapageorgiou@uth.gr

* Correspondence: elpinikipapageorgiou@uth.gr

Abstract: This study addresses the importance of focal nodes in understanding the structural composition of networks. To identify these crucial nodes, a novel technique based on parallel Fuzzy Cognitive Maps (FCMs) is proposed. By utilising the focal nodes produced by the parallel FCMs, the algorithm efficiently creates initial clusters within the population. The community discovery process is accelerated through a distributed genetic algorithm that leverages the focal nodes obtained from the parallel FCM. This approach mitigates the randomness of the algorithm, addressing the limitations of the random population selection commonly found in genetic algorithms. The proposed algorithm improves the performance of the genetic algorithm by enabling informed decision making and forming a better initial population. This enhancement leads to improved convergence and overall algorithm performance. Furthermore, as graph sizes grow, traditional algorithms struggle to handle the increased complexity. To address this challenge, distributed algorithms are necessary for effectively managing larger data sizes and complexity. The proposed method is evaluated on diverse benchmark networks, encompassing both weighted and unweighted networks. The results demonstrate the superior scalability and performance of the proposed approach compared to the existing state-of-the-art methods.

Keywords: parallel fuzzy cognitive maps; distributed genetic algorithm; community detection; focal nodes; social networks

check for updates

Citation: K., H.; V., J.M.; Papageorgiou, K.; Papageorgiou, E. Distributed Genetic Algorithm for Community Detection in Large Graphs with a Parallel Fuzzy Cognitive Map for Focal Node Identification. *Appl. Sci.* **2023**, *13*, 8735. https://doi.org/10.3390/app13158735

Academic Editor: Gaetano Zizzo

Received: 21 May 2023
Revised: 29 June 2023
Accepted: 14 July 2023
Published: 28 July 2023

1. Introduction

Recent years have seen a dramatic increase in the amount of research on fuzzy cognitive maps. Numerous applications of FCMs have been made in a variety of fields due to their capacity to improve accuracy and handle uncertainties in data. Network science is one such field in which FCMs' potential is being intensively studied. Identifying the focal nodes that make up a network is a significant challenge in network analysis. It has various applications ranging from speeding up the propagation of information through the network to understanding a particular network's organization [1–4]. The term "focal nodes" refers to the most prominent nodes that make up a network that have a substantial role in the network's overall organisation.

The various possibilities for identifying focal nodes in a given network have been explored in many works in recent years. Many methods based on node centralities, such as degree, closeness [5], betweenness [6], and eigenvector [7], as well as diffusion-based or random-walk-based methods, like PageRank [8] and LeaderRank [2,9], exist. Some of these techniques already account for the effects of neighbouring nodes, but they do not make direct use of the interactions between them. Most of the existing techniques for the discovery of focal nodes do not perform well in large networks. Identifying the focal nodes in the graphs becomes increasingly difficult as the size and complexity of the graphs increase. This paper explores the possibility of using a fuzzy-cognitive-map-based approach to identify the focal nodes in a given network. The fuzzy cognitive map is a system motivated by the human capability of reasoning. It embodies a perception-based system. It uses an iterative learning method to acquire knowledge about a particular system and to determine the system's various attributes. One of the largest impediments to utilising FCM for complicated systems is the difficulty of the algorithm in handling huge datasets. The suggested study tackles this specific issue and discusses a parallel fuzzy cognitive map method. When a genetic algorithm is employed to detect the community structure of the network, there is a quotient of randomness associated with the initial population selection. The focal nodes identified are then used to detect the community structure in the network thus reducing the randomness of the genetic algorithm. Following that, a distributed genetic algorithm is employed to detect the network's communities.

The key contributions of this research study can be summarised as follows:

- A parallel fuzzy cognitive map is used for the identification of focal nodes, which reduces the time complexity of the problem;
- A novel distributed approach for community detection using a genetic algorithm is proposed;
- To accelerate the convergence of the genetic algorithm, the focal nodes of the network are provided as input, thus reducing the randomness of the initial population.

The paper is organized as follows. In Section 2, recent works in fuzzy cognitive maps and the use of genetic algorithms for community detection are discussed. In Section 3, a detailed elucidation of the proposed methods is given. Section 4 elaborates on the experimental framework used. In Section 5, the results obtained are compared and analysed while in Section 6, conclusions are provided.

2. Literature Review

Fuzzy cognitive maps, a soft computing technique based on a human reasoning approach, were proposed by Bart Kosko in 1986 [10]. They were inspired from cognitive maps, proposed by Axelrod in 1976 [11]. Fuzzy cognitive maps, unlike conventional cognitive maps, integrate the power of fuzzy logic into their framework. FCMs have proved to be a good tool for modelling complex systems due to their capability of addressing uncertainties and improving the dataset's accuracy. Recent years have seen a tremendous increase in research related to FCMs. In 2003, an FCM was used along with decision trees for urinary bladder grading [12]. A hybrid model of an FCM with neural networks was used for pattern classification in 2008 [13]. A new text categorization method based on a similar rough set and an FCM was proposed in 2008 [14]. An extension of an FCM to aid in decision making regarding pulmonary infections, known as the intuitionistic fuzzy cognitive map, was introduced by Iakovidis and Papageorgiou [15] and considered the expert's hesitancy in decision making. An FCM combined with ensemble learning for autism identification problems was proposed by Papageorgiou and Kannappan in 2012 [16]. Salmeron presented fuzzy grey cognitive maps for modelling uncertainties [17]. Aguilar proposed dynamic random fuzzy cognitive maps (DRFCM) to model dynamic systems [18]. In 2013, particle swarm optimization and FCMs were used for autism classification [19]. Nápoles et al. proposed a two-step learning process for FCM in which the first step was particle swarm optimization, and the second step was ant colony optimization [20]. A time-dependent FCM used for diagnosing pulmonary diseases was introduced by Bourgani et al. [21]. Ruan et al. developed the belief-degree-distributed fuzzy cognitive maps (BDD-FCMs), where causal connections were represented by a belief structure [22]. In 2018, a fuzzy cognitive map model utilising map-reduce was introduced, presenting a parallel fuzzy cognitive map approach [23]. Choi et al. proposed a big-data-driven fuzzy cognitive map model to handle big datasets using a fuzzy cognitive map [24]. Puerto et al. proposed multilayer fuzzy cognitive maps to diagnose autism spectrum disorder [25]. A model for identifying the pattern of load distribution on the plantar muscle of the foot to detect a flat or cavus foot using fuzzy cognitive maps (FCMs) trained by a genetic algorithm (GA) against a multilayer perceptron neural network (MLPNN) was proposed in 2020 [26].

Community detection recognizes clusters of nodes that are more intimately connected to one another than they are to other nodes in the network. These connected nodes in networks are of crucial importance across different research domains, providing valuable insights [27]. In social network analysis, identifying communities helps uncover hidden social structures, such as groups of friends or communities of interest, facilitating targeted marketing, understanding information diffusion, and analysing online behaviour [28,29]. In biology, community detection aids in uncovering functional modules in protein–protein interaction networks, shedding light on cellular processes, disease mechanisms, and potential drug targets [30]. In financial networks, community detection assists in identifying risk concentrations, systemic vulnerabilities, and contagion paths [31]. A multitude of techniques have been developed to identify communities within networks, such as the Louvain method [32], the Walktrap algorithm [33], the Newman–Girvan algorithm [34], simulated annealing [35], the random walk algorithm [36], influence-guided label propagation [37], IGLP-weighted-ensemble [37], etc. Genetic algorithms are one of the widely adopted techniques for the identification of the community structure of the network [38–43]. It is essential to encode the chromosome when utilising genetic algorithms to solve problems. For community detection, several encoding methods have been employed. The most widely used encoding methods are label-based encoding [38,44–49] and locus-based encoding [39,50–54]. The bulk of the implementations of genetic algorithms make use of modularity as the fitness function [38,44,45,48,49,52,54]. Additionally, fitness functions such as the community score [40,51] and modularity density [46,47] are utilised. The initial population determination is conducted randomly in most of the existing methods. Also, the network's centrality is not considered while evaluating its community structure.

Many methods have been used in literature to detect the most influential nodes in the network. The existing methods can be classified into topology-based methods and diffusion-based methods. Degree-based methods [55], centrality-based methods [5–7], and K-Shell [56] decomposition methods are topology-based. PageRank [8], LeaderRank [2,9], and HITS Score [57] methods are diffusion-based. The topology-based methods take into consideration the attributes of the nodes in the network, like degree, centrality, etc., whereas the diffusion-based methods take into consideration the nodes visited by the diffusion process. Based on the findings of the preceding investigation, we determined that the existing techniques take longer to converge as the size of the network increases in scale. The existing algorithms are incapable of dealing with the massive amounts of data generated across the world.

3. Proposed Methodology

This research introduces a model that utilizes a parallel fuzzy cognitive map approach for the purpose of identifying the most prominent nodes within a designated network. Though an FCM has been implemented in a variety of domains, the use of an FCM in network science is limited. A focal node detection mechanism based on a fuzzy cognitive map was proposed in [58]. Focal nodes have a profound influence on community formation within a network. Focal nodes tend to attract other nodes due to their property of a high degree of centrality. Nodes seeking connectivity, influence, or access to resources are more likely to gravitate towards focal nodes. As nodes join the network, they often form communities around these central figures. Focal nodes often become the core of a community, around which other nodes cluster. They provide a focal point of connectivity and influence, shaping the community's structure and dynamics. Peripheral nodes in the community are connected to the focal node but may have fewer connections with each other. Focal nodes also act as bridges or connectors between communities. Their connections to multiple communities facilitate the exchange of information, resources, or influence across otherwise separate groups. Focal nodes enable the formation of cross-community interactions and integration, contributing to a more cohesive network. Focal nodes also influence the boundaries of communities within a network. Their connections and interactions with nodes from different communities can determine the extent of overlap or separation between these communities. Focal nodes may attract nodes from different communities, leading to the merging or expansion of communities. Alternatively, they may repel nodes from certain communities, resulting in the formation of distinct isolated groups. Focal nodes play a crucial role in maintaining community cohesion. Their high connectivity ensures efficient communication and information flow within the community. Focal nodes often possess a greater influence over decision-making processes and can shape the community's shared goals, norms, and values.

To uncover the community structure of the network, the identified focal nodes are provided to a distributed genetic algorithm. The overall framework of the proposed algorithm is depicted in Figure 1. To identify the focal nodes within a network, the fuzzy cognitive map is provided with the network as input. The focal nodes, thus detected, aid the genetic algorithm in detecting the community structure of the network.

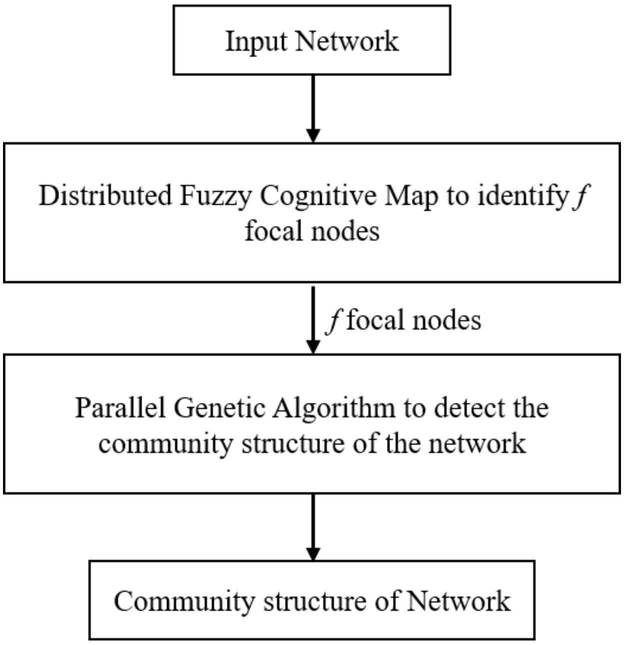

Input Network
Distributed Fuzzy Cognitive Map to identify *f* focal nodes
f focal nodes
Parallel Genetic Algorithm to detect the community structure of the network
Community structure of Network

Figure 1. The genetic algorithm framework for community detection.

3.1. Focal Node Identification Using Distributed Fuzzy Cognitive Map

Fuzzy cognitive maps have a broad spectrum of application domains. Their ability to model uncertainties and improve the accuracy of the modelled system makes them an ideal choice to model extremely simple to highly complex systems. A fuzzy cognitive map (FCM) is a directed graph that contains weighted edges with signed values that represent the fuzzy causal relationships between its nodes; hence, an FCM can be adopted to represent complex networks that comprise nodes and relationships between the nodes. The three components of an FCM are:

- Concepts (C_i)
- Concepts are the fundamental components of a system that have a significant role in resolving the issue at hand.
- State Vector (A) [0, 1]
- A vector is formulated through the process of integrating the values of each individual concept from the given system, which usually falls within the range of 0 to 1.
- Weight Matrix (W_{ij})
- The weight matrix represents a collection of weights that corresponds to all the causal relationships within the system. The presence of a link between the concepts is represented by its weight value; otherwise, it is 0. The diagonal elements are always

zero. The weight values can be positive or negative. A positive weight value indicates a positive causality between the concepts, and a negative value indicates a negative causality.

The steps in using an FCM for focal node identification are initializing the state vector, initializing the weight matrix, applying FCM learning, and finally identifying the focal nodes. An initial state vector that contains the initial values of all the system concepts is determined either by expert knowledge or computationally. The initial state vector is obtained by examining all of the system's characteristics. The initial state vector is subjected to FCM learning until it converges to provide the desired outcome.

3.1.1. Initializing the State Vector and Weight Matrix

The state vector comprises all the values of the concepts. Since the problem under consideration is the identification of focal nodes in a given network, the concept values should depict the degree of connections between the nodes in the network and consider the structural properties of the network. The betweenness centrality [59] measure was used for this purpose. Betweenness centrality was chosen to measure the degree of connections between nodes because it captures the flow of information by considering the number of shortest paths passing through a node. This provides insight into its role in facilitating communication within the network. Additionally, betweenness centrality takes into account the weights of the edges, assigning higher scores to nodes that lie on paths with greater weights. This aspect is especially pertinent in the context of fuzzy cognitive maps (FCMs), where the weights represent the causal relationships between nodes. Equation (1) determines the betweenness centrality measure, which quantifies the number of shortest routes passing through a specific vertex.

$$g(v) = \sum\nolimits_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}. \tag{1}$$

The overall number of shortest routes from the source to the target is equal to σ_{st}, given in Equation (1), and the count of shortest routes that traverse through the vertex v can be determined by $\sigma_{st}(v)$. High centrality ratings imply that a particular vertex is included in a significant percentage of the shortest routes that link pairs of vertices.

For a weighted undirected graph, the weight matrix of the fuzzy cognitive map is initialized with the weight values of the connections between nodes of the network; when there is no connection between two nodes the weight value is set to 0. In the case of unweighted undirected networks, if a connection exists between two nodes, the corresponding weights are set to 1; otherwise, it is 0. A resilient distributed dataset is employed for the storage of both the weight matrix and the state vector. Figure 2 represents the initial state vector and weight matrix evaluated for a sample weighted network.

3.1.2. FCM Learning

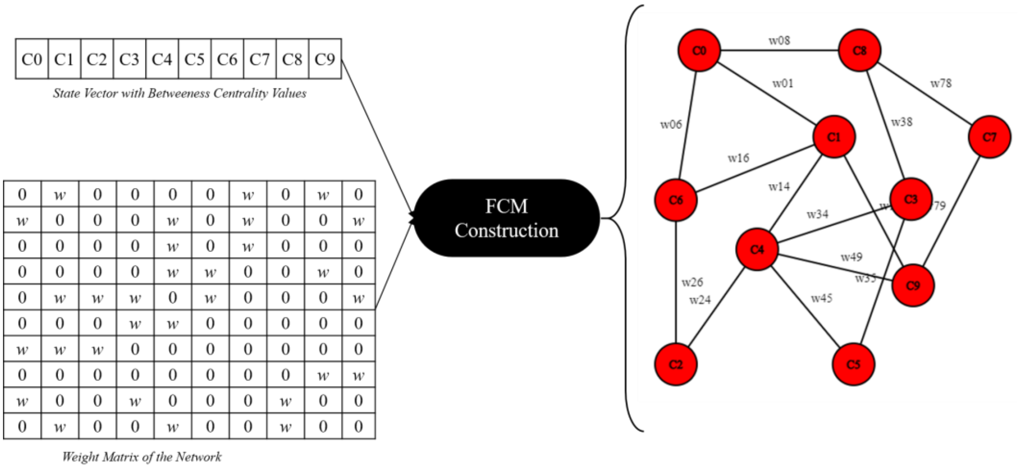
Equation (2) was utilised to apply FCM learning on the initial state vector:

$$A_i^{(k+1)} = f\left(A_i^k + \sum\nolimits_{j \neq i, j=1}^{N} A_j^k \cdot W_{ij}\right), \tag{2}$$

where W_{ij} indicates the weight of the connection between concepts C_i and C_j, and $A_i^{(k+1)}$ denotes the concept C_i at step $k+1$. The threshold function $f(x)$ selected is the sigmoid function defined in Equation (3).

$$f(x) = \frac{1}{1 + e^{-\lambda x}}. \tag{3}$$

Repetitively, the calculation of the state vector continued until epsilon was reached, which represents a residual value indicating the difference in error between successive concepts, with the aim of minimizing this difference. The ε value was set as 0.001. The maximum number of iterations of the FCM was set to 1000. State vector data were then filtered on the basis of a predefined threshold, and the outcome indicated the total number of focal nodes. The λ value was computed using a grid search method where the possible values of λ were assigned in the grid in the range of 1 to 10. The optimum value was determined using exhaustive evaluation.



0	*w*	0	0	0	0	*w*	0	*w*	0
w	0	0	0	*w*	0	*w*	0	0	*w*
0	0	0	0	*w*	0	*w*	0	0	0
0	0	0	0	*w*	*w*	0	0	*w*	0
0	*w*	*w*	*w*	0	*w*	0	0	0	*w*
0	0	0	*w*	*w*	0	0	0	0	0
w	*w*	*w*	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	*w*	*w*
w	0	0	*w*	0	0	0	*w*	0	0
0	*w*	0	0	*w*	0	0	*w*	0	0

Figure 2. Constructing the FCM using the initial state vector and weight matrix.

3.1.3. Parallelization of FCM

This FCM learning process was parallelized in the proposed work. The weight matrix RDD was provided as an input to the *parallelize* function. By employing the *parallelize* function, the weight matrix RDD was partitioned into distinct subsets, each representing specific causal relations within the system. This division led to the creation of multiple new RDDs, each containing a specific subset of weight matrix values. These RDDs were then distributed across individual nodes within the distributed system.

The process of FCM learning entails operating on both the weight matrix and the state vector. In order to enable this process, it is necessary for the state vector to be readily available across all the nodes in which the weight matrix is spread out. To accomplish this task, the broadcast function was utilised on the state vector RDD, thereby replicating the state vector spanning every node in the network. Moreover, the state vector was cached on each of the distributed nodes. The state vector RDD was a unidimensional vector; thus, its replication across nodes resulted in negligible effects on the memory capability of each node. In contrast, the RDD for the weight matrix was characterised by a considerable number of rows and columns, resulting in a substantial spatial requirement. As a result, it was disseminated across the various nodes within the cluster. At each node, the FCM learning process was applied using Equation (1), generating partial results. The final state vector, representing the final global solution, was derived by amalgamating the aforementioned partial outcomes. The parallel learning of the FCM is depicted in Figure 3, which was adopted from [60].

3.2. Parallel Genetic Algorithm to Determine the Community Structure of the Network

A genetic algorithm was used to detect the communities in the network. A genetic algorithm is an optimization method that works with a population of individuals and updates the population until the optimal result is reached. A genetic algorithm maintains the population's genetic diversity through crossovers and mutations. The fitness function was used to determine the generated result's efficiency. Each generation incorporated the best characteristics of the previous generations, resulting in a genetically enhanced generation.

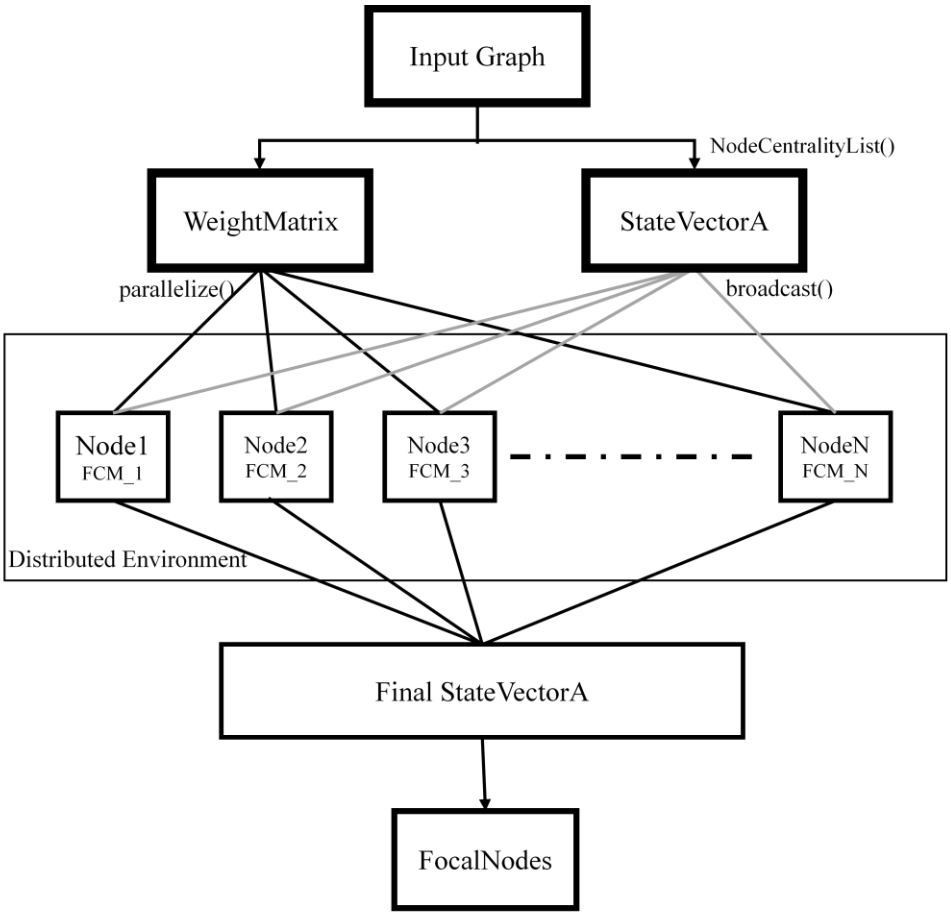


Figure 3. Parallel FCM learning [60].

3.2.1. Initialization of the Population:

The initial communities in the network were created using a deterministic strategy that took into account the focal nodes identified by the FCM. Each prominent node was chosen as a seed node, and the community was expanded by iteratively adding neighbouring nodes based on different similarity criteria such as connectivity, distance measures, etc. A distributed genetic algorithm model was adopted. The population for the genetic algorithm was initialized using the population size and the number of focal nodes obtained from the FCM. Every chromosome within the population was depicted using a label-based encoding technique. Each chromosomal gene related to a community.

3.2.2. Parallelizing the Population and Calculating the Fitness Function

The initialized population was then parallelized using the *parallelize* method into populationRDD. The entire population was divided into segments, and each segment of the population was assigned to an individual node in the distributed environment to process as depicted in Figure 4. The island model was used to parallelize the genetic algorithm in which the entire population was subdivided into finite populations. After mapping the parallel population segments with a fitness function, each population partition was assessed in parallel. Modularity [34] was employed as the fitness function to optimise the solutions until the network achieved the optimal community structure. Modularity is a metric that is used to analyse the different communities in a network, and the ideal community structure has the maximum modularity value. Modularity exists within the range $[-1, 1]$. Equation (4) can be used to accurately estimate the modularity.

$$Q = \frac{1}{2w}\sum_{i=1}^{N}\sum_{j}^{N}\left(w_{ij} - \frac{w_i^{out}w_j^{in}}{2w}\right)\delta\left(C_iC_j\right), \tag{4}$$

where

$$w_i^{out} = \sum_j w_{ij}, \tag{5}$$

$$w_j^{in} = \sum_i w_{ij}, \tag{6}$$

$$2w = \sum_i w_i^{out} = \sum_j w_j^{in} = \sum_{i=1}^{N}\sum_j^{N} w_{ij}, \tag{7}$$

where the Kronecker delta function $\delta\left(C_iC_j\right)$ is 1 if there is a link between vertex i and j, and it is 0 otherwise. Following that, the evolution operations were carried out.

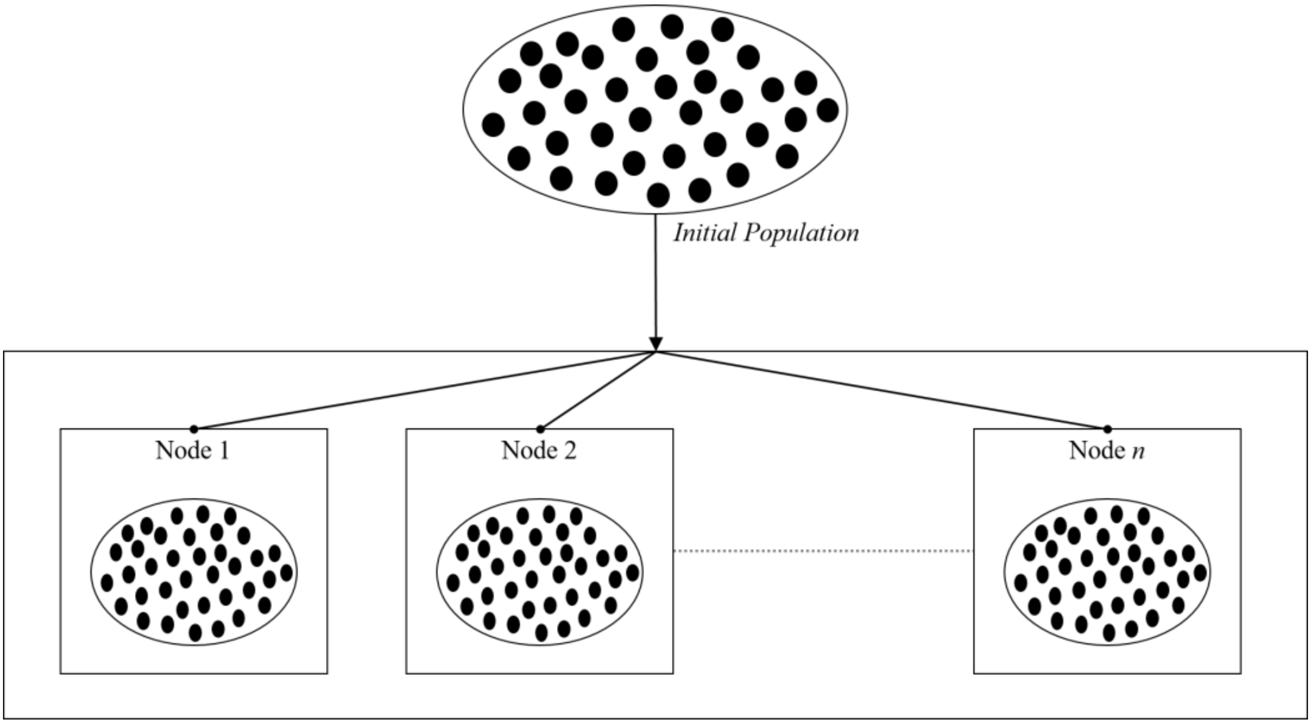


Figure 4. Parallelizing the population in the GA.

3.2.3. Selection, Crossovers, and Mutation

A roulette selection mechanism was used to choose chromosomes depending on their fitness scores. On the basis of a crossover probability value, the crossover technique was implemented in a small subsection of the chromosomes. The descendants were generated using the single-point crossover technique. After applying the crossovers, mutation was applied on randomly chosen chromosomes based on the mutation rate [28]. When the stopping condition was met, the fittest individual in the population was returned; otherwise, the evolution continued. Since the genetic algorithm is a non-deterministic technique, the final results were computed by taking an average of the results of five runs of the model. The pseudocode for the distributed genetic algorithm is given in Algorithm 1.

Algorithm 1 Distributed Genetic Algorithm.

```
procedure ParallelGeneticAlgorithm(no_of_focalnodes, population_size, maxgenerations,
crossover_rate, mutation_rate):
population ← initializePopulation(no_of_focalnodes, population_size); populationRDD
← parallelize(population);
while generation < maxgeneration do
    populationRDD ← populationRDD.map(fitnessFunctionModularity());
    populationRDD ← sortByFitness(populationRDD);
    populationRDD ← selectPopulation(populationRDD);
    populationRDD ← crossoverPopulation(populationRDD, crossover_rate);
    populationRDD ← mutatePopulation(populationRDD, mutation_rate);
    population ← populationRDD.collect();
    generation ← generation + 1;
end
return population.getFittest();
```

3.2.4. Evaluation of the Effectiveness of the Detected Communities

The effectiveness of the detected communities can be evaluated using mainly two methods: normalised mutual information (NMI) and modularity. Normalized mutual information [61] is an evaluation measure to determine the quality of the clusters formed and to determine how accurately the community detection algorithm has performed. Normalized mutual information determines the similarity of the detected communities to the existing communities in the network. It requires the groundtruth information to evaluate a community. It is predicated on a theory known as mutual information (MI), which attempts to quantify the amount of information that is shared between two distinct random variables. The NMI is a normalised variant of the MI that takes into consideration both the overall number of data points as well as the size of the clusters. Calculating the NMI requires first calculating the mutual information that exists between the two clusters and then normalising that value by the entropy that exists in each cluster:

$$\mathrm{NMI(X,Y)} = \frac{2 \times \mathrm{MI(X,Y)}}{\mathrm{H(X)} + \mathrm{H(Y)}} \tag{8}$$

where

- X and Y are the two different clusters.
- MI(X, Y) is the mutual information between X and Y, which measures the amount of shared information.
- H(X) and H(Y) are the entropies of clusters X and Y, respectively, which measure the uncertainty or randomness in each cluster.

The mutual information is given by

$$\mathrm{MI(X,Y)} = \sum_{\mathrm{i=1}}^{|\mathrm{X}|} \sum_{\mathrm{j=1}}^{|\mathrm{Y}|} \mathrm{P(i,j)} \log \frac{\mathrm{P(i,j)}}{\mathrm{P(i)} + \mathrm{P'(j)}} \tag{9}$$

where

- P(i,j) is the probability of data occurring in cluster i (actual) and cluster j (predicted);
- P(i) is the probability of data occurring in cluster i (actual);
- P(j) is the probability of data occurring in cluster j (predicted).

The entropy is given by

$$\mathrm{H(X)} = \sum_{\mathrm{i=1}}^{|\mathrm{X}|} \mathrm{P(i)} \log \mathrm{P(i)} \tag{10}$$

$$H(Y) = \sum_{j=1}^{|Y|} P(j) \log P(j) \tag{11}$$

where

- H(X) is the actual cluster assignments.
- H(Y) is the actual cluster assignments.

Another measure used to evaluate the communities detected is modularity. Modularity gives the strength of the partitions in the network. In modularity, the groundtruth information is not needed, which means that the modularity-based community detection algorithms do not require information about the real communities in order to detect communities in a network. The equation for modularity is given in Equation (4).

3.3. Time Complexity Analysis

In this section, we analyse the time complexity of the proposed distributed genetic algorithm for community detection along with the parallel fuzzy cognitive map. The time complexity of the algorithm can be analysed by examining each step and considering the dominant factors that contribute to the overall complexity. In the given distributed genetic algorithm for community detection, modularity was used as the fitness function. To compute its time complexity, we analysed each step of the algorithm. The algorithm involved mainly two steps, the focal node identification using a parallel fuzzy cognitive map and the community detection using the distributed genetic algorithm.

We let G be a network, where n was the number of nodes, and m was the number of edges in the network. Firstly, the state vector was computed using the betweenness centrality measure for the focal node identification part, which took $O(n \times (n + m))$ for serial execution. Since a distributed approach was adopted, and the data were partitioned across PR physical nodes, the complexity was $O((n \times (n + m))/PR)$. Also, the weight matrix was distributed to the PR nodes of the network; hence, it took the complexity $O(PR)$. Subsequently, the FCM iterations were performed, which involved updating the state vector based on the weights and biases, which took $O((max\_iterations \times n^2)/PR)$. Therefore, the total time complexity of the FCM algorithm was expressed as $O(PR) + O((n \times (n + m))/PR) + O((max\_iterations \times n^2)/PR)$, and simplifying this expression, the total complexity of the FCM was $O((max\_iterations \times n^2)/PR)$.

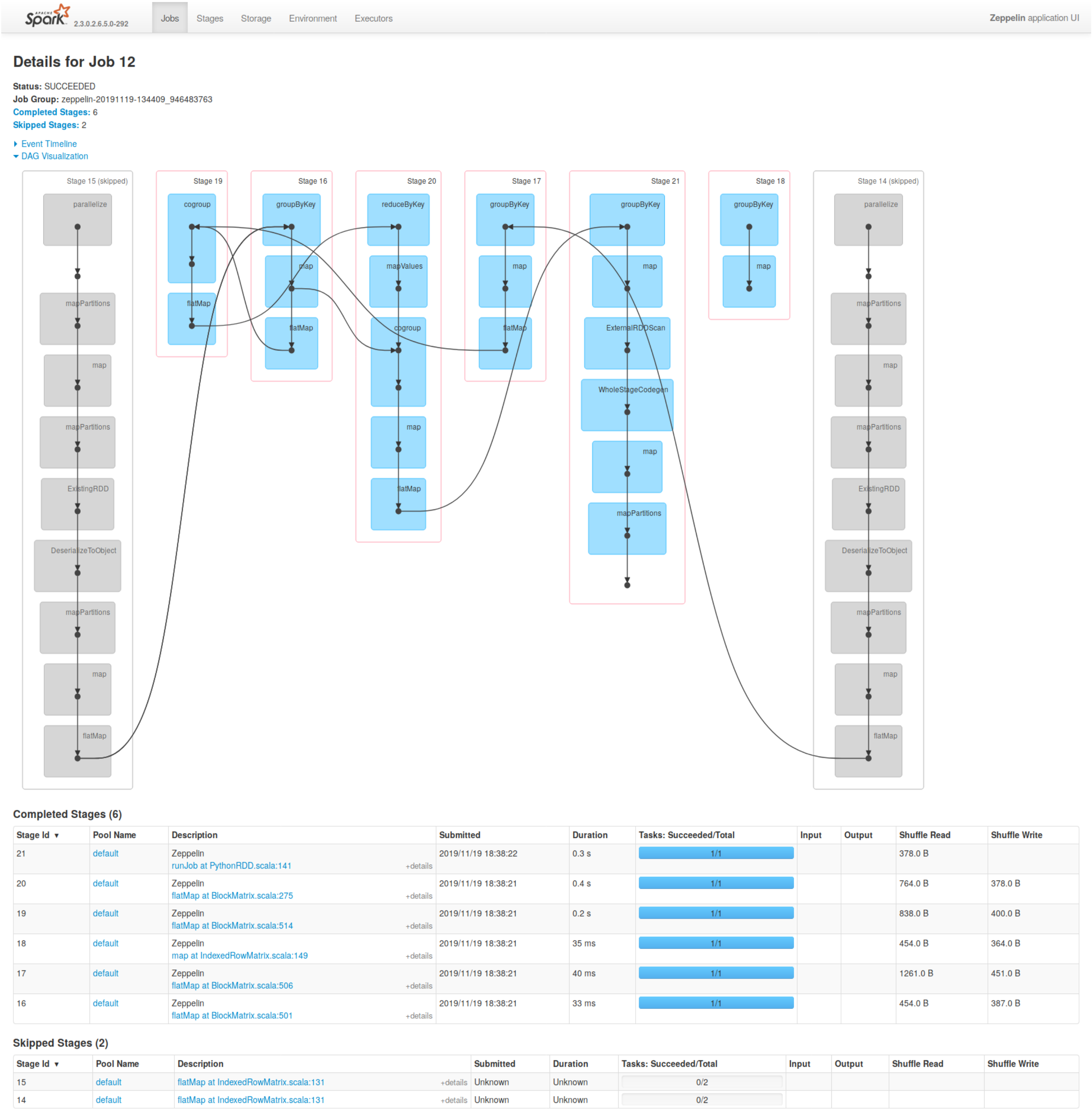
The next step was to calculate the time complexity of the distributed genetic algorithm. For the GA, the main operations that contributed to the time complexity were the evaluation of the fitness, selection, crossover, and the mutation. The first step was initialising the population, which took $O(p)$, where p was the size of the population. The fitness was evaluated for the population using modularity as the fitness function. The time complexity for the fitness function was $O(p \times n \times m)$. The selection operation had a complexity of $O(plogp)$. The crossover and mutation operations had a complexity of $O(p)$. Hence, combining these, the total complexity of the distributed GA was $O((max\_generations \times (p \times n \times m + p \times \log(p)))/PR)$, which was further simplified to $O((n \times m \times log(p))/PR)$.

4. Experimental Framework

The study was conducted leveraging a Hadoop cluster of high performance, consisting of a single name node server and two data node servers. The combined computational power of the servers was 768 GB of RAM and a 144-core processor. The cluster supported the Hortonworks Data Platform, HDP 3.0. The software used was Apache Spark 2.3.0. The Spark platform produced a directed acyclic graph that was used to track all the operations performed by the Spark engine. In Spark, a job is associated with a chain of RDD dependencies organised in a direct acyclic graph (DAG).

One of the execution DAGs produced during the proposed work is shown in Figure 5. The outer rectangle represents the different stages of the operation, the inner square boxes represent the user function calls, and the dots within the boxes represent the RDDs produced. Under the hood, this visualization illustrates a sequence of map, join, and group-

ByKey processes. Additionally, it illustrates the succession of caching operations that occur during the execution of a spark job, which accelerates the execution.



Details for Job 12

Status: SUCCEEDED
Job Group: zeppelin-20191119-134409_946483763
Completed Stages: 6
Skipped Stages: 2

▸ Event Timeline
▾ DAG Visualization

Completed Stages (6)

Stage Id ▾	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
21	default	Zeppelin runJob at PythonRDD.scala:141 +details	2019/11/19 18:38:22	0.3 s	1/1			378.0 B	
20	default	Zeppelin flatMap at BlockMatrix.scala:275 +details	2019/11/19 18:38:21	0.4 s	1/1			764.0 B	378.0 B
19	default	Zeppelin flatMap at BlockMatrix.scala:514 +details	2019/11/19 18:38:21	0.2 s	1/1			838.0 B	400.0 B
18	default	Zeppelin map at IndexedRowMatrix.scala:149 +details	2019/11/19 18:38:21	35 ms	1/1			454.0 B	364.0 B
17	default	Zeppelin flatMap at BlockMatrix.scala:506 +details	2019/11/19 18:38:21	40 ms	1/1			1261.0 B	451.0 B
16	default	Zeppelin flatMap at BlockMatrix.scala:501 +details	2019/11/19 18:38:21	33 ms	1/1			454.0 B	387.0 B

Skipped Stages (2)

Stage Id ▾	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
15	default	flatMap at IndexedRowMatrix.scala:131 +details	Unknown	Unknown	0/2				
14	default	flatMap at IndexedRowMatrix.scala:131 +details	Unknown	Unknown	0/2				

Figure 5. One of the directed acyclic graphs of the parallel FCM processing.

5. Results and Discussion

In this work, the efficiency of the proposed model was tested on 11 real-world benchmark network datasets and five synthetic network datasets.

5.1. Real-World Benchmark Network Datasets

The results obtained for the real-world benchmark datasets compared the time difference between the FCM focal-node identification in normal mode and in parallel mode, as the proportions of the network grew. The results depicted that while there was no considerable difference in the time taken to detect the focal nodes in the case of smaller networks, the time taken by the simple FCM increased exponentially as the size of the network increased, whereas the parallel FCM took just fractions of a second to process even the largest network being considered. The results are depicted in Table 1.

Table 1. Focal node identification in a normal FCM vs. the parallel FCMs.

Input Network	No. of Nodes	No. of Edges	No. of Focal Nodes	FCM	Parallel FCM
Karate club network [62]	34	78	13	300 ms	350 ms
Dolphin social network [63]	62	159	32	500 ms	600 ms
Books about U.S. politics [64]	105	441	41	574 ms	651 ms
U.S. college football [28]	115	613	50	692 ms	664 ms
Les Miserables [34]	77	254	37	887 ms	809 ms
C. Elegans metabolic Network [65]	453	2025	244	0 m 1 s	890 ms
Human protein (Figeys) [66]	2239	6452	1260	0 m 2 s	786 ms
U.S. power grid [67]	4941	6594	2427	1 m 20 s	793 ms
Pretty good privacy [65]	10,680	24,314	5040	3 m 5 s	853 ms
Cora citation [68]	23,166	91,500	12,306	10 m 15 s	903 ms
Online social network epinions [69]	75,879	508,837	25,498	1 h 48 m 25 s	941 ms

The focal nodes identified by the fuzzy cognitive map were given as input to the genetic algorithm. The proposed GA identified the community structure in the network by maximizing the modularity (Equation (4)) values. To assess the performance of the proposed model in detecting the community structure of the network, a comparison of the execution times of the distributed GA with a parallel FCM and other genetic algorithms in the literature for community detection is depicted in Figure 6.

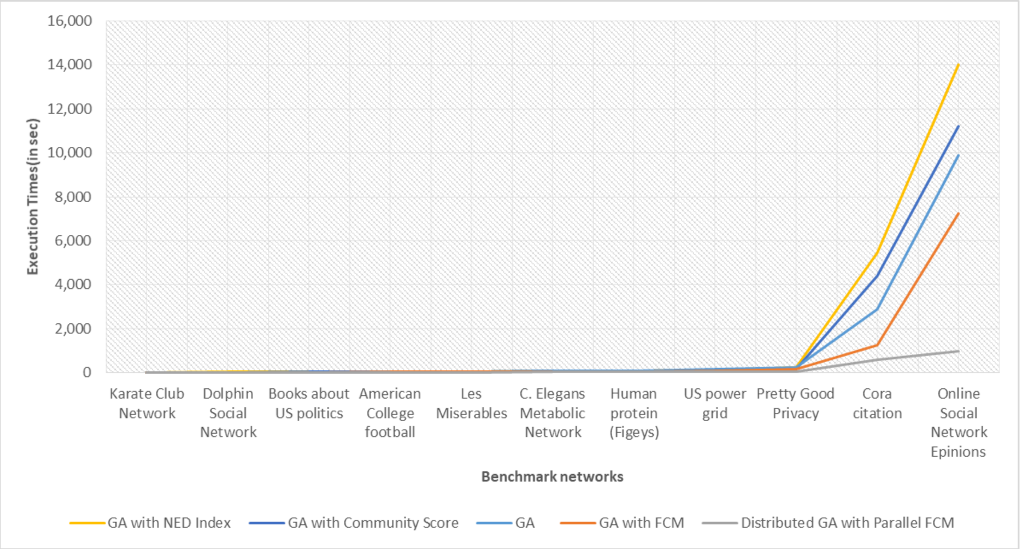


Figure 6. Comparison of the execution times of a distributed GA with a parallel FCM and other genetic algorithms for community detection.

The results show that the use of a distributed genetic algorithm with parallel FCM significantly reduced the execution time. It is visible that while there was not much improvement in the execution times when smaller networks were considered, as the network size increased, there was a massive difference between the execution times. In the case of community detection, when the size of the network was small, the use of the distributed algorithm did not yield a considerable difference in the execution time because, in smaller networks, the overhead of distributing the network across the nodes outweighed the processing time. In larger networks, the time necessary to disseminate the data was trivial in comparison to the time required for processing. It can be observed that while the time taken by other algorithms increased exponentially as the scale of the network expanded, the time taken by the distributed GA with a parallel FCM increased linearly with the size of the network. Additionally, the comparison study demonstrated that when compared to alternative techniques for calculating chromosomal fitness in a GA, such as the community score, NED index, and so on, using modularity as the fitness function produced the best outcome. Furthermore, in all the cases, we observed that the adoption of a parallel fuzzy cognitive map to determine the initial community composition, along with a distributed genetic algorithm, considerably reduced the program's overall computational time and helped it to converge faster.

5.2. Synthetic Benchmark Network Datasets

In order to assess the effectiveness of our algorithm, we also utilised the benchmark network introduced by Lancichinetti and Fortunato in 2009 [70]. In Table 2, synthetic benchmark networks with a range of nodes from 110 to 10,000 are shown. As in the case of real-world networks, the model did not show much performance improvement in the case of smaller networks due to the distribution overhead being more than the performance improvement. But as the size of the network increased, the performance of the model also increased.

Table 2. Execution time obtained for the synthetic benchmark dataset of varying community size and network size.

Input Network (LFR(N, k, maxk, mu, min_c, max_c))	No. of Nodes	No. of Edges	No. of Focal Nodes	Execution Time
(128, 10, 10, 0.1, 32, 32)	110	1024	73	370 ms
(333, 10, 16, 0.2, 10, 30)	333	2359	183	698 ms
(1500, 15, 15, 0.1, 20, 50)	1500	10,473	1764	772 ms
(5000, 20, 40, 0.1, 30, 60)	5000	25,784	2140	789 ms
(10,000, 20, 30, 0.2, 100, 200)	10,000	54,396	5647	867 ms

5.3. Accuracy Analysis of the Communities Detected

To determine the quality of the communities formed, two methods were used, normalised mutual information (NMI) and modularity. Tables 3 and 4 represent the normalized mutual information results obtained when evaluated on networks with the ground truth available, and Table 5 depicts the modularity obtained for the networks without the ground truth available.

Table 3. Normalized mutual information (NMI) obtained for real-world network datasets with the ground truth available.

Network	Proposed Method	GA with NED Index	GA with Community Score
Karate club network [62]	0.7324	0.5513	0.5426
Dolphin social network [63]	0.6507	0.5673	0.6201
Books about U.S. politics [64]	0.8311	0.7512	0.7937
American college football [28]	0.5520	0.6725	0.5844

Table 4. Normalized mutual information (NMI) obtained for the synthetic benchmark datasets with the ground truth available.

Network (LFR(N, k, maxk, mu, min_c, max_c))	Proposed Method	GA with NED Index	GA with Community Score
(128, 16, 16, 0.1, 32, 32)	1.0000	1.0000	0.9901
(333, 10, 16, 0.2, 10, 30)	0.8836	0.8532	0.8701
(1500, 15, 15, 0.1, 20, 50)	0.7530	0.7980	0.7461
(5000, 20, 40, 0.1, 30, 60)	0.7594	0.7422	0.7321
(10,000, 20, 30, 0.2, 100, 200)	0.7236	0.7254	0.7198

Table 5. Modularity obtained for real-world network datasets without the ground truth.

Network	Proposed Method	GA with NED Index	GA with Community Score
Les Miserables [34]	0.5547	0.4721	0.5211
C. Elegans metabolic network [65]	0.4724	0.4473	0.4562
Human protein (Figeys) [66]	0.6182	0.5820	0.5831
U.S. power grid [67]	0.4901	0.4546	0.4777
Pretty good privacy [65]	0.5213	0.4912	0.4623
Cora citation [68]	0.6420	0.5997	0.6232
Online social network epinions [69]	0.5604	0.5031	0.4987

5.4. Evaluation of the Time Complexity Estimates versus the Parallel Execution Time

The goal of parallelization is to execute processes in parallel by distributing the computing workload across numerous processors or computers. This makes it possible to perform multiple tasks at the same time. It has the potential to cut down on the total amount of time required for the algorithm's execution and to improve its adaptability to larger networks. However, this does not have a direct effect on the algorithm's asymptotic complexity. The complexity analysis typically considers the sequential execution of the algorithm, without accounting for parallelization. So, the theoretical complexity of the algorithm remains the same. However, parallelization can offer a practical performance improvement by exploiting the available parallel processing resources. This improvement is often reflected in reduced execution time, which is valuable for large-scale networks. Figures 7 and 8 represent the comparison of the derived time complexity and the actual running time for Lancichinetti–Fortunato–Radicchi (LFR) datasets [70] of varying sizes.

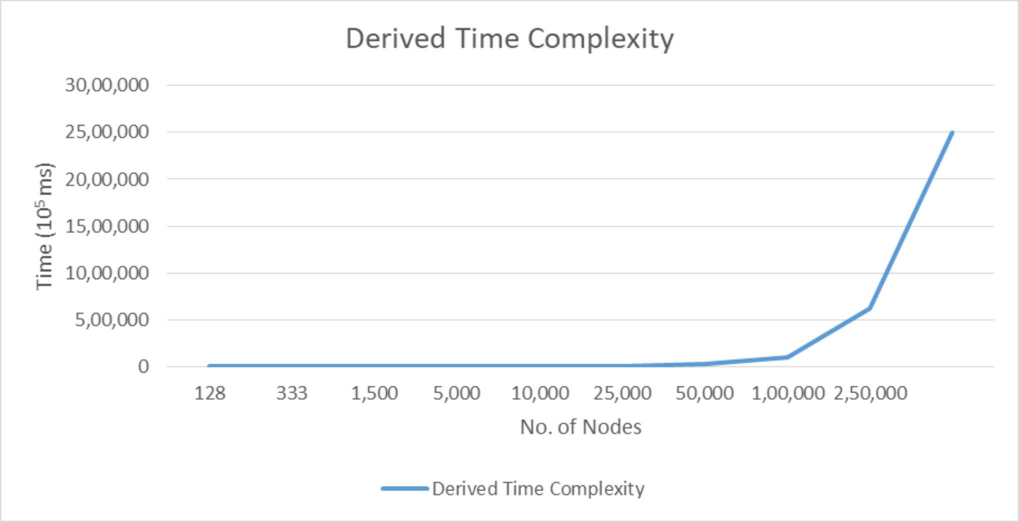


Figure 7. Derived time complexity.

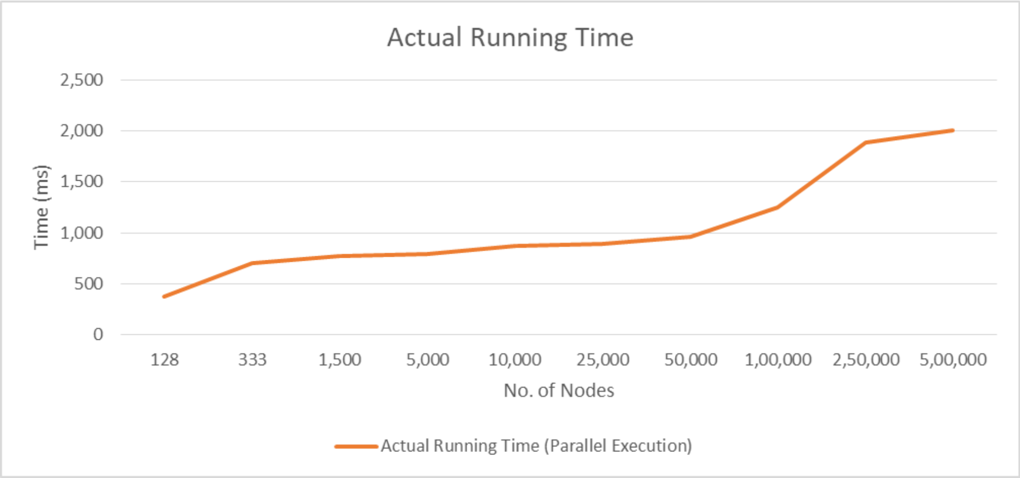


Figure 8. Actual running time.

It can be observed that when parallel execution was introduced to the proposed model, there was a considerable reduction in the execution time for each of the networks. This observation highlights the positive impact of parallelization on the overall performance of the algorithm.

6. Conclusions

This article discussed a parallel fuzzy cognitive map method for identifying the network's focal nodes. Using a distributed genetic algorithm, these focal nodes are employed to determine a community structure in the network. When finding focal nodes, the fuzzy cognitive map takes the network's centrality traits into consideration. A distributed genetic algorithm is used to discover the communities within the network that optimize the modularity of the network in order to achieve optimal solutions. On 11 different benchmark networks, the method was evaluated. The proposed model was compared to existing genetic algorithm-based community detection models to assess its performance. Also, it was evaluated on five synthetic benchmark datasets. It was discovered that combining a distributed genetic algorithm with a parallel fuzzy cognitive map considerably reduced the time required to find communities in a network. The quality of the communities produced was also evaluated using NMI and modularity measures. The results obtained are on par with the values obtained by different community detection algorithms in the literature.

Future work is oriented toward adopting the proposed parallel fuzzy cognitive map model, which is much more efficient and faster than the normal FCM, to be used by decision makers to perform various prediction and classification tasks in the cases where the size of the dataset is considerably large. Also, the possibility of adopting a genetic algorithm with a fuzzy cognitive map to tackle various problems associated with community detection in large networks such as biological networks, social networks, disease spread networks, and other weighted networks can be explored. Also, the proposed algorithm can be extended to detect overlapping communities in the network. The different possibilities where the focal nodes identified by the fuzzy cognitive map model can be used need to be inspected. Another potential avenue for future research involves enhancing the efficiency of community detection in large graphs through the parallelization of established high-quality algorithms, such as Infomap and IGLP-DP.

Author Contributions: Conceptualization, H.K. and J.M.V.; methodology, H.K., J.M.V., K.P. and E.P.; formal analysis and investigation, H.K., K.P. and J.M.V.; validation, H.K., K.P. and E.P.; writing—original draft preparation, H.K.; writing—review and editing, H.K., J.M.V., K.P. and E.P.; visualization, K.P.; supervision, J.M.V. and E.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors affirm that there are no conflict of interest.

References

1. Hou, B.; Yao, Y.; Liao, D. Identifying all-around nodes for spreading dynamics in complex networks. *Phys. A Stat. Mech. Its Appl.* **2012**, *391*, 4012–4017. [CrossRef]
2. Lü, L.; Zhang, Y.C.; Yeung, C.H.; Zhou, T. Leaders in social networks, the delicious case. *PLoS ONE* **2011**, *6*, e21202. [CrossRef] [PubMed]
3. Zhou, Y.-B.; Lü, L.; Li, M. Quantifying the influence of scientists and their publications: Distinguishing between prestige and popularity. *New J. Phys.* **2012**, *14*, 033033. [CrossRef]
4. Lü, L.; Chen, D.-B.; Zhou, T. The small world yields the most effective information spreading. *New J. Phys.* **2011**, *13*, 123005. [CrossRef]
5. Sabidussi, G. The centrality index of a graph. *Psychometrika* **1966**, *31*, 581–603. [CrossRef]
6. Freeman, L.C. Centrality in social networks conceptual clarification. *Soc. Netw.* **1978**, *1*. [CrossRef]
7. Bonacich, P. Some unique properties of eigenvector centrality. *Soc. Netw.* **2007**, *29*, 555–564. [CrossRef]
8. Brin, S.; Page, L. The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [CrossRef]
9. Li, Q.; Zhou, T.; Lü, L.; Chen, D. Identifying influential spreaders by weighted LeaderRank. *Phys. A Stat. Mech. Its Appl.* **2014**, *404*, 47–55. [CrossRef]
10. Kosko, B. Cognitive fuzzy maps. *Int. J. Man-Mach. Stud.* **1986**, *24*, 65–75. [CrossRef]
11. Axelrod, R. *Structure of Decisions: The Cognitive Maps of Political Elites*; Princeton University Press: Princeton, NJ, USA, 1976.
12. Papageorgiou, E.; Stylios, C.; Groumpos, P. An integrated two-level hierarchical system for decision making in radiation therapy based on fuzzy cognitive maps. *IEEE Trans. Biomed. Eng.* **2003**, *50*, 1326–1339. [CrossRef] [PubMed]
13. Papakostas, G.A.; Boutalis, Y.S.; Koulouriotis, D.E.; Mertzios, B.G. Fuzzy cognitive maps for pattern recognition applications. *Int. J. Pattern Recognit. Artif. Intell.* **2008**, *22*, 1461–1486. [CrossRef]
14. Zhou, X.; Zhang, H. An algorithm of text categorization based on similar rough set and fuzzy cognitive map. In Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008, Jinan, China, 18–20 October 2008.
15. Iakovidis, D.K.; Papageorgiou, E. Intuitionistic Fuzzy Cognitive Maps for Medical Decision Making. *IEEE Trans. Inf. Technol. Biomed.* **2011**, *15*, 100–107. [CrossRef] [PubMed]
16. Papageorgiou, E.I.; Kannappan, A. Fuzzy cognitive map ensemble learning paradigm to solve classification problems: Application to autism identification. *Appl. Soft Comput.* **2012**, *12*, 3798–3809. [CrossRef]
17. Salmeron, J.L. Modelling grey uncertainty with Fuzzy Grey Cognitive Maps. *Expert Syst. Appl.* **2010**, *37*, 7581–7588. [CrossRef]
18. Aguilar, J. Dynamic Random Fuzzy Cognitive Maps. *Comput. Y Sist.* **2004**, *7*, 260–271.
19. Oikonomou, P.; Papageorgiou, E.I. Particle Swarm Optimization Approach for Fuzzy Cognitive Maps Applied to Autism Classification. In *IFIP Advances in Information and Communication Technology*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 516–526. [CrossRef]
20. Nápoles, G.; Grau, I.; Bello, R.; Grau, R. Two-Steps learning of Fuzzy Cognitive Maps for prediction and knowledge discovery on the HIV-1 drug resistance. *Expert Syst. Appl.* **2014**, *41*, 821–830. [CrossRef]
21. Bourgani, E.; Stylios, C.D.; Manis, G.; Georgopoulos, V.C. Time dependent fuzzy cognitive maps for medical diagnosis. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Ioannina, Greece, 15–17 May 2014.
22. Ruan, D.; Mkrtchyan, L. Using belief degree-distributed fuzzy cognitive maps for safety culture assessment. In *Advances in Intelligent and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2011.
23. Judy, M.V.; Soman, G. Parallel Fuzzy Cognitive Map Using Evolutionary Feature Reduction for Big Data Classification Problem. In *Communications in Computer and Information Science*; Springer: Singapore, 2018; pp. 226–239. [CrossRef]
24. Youngseok, C.; Habin, L.; Zahir, I. Big data-driven fuzzy cognitive map for prioritising IT service procurement in the public sector. *Ann. Oper. Res.* **2018**, *270*, 75–104.
25. Puerto, E.; Aguilar, J.; López, C.; Chávez, D. Using Multilayer Fuzzy Cognitive Maps to diagnose Autism Spectrum Disorder. *Appl. Soft Comput.* **2019**, *75*, 58–71. [CrossRef]
26. Ramirez-Bautista, J.A.; Huerta-Ruelas, J.A.; Kóczy, L.T.; Hatwágner, M.F.; Chaparro-Cárdenas, S.L.; Hernández-Zavala, A. Classification of plantar foot alterations by fuzzy cognitive maps against multi-layer perceptron neural network. Biocybern. *Biomed. Eng.* **2020**, *40*, 404–414.
27. Gao, Y.; Yu, X.; Zhang, H. Overlapping community detection by constrained personalized PageRank. *Expert Syst. Appl.* **2021**, *173*, 114682. [CrossRef]

28. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef] [PubMed]
29. Bedi, P.; Sharma, C. Community detection in social networks. *WIREs Data Min. Knowl. Discov.* **2016**, *6*, 115–135. [CrossRef]
30. Jia, G.; Cai, Z.; Musolesi, M.; Wang, Y.; Tennant, D.A.; Weber, R.J.M.; Heath, J.K.; He, S. Community Detection in Social and Biological Networks Using Differential Evolution. In *Learning and Intelligent Optimization*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 71–85. [CrossRef]
31. Chan-Lau, J.A. Systemic centrality and systemic communities in financial networks. *Quant. Finance Econ.* **2018**, *2*, 468–496. [CrossRef]
32. Blondel, V.D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
33. Pons, P.; Latapy, M. Computing Communities in Large Networks Using Random Walks. *Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.)* **2005**, *3733*, 284–293.
34. Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef] [PubMed]
35. He, J.; Chen, D.; Sun, C. A fast simulated annealing strategy for community detection in complex networks. In Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October 2016; pp. 2380–2384.
36. Lai, D.; Lu, H.; Nardini, C. Enhanced modularity-based community detection by random walk network preprocessing. *Phys. Rev. E* **2010**, *81*, 066118. [CrossRef]
37. Wang, W.; Street, W.N. Finding Hierarchical Communities in Complex Networks Using Influence-Guided Label Propagation. In Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 14–17 November 2015; pp. 547–556. [CrossRef]
38. Tasgin, M.; Herdagdelen, A.; Bingol, H. Community Detection in Complex Networks Using Genetic Algorithms. *arXiv* **2007**, arXiv:0711.0491.
39. Mazur, P.; Zmarzłowski, K.; Orłowski, A. Genetic Algorithms Approach to Community Detection. *Acta Phys. Pol. A* **2010**, *117*, 703–705. [CrossRef]
40. Pizzuti, C. GA-Net: A Genetic Algorithm for Community Detection in Social Networks. *Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.)* **2008**, *5199*, 1081–1090. [CrossRef]
41. Guerrero, M.; Montoya, F.G.; Baños, R.; Alcayde, A.; Gil, C. Adaptive community detection in complex networks using genetic algorithms. *Neurocomputing* **2017**, *266*, 101–113. [CrossRef]
42. Pizzuti, C. Evolutionary Computation for Community Detection in Networks: A Review. *IEEE Trans. Evol. Comput.* **2018**, *22*, 464–483. [CrossRef]
43. Tasgin, M.; Bingol, H. Community Detection in Complex Networks using Genetic Algorithm. *arXiv* **2006**, arXiv:0711.0491.
44. Gog, A.; Dumitrescu, D.; Hirsbrunner, B. Community Detection in Complex Networks Using Collaborative Evolutionary Algorithms. In Proceedings of the Advances in Artificial Life: 9th European Conference, ECAL 2007, Lisbon, Portugal, 10–14 September 2007; pp. 886–894. [CrossRef]
45. He, D.; Wang, Z.; Yang, B.; Zhou, C. Genetic algorithm with ensemble learning for detecting community structure in complex networks. In Proceedings of the ICCIT 2009—4th International Conference on Computer Sciences and Convergence Information Technology, Seoul, Republic of Korea, 24–26 November 2009; pp. 702–707.
46. Gong, M.; Fu, B.; Jiao, L.; Du, H. Memetic algorithm for community detection in networks. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2011**, *84*, 056101. [CrossRef] [PubMed]
47. Gong, M.; Cai, Q.; Li, Y.; Ma, J. An improved memetic algorithm for community detection in complex networks. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June 2012; pp. 1–8. [CrossRef]
48. Jia, G.; He, S.; Zhu, Z.; Liu, J.; Tang, K. A Multimodal Optimization and Surprise Based Consensus Community Detection Algorithm. In Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, Spain, 11–15 July 2015. [CrossRef]
49. Shang, R.; Bai, J.; Jiao, L.; Jin, C. Community detection based on modularity and an improved genetic algorithm. *Phys. A Stat. Mech. Its Appl.* **2013**, *392*, 1215–1231. [CrossRef]
50. Shi, C.; Cai, Y.; Fu, D.; Dong, Y.; Wu, B. A link clustering based overlapping community detection algorithm. *Data Knowl. Eng.* **2013**, *87*, 394–404. [CrossRef]
51. Pizzuti, C. Overlapped community detection in complex networks. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, Montréal, QC, Canada, 8–12 July 2009; pp. 859–866. [CrossRef]
52. Shi, C.; Wang, Y.; Wu, B.; Zhong, C. A New Genetic Algorithm for Community Detection. In Proceedings of the Complex Sciences: First International Conference, Complex 2009, Shanghai, China, 23–25 February 2009; pp. 1298–1309. [CrossRef]
53. Jin, D.; He, D.; Liu, D.; Baquero, C. Genetic Algorithm with Local Search for Community Mining in Complex Networks. In Proceedings of the 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, Arras, France, 27–29 October 2010.
54. Liu, D.; Jin, D.; Baquero, C.; He, D.; Yang, B.; Yu, Q. Genetic Algorithm with a Local Search Strategy for Discovering Communities in Complex Networks. *Int. J. Comput. Intell. Syst.* **2013**, *6*, 354–369. [CrossRef]

55. Liu, J.-G.; Ren, Z.-M.; Guo, Q.; Wang, B.-H. Node importance ranking of complex networks. *Acta Phys. Sin.* **2013**, *62*, 178901.
56. Kitsak, M.; Gallos, L.K.; Havlin, S.; Liljeros, F.; Muchnik, L.; Stanley, H.E.; Makse, H.A. Identification of influential spreaders in complex networks. *Nat. Phys.* **2010**, *6*, 888–893. [CrossRef]
57. Kleinberg, J.M. Authoritative sources in a hyperlinked environment. In *The Structure and Dynamics of Networks*; Princeton University Press: Princeton, NJ, USA, 1999.
58. Haritha, K.; Judy, M.V. Fuzzy Cognitive Map-Based Genetic Algorithm for Community Detection. In *Progress in Advanced Computing and Intelligent Engineering*; Springer: Singapore, 2020; pp. 412–426. [CrossRef]
59. Freeman, L.C. A Set of Measures of Centrality Based on Betweenness. *Sociometry* **1977**, *40*, 35–41. [CrossRef]
60. Haritha, K.; Judy, M.V.; Papageorgiou, K.; Georgiannis, V.C.; Papageorgiou, E. Distributed Fuzzy Cognitive Maps for Feature Selection in Big Data Classification. *Algorithms* **2022**, *15*, 383. [CrossRef]
61. Danon, L.; Díaz-Guilera, A.; Duch, J.; Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, *2005*, P09008. [CrossRef]
62. Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. *Anthropol. Res.* **1977**, *33*, 452–473. [CrossRef]
63. Lusseau, D.; Schneider, K.; Boisseau, O.J.; Haase, P.; Slooten, E.; Dawson, S.M. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: Can geographic isolation explain this unique trait? *Behav. Ecol. Sociobiol.* **2003**, *54*, 396–405. [CrossRef]
64. Clauset, A.; Newman, M.E.J.; Moore, C. Finding community structure in very large networks. *Phys. Rev. E Stat. Physics Plasmas Fluids Relat. Interdiscip. Top.* **2004**, *70*, 066111. [CrossRef]
65. Duch, J.; Arenas, A. Community detection in complex networks using extremal optimization. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2005**, *72*, 027104. [CrossRef]
66. Ewing, R.M.; Chu, P.; Elisma, F.; Li, H.; Taylor, P.; Climie, S.; McBroom-Cerajewski, L.; Robinson, M.D.; O'Connor, L.; Li, M.; et al. Large-scale mapping of human protein-protein interactions by mass spectrometry. *Mol. Syst. Biol.* **2007**, *3*, 89. [CrossRef]
67. Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world' networks. *Nature* **1998**, *393*, 440–442. [CrossRef]
68. Šubelj, L.; Bajec, M. Model of complex networks based on citation dynamics. In Proceedings of the WWW 2013 Companion—Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013.
69. Richardson, M.; Agrawal, R.; Domingos, P. Trust Management for the Semantic Web. In Proceedings of the International Semantic Web Conference, Sanibel Island, FL, USA, 20–23 October 2003; pp. 351–368. [CrossRef]
70. Lancichinetti, A.; Fortunato, S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2009**, *80*, 016118. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.