



ISSN: 0067-2904

Community Detection in Modular Complex Networks Using an Improved Particle Swarm Optimization Algorithm

Dhuha Abdulhadi Abduljabbar

Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Received: 30/4/2022

Accepted: 7/11/2022

Published: 30/8/2023

Abstract

Community detection is an important and interesting topic for better understanding and analyzing complex network structures. Detecting hidden partitions in complex networks is proven to be an NP-hard problem that may not be accurately resolved using traditional methods. So it is solved using evolutionary computation methods and modeled in the literature as an optimization problem. In recent years, many researchers have directed their research efforts toward addressing the problem of community structure detection by developing different algorithms and making use of single-objective optimization methods. In this study, we have continued that research line by improving the Particle Swarm Optimization (PSO) algorithm using a local improvement operator to effectively discover community structure in the modular complex networks when employing the modularity density metric as a single-objective function. The framework of the proposed algorithm consists of three main steps: an initialization strategy, a movement strategy based on perturbation genetic operators, and an improvement operator. The key idea behind the improvement operator is to determine and reassign the complex network nodes that are located in the wrong communities if the majority of their topological links do not belong to their current communities, making it appear that these nodes belong to another community. The performance of the proposed algorithm has been tested and evaluated when applied to publicly-available modular complex networks generated using a flexible and simple benchmark generator. The experimental results showed the effectiveness of the suggested method in discovering community structure over modular networks of different complexities and sizes.

Keywords: Community detection, Modular networks, Particle swarm optimization algorithm, Solution improvement operator.

تحديد المجتمعات في الشبكات المعقدة القياسية باستخدام خوارزمية سرب الجسيمات المحسنة

ضحى عبد الهادي عبد الجبار

قسم علوم الحاسوب، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

يعد تحديد المجتمعات موضوعاً مهماً ومثيراً للاهتمام لفهم وتحليل أفضل لبنية الشبكات المعقدة. ثبت أن الكشف عن الأقسام المخفية في الشبكات المعقدة يمثل مشكلة صعبة للغاية غير حتمية متعددة الحدود (NP-hard)، والتي قد لا يتم حلها بدقة باستخدام الطرق التقليدية. لذلك تم حلها باستخدام طرق الحساب التطوري،

و تصميمها في الأدبيات كمشكلة تحسين. في السنوات الأخيرة، وجه العديد من الباحثين جهودهم البحثية لمعالجة مشكلة اكتشاف بنية المجتمع من خلال تطوير خوارزميات مختلفة والاستفادة من طرق التحسين احادية الهدف. في هذه الدراسة، واصلنا خط البحث هذا من خلال تحسين أداء خوارزمية سرب الجسيمات (PSO) باستعمال عامل تحسين محلي لاكتشاف بنية المجتمع بشكل فعال في الشبكات المعقدة القياسية عند استعمال مقياس كثافة الوحدات النمطية كدالة احادية الهدف. يتكون إطار الخوارزمية المقترحة من ثلاث خطوات رئيسية؛ استراتيجية التهيئة، استراتيجية البحث التي تستند إلى العوامل الجينية المضطربة، وعامل التحسين. تتمثل الفكرة الرئيسية لعامل التحسين في تحديد وإعادة تعيين عقد الشبكة المعقدة الموجودة في المجتمعات الخطأ في حالة أن غالبية روابطهم الطوبولوجية لا تنتمي إلى مجتمعاتهم الحالية، مما يجعل الأمر يبدو أن هذه العقد تنتمي إلى مجتمع آخر. تم اختبار وتقييم أداء الخوارزمية المقترحة عند تطبيقها على الشبكات المعقدة القياسية التي تم إنشاؤها باستعمال مولد قياسي مرن وبسيط. أظهرت النتائج التجريبية فاعلية الطريقة المقترحة في الكشف عن بنية المجتمع عبر الشبكات القياسية ذات التعقيدات والأحجام المختلفة.

1. Introduction

In the last decade, analyzing complex networks witnessed great interest since many complex systems that are present today, like social networks, collaboration networks, metabolic networks, neural networks, technological networks, and also political election networks, could be embodied and modeled as complex networks. Mathematically, a graph is considered an efficient way, and is often used in practice, to represent a complex network, where generally the graph nodes correspond to the objects of the complex network and the graph edges correspond to the connections between these objects. A key feature of most, if not all, complex networks is the community structure; based on that, research work related to the discovery of the hidden complex community structure has received the attention of a large number of investigators and researchers from various scientific disciplines. Informally speaking, the main function of network clustering (or detecting the community structure in a complex network) is to divide the entities of a complex network into a number of groups based on two fundamental conditions, namely, that the connections or links between entities in one cluster are dense while the links between various clusters are sparse. This is called a cluster, module, or the well-known common term, community [1-4]. In sum, research based on network analysis and partitioning it into clusters is necessary to comprehend its organization and determine its functions. Different technologies have been customized and developed for clustering networks over the past decade; some recent surveys can be found in [5, 6, 7].

In most of the proposed studies, the problem of detecting community structure has been addressed as an optimization problem in terms of maximization or minimization of a specific objective function. The main purpose of developing optimization-based community detection algorithms is to try to find the optimal solution to the relevant problem, and this mainly depends on the adopted objective function formula and evolutionary operators. Oftentimes, optimizing this objective function is difficult and is known in the literature as an NP-hard problem. Therefore, numerous studies have been suggested based on adopting diverse metaheuristic algorithms like GA (Genetic Algorithm), MA (Memetic Algorithm), PSO (Particle Swarm Optimization) algorithm, and ACO (Ant Colony Optimization) algorithm [8-12, 3].

Most community detection studies, such as those by Cao et al. and Liu et al. [13, 14], were developed based on metaheuristic algorithms by optimizing the well-known single-objective function, namely, modularity (Q), that was introduced by Newman and Girvan in 2004 [15]. In 2007, Fortunato and Barthélemy [16] found that the community structure identified by the community detection methods based on optimizing the Q function is large and that these methods may be unsuccessful at detecting tiny community structures, which leads to the known

resolution limit problem. To avoid this problem, Li et al. [17] designed a modularity density (D) function, and for exploring the structure of complex network communities at different resolution levels, they developed a generic modularity density (D_λ) by adding a special parameter (λ) to the D function. The general modularity density (D_λ) represents the sum of the average Q metric's degree of the hidden communities in the complex networks [18]. The single-objective community detection methods have proven effective in addressing the problem in both synthetic and real-world complex networks, most of which are based on evolutionary algorithms [8, 18- 20].

The particle swarm optimization (PSO) algorithm is another well-regarded metaheuristic method that was initially suggested to tackle single-objective continuous optimization problems [21, 22]. PSO appoints a set of particles that explore the solutions by moving locally and globally in the search landscape to identify the optimal solution. The movement strategy of the employed particles is inspired by the movement mechanism of a bird swarm, where each particle saves its coordinate path in the search space and correlates with the best captured local and global solution (i.e., local and global optima) to the swarm [3,23,24]. To identify the best solution, the movement of each particle will direct toward the obtained local solution as well as the global solution. Thanks to PSO's effectiveness in resolving different continuous optimization problems [3].

In order to address the problem of community structure detection more efficiently using the PSO algorithm, in this paper we have developed a framework called IPSO-Net (Improved PSO for community detection in a modular complex network) that integrates a framework of the Particle Swarm Optimization (PSO) algorithm introduced in 2018 by Abdollahpouri *et al.* [25] with an improvement operator introduced in 2019 by Moradi and Parsa [8], which relies on identifying and resetting the complex network nodes that appear to have been mapped into the wrong communities. The key characteristics of employing the PSO algorithm compared to the other existing methods are that it has a quick convergence speed with uncomplicated implementation and, moreover, a large number of different variants [3]. The proposed method (IPSO-Net) has employed the general modularity density as a fitness function and perturbation operators in terms of crossover and mutation within the PSO framework to discover communities in networks. For assessing its performance, several systematic experiments have been done on modular networks with different sizes and complexities. The obtained results showed that the integration between the PSO algorithm and the improvement operator has a positive effect and significantly enhances the performance of the PSO algorithm in terms of convergence reliability.

The remaining sections of the paper are organized as follows: Section 2 provides a brief overview of the relevant proposed works. Section (3) presents a detailed description of the proposed PSO algorithm in terms of the adopted fitness function (i.e., general modularity density optimization model) and perturbation operators, as well as a detailed explication of the solution improvement operator. In Section (4), the settings of the experiments are presented in terms of the dataset used, standard evaluation metrics, the setting of PSO parameters, and the experimental results. Finally, Section 5 summarizes the research work and presents some conclusions and future work.

2. Literature Review

Recently, many research efforts have been made to adopt the PSO algorithm to capture hidden complex community structures in different types of networks [25- 30]. Abdollahpouri et al. [25] suggested a novel method, called PSO-Net, for community detection based on a new version of the PSO framework. The proposed algorithm selected the modularity function as an

objective function. Moreover, PSO-Net changed the particles' moving strategy by applying a crossover operator between each particle and its personal best location and the global best location over the whole swarm. After that, the 1-point neighbor mutation operator was applied to avoid falling into a local optimal situation. Experiments confirmed the effectiveness of the proposed algorithm PSO-Net in discovering communities over real and synthetic networks.

Cai *et al.* [26] proposed Q-PSO, a new algorithm based on the modularity function, to accurately and effectively detect community structure in several representative complex networks and synthetic benchmark (LFR) networks. Chen *et al.* [27] put forward a novel algorithm, P-PSO (particle swarm optimization based on the Physarum model), for detecting communities by combining the computational power of a type of slime called Physarum. The P-PSO algorithm improved the effectiveness of PSO by identifying the outer edges of communities based on a Physarum-inspired network model.

Cai *et al.* [28] proposed a greedy discrete PSO algorithm to detect community structure in large-complex social networks. The statuses of particles were redefined based on a discrete scenario; and based on network topology, the status update rules were reconsidered. In addition, a greedy strategy is introduced to guide particles into a promising area. Shi *et al.* [29] suggested a novel method based on PSO to discover complex community structures by applying the modularity model as an optimization function. Initially, an enhanced spectral method was employed to represent community detection as a cluster problem, and the weighted distance that combines eigenvectors and eigenvalues was developed to measure the difference between two nodes. Xiaodong *et al.* [30] proposed a new detection model based on PSO to discover complex web communities within the network without previous knowledge about domain information.

In order to effectively detect community structure in complex networks and guide the particles' movement towards optimal regions when employing the modularity density metric as an objective function, we have continued this line by improving the Particle Swarm Optimization (PSO) algorithm performance introduced by Abdollahpouri *et al.* [25] using a local improvement operator introduced in 2019 by Moradi and Parsa [8].

3. Material and Methods

In this section, the proposed IPSO-Net method is described in detail. The framework of the IPSO-Net method consists of three main steps: initialization strategy (i.e., particle structure representation scheme and fitness computation), movement strategy (i.e., search strategy), and an improvement operator that is developed based on identifying and resetting complex network nodes that seem to belong to other communities. The flowchart of the suggested PSO algorithm for determining the community structure in modular complex networks is shown in Figure 1. A detailed explanation of each of the above steps is provided in the next sub-section.

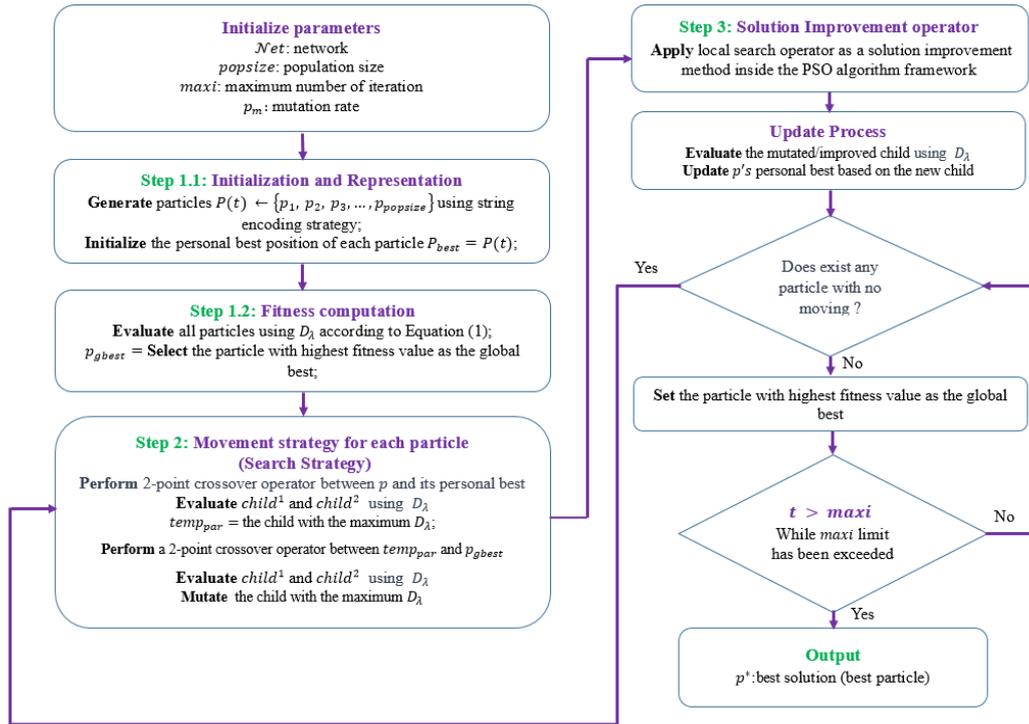


Figure 1: The flowchart of the suggested IPSO-Net algorithm. The algorithm is based on the PSO version of [25]

3.1 Particle Structure Representation Scheme and Initialization Process

The proposed IPSO-Net algorithm utilizes the string encoding strategy as its representation scheme [31]. By using the string encoding strategy, network partitions are encoded as an integer string $Net = \{c^1, c^2, \dots, c^n\}$, where n indicates the number of network vertices, while c^i denotes the integer cluster identifier for the vertex (v_i) , and its values range from 1 to n . In order to accelerate the convergence of the proposed optimization algorithm, it was applied with a biased initialization and not with a fully random initialization. Practically, we have randomly chosen a vertex (v_i) and assigned its cluster identifier (c^i) to all of its neighbors [32]. This process was performed for each particle αn times when initializing the population with α set to 0.3 in this paper.

3.2 Fitness Computation

In this study, the general modularity density (D_λ) has been employed as an objective function in IPSO-Net to obtain the community structure at a different resolution of the complex networks. Given an undirected complex network, $Net = (V, E)$, where V represents the vertices set and E represents the edges (or connections) set. One way to represent a complex network (Net) is to define an adjacent binary matrix $A_{n \times n}$, where n denotes the number of the network vertices, such that A_{ij} element is equal to 1 when there is an existing connection (or an edge) between vertices v_i and v_j , otherwise A_{ij} element is equal 0. Let's assume that V_i and V_j are the vertices sets of sub-networks C_i and C_j , respectively, then $E(V_i, V_j) = \sum_{v_x \in V_i, v_y \in V_j} A_{xy}$ points to the number of edges between C_i and C_j , $E(V_i, V_i) = \sum_{v_x, v_y \in V_i} A_{xy}$ points to the internal degree of C_i , and $E(V_i, \bar{V}_i) = \sum_{v_x \in V_i, v_y \in \bar{V}_i} A_{xy}$ points to the external degree of C_i wherein $\bar{V}_i = V - V_i$. Given k sub-networks $C_1 (V_1, E_1), \dots, C_k (V_k, E_k)$ of a complex network (Net) provided by a particle (p) , the objective function (general modularity density D_λ) can be defined as:

$$D_\lambda = \sum_{i=1}^k d(C_i) = \sum_{i=1}^k \frac{2\lambda E(V_i, V_i) - 2(1-\lambda)E(V_i, \bar{V}_i)}{|V_i|} \quad (1)$$

Where $d(C_i)$ points to the variance between the average internal degree $\frac{E(V_i, V_i)}{|V_i|}$ and average external degree $\frac{E(V_i, \bar{V}_i)}{|V_i|}$ of C_i [18].

To identify the best global particle in the swarm, the particles are sorted in descending order based on their fitness value, and the particle with the highest fitness value is chosen to be the best global solution. Obtaining a high fitness value means detecting a high-quality community structure with dense connections in the complex network. To explore the complex network topology at different resolutions, the parameter λ is employed. If the parameter λ is equal to 0 then D_λ will tend to aggregate the network into large communities. While if λ is equal to 1, D_λ will tend to aggregate the network into small communities, and when λ is equal to 0.5, D_λ will perform equivalently to the modularity density function [17, 18, 33].

3.3 Movement Strategy (Search Strategy)

PSO's search strategy depends on the mechanism of moving the particles towards their best local position while also moving them towards the best global position in the swarm. To guide the movement of each particle to the optimal possible positions, perturbation genetic operators (like crossover and mutation) are used. Below, the movement strategy steps of IPSO-Net are demonstrated in detail [3, 25].

3.3.1 Moving towards the best personal (local) position

At first, a 2-point crossover operator is performed for each particle along with its best personal (local) position. Accordingly, two new solutions are obtained as a result of applying the crossover operator. Then, the obtained results are compared, and the particle (or solution) with the highest fitness value is chosen to be a temporary position of the present particle. Figure 2 illustrates two examples of a 2-point crossover operator. Figures 2 (a) and (b) show, respectively, two random solutions representing the parents p^1 and p^2 with their respective topological community structures.

In Figure 2(c), two arbitrary points, $x = 5$ and $y = 7$ are selected. After that, the 1st child is produced by copying the cluster identifier from the beginning of parent p^1 to point x , the portion from x to y is cloned from parent p^2 and the cluster identifier of the remaining set of nodes is cloned from parent p^1 . while the second child is produced by doing the previous action in reverse order. Figure 2(d) shows the string encoded representation of the first child with its related graphical division.

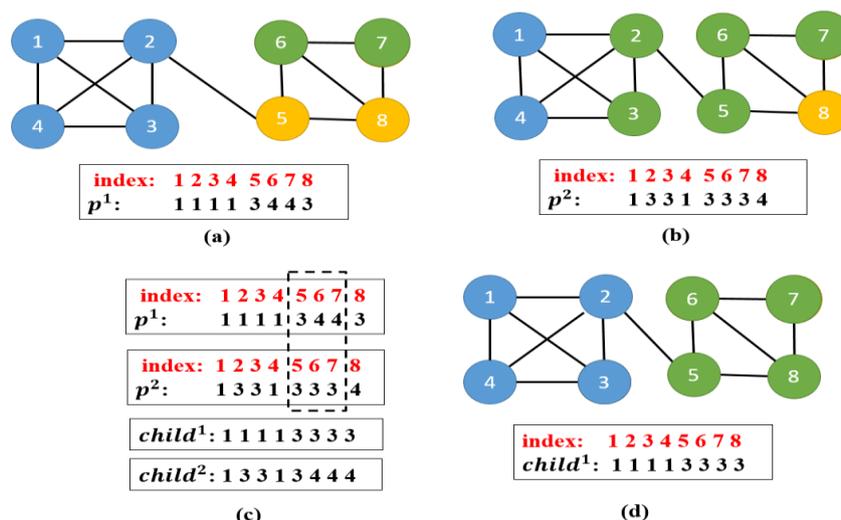


Figure 2: An illustration of a 2-point crossover. (a) p^1 and its respective topological community structure. (b) p^2 and its respective topological community structure. (c) A random 2-point crossover between p^1 and p^2 yields the children $child^1$ and $child^2$. (d) $child^1$ and its respective topological community structure

3.3.2 Moving towards the best global position in the swarm

After each particle is moved towards its personal (local) best position, it will also move towards the best global position in the swarm. To achieve this, a 2-point crossover operator is performed between the particle's temporary position (that obtained from the previous subsection) and the best global position defined. In this respect, two new solutions are also obtained and compared together, such as the former crossover operator, in order to apply the mutation operator to the best selected particle.

3.3.3 Mutation

Lastly, the particles are mutated over the entire search landscape using the 1-point mutation operator. Under the predetermined probability of the mutation operator, a random node from the given particle (p_i) is picked, and its cluster identifier is altered by a new possible cluster identifier of its adjoining nodes in order to ensure that only possible solutions are generated [3]. The output of the mutation operator for the p_i particle is p'_i which is compared with its personal best ($p_{i,best}$). If the fitness value of (p'_i) outperforms the fitness value of ($p_{i,best}$) then $p_{i,best}$ is substituted by p'_i , otherwise, the $p_{i,best}$ remains unchanged. When all the particles have moved and their personal best positions have been updated, fitness values are then computed again using the general modularity density measure, and the particle with the highest fitness value is chosen to be the best global position of the entire swarm. The above process is repeated until the predetermined number of iterations has been reached.

In order to enhance the proposed algorithm's performance and exploit the available knowledge about the problem, a local solution improvement operator is proposed whose main idea is to identify and reset the complex network nodes that seem to belong to other communities in case the majority of their topological links do not belong to their current communities. The details of the solution improvement operator are shown in the next section.

3.3.4 Solution Improvement Operator

When looking closely at the detected partitions for a given complex network, we can note that there are some nodes located in the wrong communities when the majority of their connections do not belong to their current communities and seem to belong to another community. Generally, a community within a network represents a set of closely linked nodes

whose number of internal connections is greater than the number of their connections with the rest of the network's nodes in other communities. To this end, each node is assigned a computed corresponding value. This value is calculated for a given node (i) by counting the number of its links whose targets do not belong to its current community. Accordingly, the nodes with relatively higher correspondence values are mutated into the new community. Moradi and Parsa [8] in 2019 proposed the above solution improvement method as a local search operator inside the genetic algorithm framework with locus encoding strategy representation. This local search operator decreases the inter-connections to discover high-quality clusters in the complex network. It is a very helpful operator that speeds up the population convergence and enhances the accuracy of the detected communities. Here, we have adopted Moradi and Parsa's local search operator as a solution improvement method inside the PSO algorithm framework with the string encoding strategy representation. The complete pseudo-code of the IPSO-Net algorithm for community detection is depicted in Algorithm 1, including the subprogram of the solution improvement operator.

Algorithm 1. IPSO-Net

Input : Net : network, $popsiz$: Population size, p_m : Mutation rate, $maxi$: Maximum number of iteration.

Output : p^* : Best solution (best particle).

Step 1: Initialization and Representation

$t \leftarrow 1$;

Generate $P(t) \leftarrow \{p_1, p_2, p_3, \dots, p_{popsiz}\}$ using string encoding strategy;

Initialize the personal best position of each particle $P_{best} = P(t)$;

Evaluate all particles using D_λ according to Equation (1);

$p_{gbest} = \mathbf{Select}$ the particle with highest fitness value as the global best;

Step 2: Movement strategy for each particle (i.e., Search Strategy)

for $i \leftarrow 1$ **to** $popsiz$ **do**

[$child^1, child^2$] = Perform a 2-point crossover operator between p_i and $p_{i,best}$;

Evaluate $child^1$ and $child^2$ using D_λ according to Equation (1);

$temp_{par}$ = the child with the highest fitness;

[$child^1, child^2$] = Perform a 2-point crossover operator between $temp_{par}$ and p_{gbest} ;

Evaluate $child^1$ and $child^2$ using D_λ according to Equation (1);

p_i = the child with the highest fitness value;

$p'_i = \mathbf{Mutate}$ p_i under p_m probability / **Apply Solution Improvement operator** (p_i);

Evaluate p'_i ;

if fitness value(p'_i) > fitness value($p_{i,best}$) **then** $p_{i,best} = p'_i$;

End for

p_{gbest} = the particle with the highest fitness value;

if $t \leq maxi$ **then** $t = t + 1$ and go to **Step 2**;

else $p^* = p_{gbest}$ and **Stop** the algorithm.

Return p^* .

// Subprogram: Solution improvement operator

Input : p : Solution (Particle).

Output : p' : Improvement Solution (Particle).

n = number of nodes in p ;

```

 $p' = p;$ 
for  $i = 1$  to  $n$  do
     $Source\_Community \leftarrow p(i);$ 
     $M(i) =$  The number of links of Node ( $i$ ) which their targets are not
     $Source\_Community (Node (i));$ 
end for
 $ti = \mathit{find}$  a node with maximum out community links using  $M$ ;
 $C_y \leftarrow Source\_Community (ti);$ 
// update the particle
 $p'(ti) = C_y;$ 
Return  $p'$ .

```

4. Experimental Results

In this section, we have described in detail the settings of the experiments in terms of presenting the dataset used, standard evaluation metrics, setting PSO parameters, and discussing the results obtained from the test experiments.

4.1 Dataset

This study has made use of the recently publicly available modular networks for validating community detection algorithms. These networks have been generated using a flexible and simple benchmark generator, called FARZ, introduced by Fagnan et al. in 2018 [34]. The FARZ model is similar to LFR, generating complex networks with built-in community structure that can be used as ground truth, which is ideal for validating the performance of community detection algorithms. FARZ generates dependable networks in the sense that it creates communities and networks that are characteristically similar to those of real-world networks. It is also composed of intuitive parameters with meaningful interpretation and is easy to tune for direct control of the generated networks' properties. There are 3 input parameters in FARZ (n ; m ; k) which are respectively responsible for the determination of the number of nodes, the average degree, and the number of communities. There are also 4 intuitive control parameters in FARZ, β , α , γ , and \emptyset which are responsible for controlling the community structure strength, the clustering coefficient, the degree correlation, and the distribution of the community size. In the next sections, these networks will be referred to by their corresponding control parameter, β as $FARZ_Net_\beta$ [34], since β is the responsible parameter for controlling the strength of community structure. Accordingly, the values of β in this study have been set over a range of 0.0 to 0.8 with a step of 0.05, in which diverse arbitrary complex networks of different sizes (from 50 to 250 nodes) were generated and modeled, and each network is composed of four communities. The parameters used when generating FARZ networks are listed in Table 1.

Table 1: Summary of parameters used to generate modular-benchmark networks ($FARZ_Net_\beta$)

General Parameter	Definition	Values
n	Number of nodes	50, 100, 150 and 250

Main parameter	m	Number of edges created per node. It determines the density of the generated networks	5
	k	The number of built-in communities in the generated networks	4
Control parameter	$beta (\beta)$	The strength of community structure	From 0.0 to 0.8 with 0.05 increments
	$alpha (\alpha)$	The strength of common neighbour's effect on edge formation edges	0.5
	$gama (\gamma)$	The strength of degree similarity effect on edge formation. It shows whether the degree correlation in the generated network is positive or negative.	0.5
Config. parameters	$phi (\phi)$	The constant added to all community sizes. It is responsible for moving the community sizes distribution form heavy tail to uniform	1
	$epsilon (\epsilon)$	The probability of noisy/random edges	1e-07
Overlap parameter	r	The maximum number of communities each node can belong to	1

4.2 Evaluation Measures

In this paper, we have used both the modularity (Q) and normalized mutual information (NMI) measures to assess the quality of the obtained complex community structures. NMI [35, 36] is a criterion for measuring the similarity between the community structure resulting from the proposed algorithm and the real complex community structure of a given network. Let, $\mathcal{C}_A = \{C_A^1, \dots, C_A^T\}$ represent the real clusters of a given complex network, and $\mathcal{C}_B = \{C_B^1, \dots, C_B^D\}$ represent the obtained clusters by the proposed algorithm, wherein T and D denote, respectively, the number of communities present in the partitions \mathcal{C}_A and \mathcal{C}_B . To calculate the NMI measure, first, we formed a confusion matrix $W = [W_{ij}]$, $i = 1, 2, \dots, T$ and $j = 1, 2, \dots, D$, where W_{ij} represents the number of nodes that appear in the community $C_A^i \in \mathcal{C}_A$ and also in the community $C_B^j \in \mathcal{C}_B$. Accordingly, $NMI(\mathcal{C}_A, \mathcal{C}_B)$ can be defined as :

$$NMI(\mathcal{C}_A, \mathcal{C}_B) = \frac{-2 \sum_{i=1}^T \sum_{j=1}^D W_{ij} \log \left(W_{ij} * \frac{n}{W_i W_j} \right)}{\sum_{i=1}^T W_i \log \left(\frac{W_i}{n} \right) + \sum_{j=1}^D W_j \log \left(\frac{W_j}{n} \right)} \quad (2)$$

where W_i , and W_j denote the sum of the elements of W , over the row (i) and the column (j), respectively. As mentioned earlier, n represents the total number of nodes in the network. The NMI value ranges from 0 to 1, i.e., when $NMI = 1$ this indicates that \mathcal{C}_A and \mathcal{C}_B are exactly equivalent, and when $NMI = 0$ this means \mathcal{C}_A and \mathcal{C}_B are totally different.

Modularity (Q) represents the most common internal quality measure proposed by Newman and Girvan in 2004 [15] and has been used basically to evaluate the predicted solutions when the real partitions are unknown. Q measures a fraction of edges that fall within communities, minus what is expected if the edges are randomly placed. It was observed that the Q value would approach a minimum value, i.e., 0, if the number of internal connections was similar to the random distribution. On the other hand, Q approaches the maximum value, i.e., 1, and deviates from the null case when all detected communities have dense intraconnections. This means that a network with strong community structures presents a high Q value. The modularity is defined as:

$$Q(\mathcal{C}) = \sum_{i=1}^K \left[\frac{2|E(C_i)|}{|E(\mathcal{C})|} - \left(\frac{\sum_{v \in C_i} deg(v)}{2|E(\mathcal{C})|} \right)^2 \right] \quad (3)$$

4.3 Parameter Setting

The IPSO-Net algorithm was implemented in Matlab R2016b. The experiments over modular networks have been performed on a computer having an Intel® Core™ i7 CPU @ 2.80 GHz and 16.0 GB (15.9 GB usable) of memory. In this paper, the number of iterations is customized to 100, the population size was set to 100, and the probability of a 1-point mutation operator is set to 0.3. In addition, we have investigated the impact of the λ parameter which varies from 0.3 to 0.7 with intervals of 0.2. All the experiments' results were reported by considering the average of 10 independent runs.

4.4 Experimental Results on Modular Networks and Discussions

To show the effectiveness of the proposed method, several experiments have been done over modular complex networks of different sizes and complexities. The performance of the IPSO-Net algorithm has been analyzed and compared for two different variants: IPSO-Net-1 and IPSO-Net-2, where IPSO-Net-1 represents the origin PSO version without the solution improvement operator, and IPSO-Net-2 represents the PSO version with the solution improvement operator. First, we discuss parameter λ on all $FARZ_Net_\beta$ benchmarks when $\beta \in [0.0, 0.8]$ with step size 0.05, $n = \{50, 100, 150, 250\}$ using IPSO-Net-1, and the effect of λ in terms of NMI is presented in Figures 3 and 4.

As community structure tends to be stronger with the increase of β , NMI increases, but it is still difficult to capture the correct partitions for community detection methods. However, IPSO-Net-1 could not find the correct partitions of the networks in all test cases. It should be noted that IPSO-Net-1 has obtained the best partitions when $\beta = 0.8$ and $\lambda = 0.7$, where NMI begins to increase with the further increase of β . Given a specified β , NMI increases with the increase of λ . The reason is that IPSO-Net-1 with λ less than 0.7 tends to provide large communities, leading to a decrease in NMI . Since IPSO-Net-1 with λ equal to 0.7 can result in the best results, we set $\lambda = 0.7$ in this study.

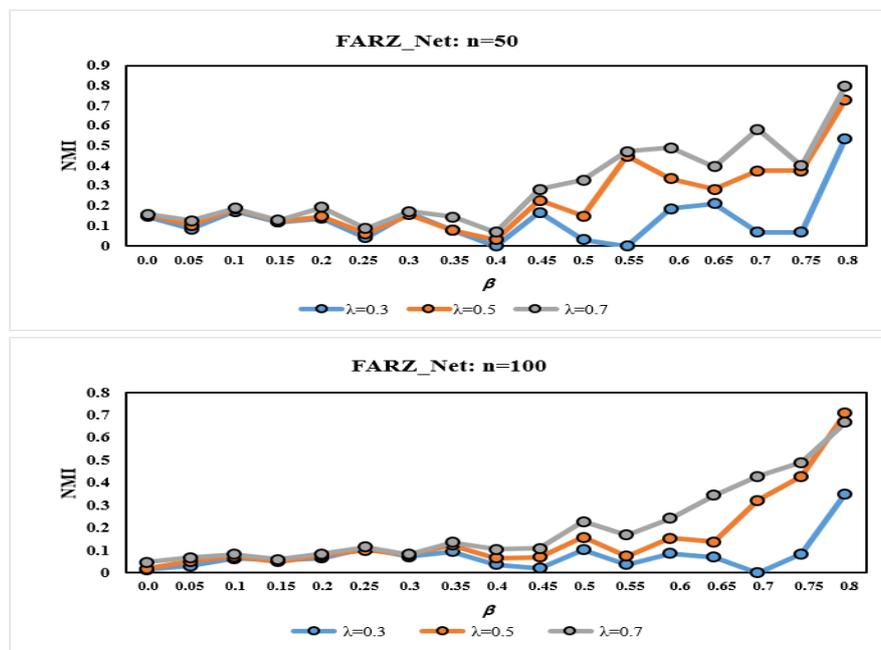


Figure 3: The effect of λ on $FARZ_Net_{\beta \in [0.0, 0.8]}$ benchmark when $n = \{50, 100\}$ for parameter analysis

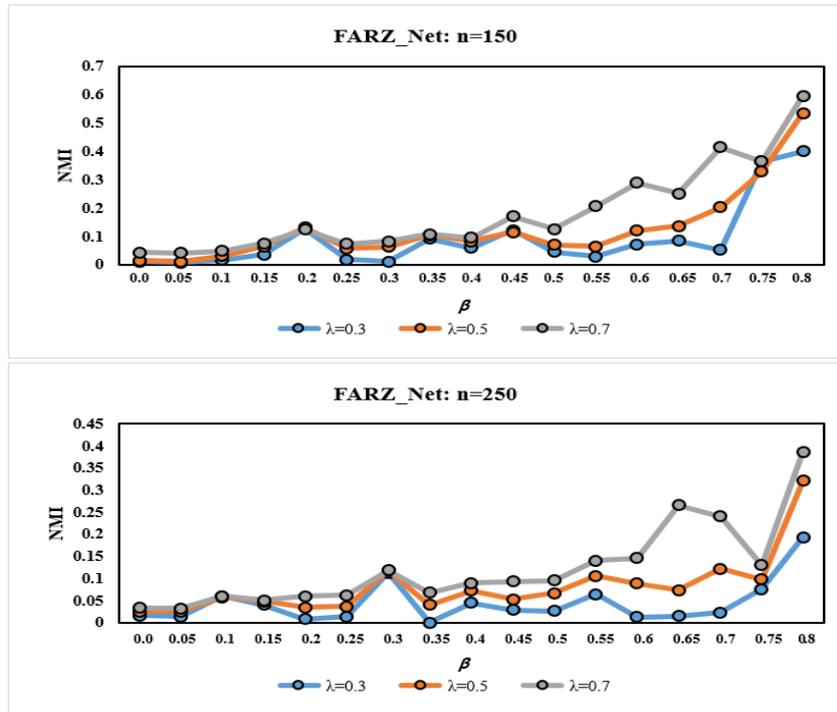


Figure 4: The effect of λ on $FARZ_Net_{\beta \in [0.0, 0.8]}$ benchmark when $n = \{150, 250\}$ for parameter analysis

To illustrate the impact of the developed solution improvement operator on the IPSO-Net performance, the result of the systematic experiments have been shown in terms of the best average of both NMI (NMI_{avg}) and Q (Q_{avg}) saved in the archive over 10 different individual runs for each complex modular network. To be fair, the same set of parameters is used for the two variants. Figures 5 and 6 present the obtained experimental results. It can be seen from Figures 5 and 6 (left side) that the NMI values obtained by IPSO-Net-2 are much higher than IPSO-Net-1 (except in a few limited cases). This indicates that the solution improvement operator is effective for enhancing the detection of community structure in the complex modular networks in terms of NMI_{avg} values.

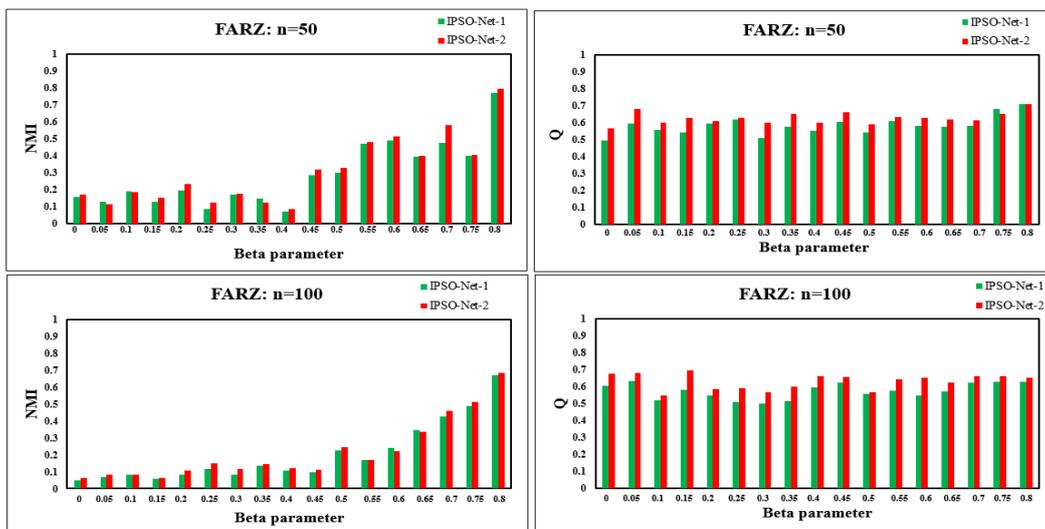


Figure 5: Results of the IPSO-Net at two different variants (IPSO-Net-1, IPSO-Net-2) on $FARZ_{Net}_{\beta \in [0,0.8]}$ benchmark with $n = (50, 100)$ in terms of NMI_{avg} (left side) and Q_{avg} (right side)

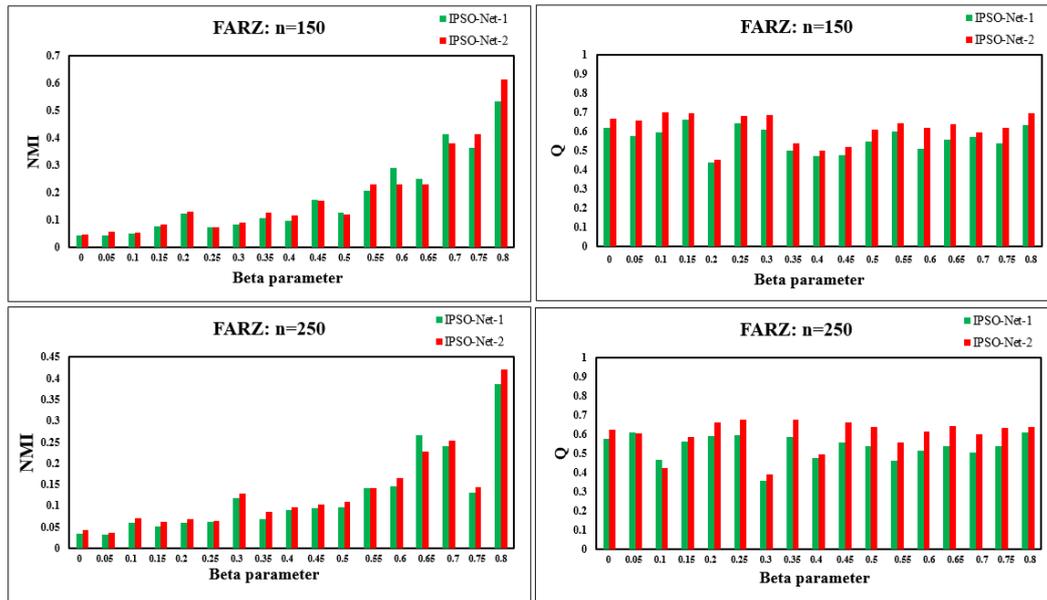


Figure 6: Results of the IPSO-Net at two different variants (IPSO-Net-1, IPSO-Net-2) on $FARZ_{Net}_{\beta \in [0,0.8]}$ benchmark with $n = (150, 250)$ in terms of NMI_{avg} (left side) and Q_{avg} (right side)

On the other hand, the obtained results in terms of modularity (Q) (Figures 5 and 6 (right side)) have proved that the algorithm in the IPSO-Net-2 version clearly outperformed the IPSO-Net-1 version by scoring high values in terms of Q_{avg} . Obtaining high results in terms of Q_{avg} means that the detected communities have strong structures with dense intraconnections that deviate from the null case.

Lastly, the results of the proposed algorithm at IPSO-Net-2 version when $\beta = 0.8$ were compared with the PSO-Net algorithm of [25] that had employed the modularity function as a single-objective function. For the sake of fairness, the common parameters were set to the same value for both methods, i.e., the number of iterations was customized to 100 and the population size was set to 100. In addition, all the experiments' results were reported by considering the average of 10 independent runs. Table 2 shows the comparison in terms of NMI_{avg} and Q_{avg} between the results obtained by both methods, IPSO-Net-2 and PSO-Net.

Table 2: Results obtained by the proposed algorithm in the IPSO-Net-2 version and the PSO-Net algorithm

Networks /Algorithms		PSO-Net	IPSO-Net-2
<i>FARZ_Net_50</i>	Q_{avg}	0.7168	0.7106
	NMI_{avg}	0.4904	0.7951
<i>FARZ_Net_100</i>	Q_{avg}	0.6604	0.6515
	NMI_{avg}	0.5017	0.6861
<i>FARZ_Net_150</i>	Q_{avg}	0.6508	0.6934
	NMI_{avg}	0.4513	0.6141
<i>FARZ_Net_250</i>	Q_{avg}	0.6481	0.6402

	NMI_{avg}	0.2257	0.4211
--	-------------	--------	---------------

The results in Table 2 have indicated that the proposed algorithm in the IPSO-Net-2 version significantly outperformed the counterpart algorithm of [25] (i.e., PSO-Net) in terms of NMI_{avg} and recorded satisfactory results in terms of Q_{avg} . According to the related literature, both measures, NMI and Q are considered the right metrics for evaluating a given solution. In fact, there is no strict positive correlation between these two metrics [37, 38]. From the presented results, it can be concluded that the solution improvement operator, which is developed based on identifying and resetting the complex network nodes in cases where most of their topological links do not belong to their home communities, has enhanced the prediction power of the PSO algorithm.

5. Conclusion

In this paper, a particle swarm optimization algorithm with a solution improvement operator is proposed to capture the hidden community structure in modular complex networks using the modularity density metric as a fitness function. The solution improvement operator enhanced PSO performance by guiding the particles toward a better solution space. The key idea behind the improvement operator is to determine and reassign those nodes that are located in the wrong communities if the majority of their connections do not belong to their current communities, making it appear that these nodes belong to another community. The experimental results showed the proposed method's effectiveness over modular networks of different complexities and sizes. In the future, however, we will focus on enhancing the PSO's performance using an improvement operator based on strong or weak community concepts and adopt other known community detection models as a fitness function. Moreover, we plan to generate different networks and discuss the effect of other control parameters on the performance of the proposed algorithm.

References

- [1] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75-174, 2010, doi: 10.1016/j.physrep.2009.11.002.
- [2] M. Gong, Q. Cai, X. Chen and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 18, no. 1, pp. 82-97, 2014, doi: 10.1109/TEVC.2013.2260862.
- [3] S. Rahimi, A. Abdollahpouri and P. Moradi, "A multi-objective particle swarm optimization algorithm for community detection in complex networks," *Swarm and Evolutionary Computation*, vol. 39, pp. 297-309, 2018, doi: 10.1016/j.swevo.2017.10.009.
- [4] D. A. Abduljabbar, S. Z. M. Hashim and R. Sallehuddin, "An improved multi-objective evolutionary algorithm for detecting communities in complex networks with graphlet measure," *Computer Networks*, vol. 169, pp. (107070-1)-(107070-24), 2020, doi: 10.1016/j.comnet.2019.107070.
- [5] M. N. Al-Andoli, S. C. Tan, W. P. Cheah and S. Y. Tan, "A Review on Community Detection in Large Complex Networks from Conventional to Deep Learning Methods: A Call for the Use of Parallel Meta-Heuristic Algorithms," *IEEE Access*, vol. 9, pp. 96501-96527, 2021, doi: 10.1109/ACCESS.2021.3095335.
- [6] D. A. Abduljabbar, S. Z. M. Hashim and R. Sallehuddin, "Nature-inspired optimization algorithms for community detection in complex networks: A review and future trends," *Telecommunication Systems*, vol. 74, no. 2, pp. 225-252, 2020, doi: 10.1007/s11235-019-00636-x.
- [7] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 464-483, 2018, doi: 10.1109/TEVC.2017.2737600.
- [8] M. Moradi and S. Parsa, "An evolutionary method for community detection using a novel local search strategy," *Physica A: Statistical Mechanics and its Applications*, vol. 523, pp. 457-475, 2019, doi: 10.1016/j.physa.2019.01.133.

- [9] A. Said, R. A. Abbasi, O. Maqbool, A. Daud and N. R. Aljohani, "CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks," *Applied Soft Computing*, vol. 63, pp. 59–70, 2018, doi: 10.1016/j.asoc.2017.11.014.
- [10] K. R. Žalik and B. Žalik, "Memetic algorithm using node entropy and partition entropy for community detection in networks," *Information Sciences*, vol. 445, pp. 38–49, 2018, doi: 10.1016/j.ins.2018.02.063.
- [11] M. N. Haque, L. Mathieson and P. Moscato, "A memetic algorithm for community detection by maximising the connected cohesion," *IEEE symposium series on computational intelligence (SSCI)* pp. 1–8, 2017, doi: 10.1109/SSCI.2017.8285404.
- [12] M. Pourkazemi and M. R. Keyvanpour, "Community detection in social network by using a multi-objective evolutionary algorithm," *Intelligent Data Analysis*, vol. 21, no. 2, pp. 385–409, 2017, doi: 10.3233/IDA-150429.
- [13] J. Cao, Z. Bu, G. Gao and H. Tao, "Weighted modularity optimization for crisp and fuzzy community detection in large-scale networks," *Physica A: Statistical Mechanics and its Applications*, vol. 462, pp. 386–395, 2016, doi: 10.1016/j.physa.2016.06.113.
- [14] H. Liu, F. Yang and D. Liu, "Genetic algorithm optimizing modularity for community detection in complex networks," In *2016 35th Chinese control conference (CCC)*, pp. 1252–1256. IEEE, July 2016, doi: 10.1109/ChiCC.2016.7553259.
- [15] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, pp. 026113, 2004, doi: 10.1103/PhysRevE.69.026113.
- [16] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the national academy of sciences*, vol. 104, no. 1, pp. 36–41, 2007, doi: 10.1073/pnas.0605965104.
- [17] Z. Li, S. Zhang, R. S. Wang, X. S. Zhang and L. Chen, "Quantitative function for community detection," *Physical review E*, vol. 77, no. 3, pp. (036109-1)(036109-9), 2008, doi: 10.1103/PhysRevE.77.036109.
- [18] Y. Zhang, Y. Liu, J. Li, J. Zhu, C. Yang, W. Yang and C. Wen, "WOCDA: A whale optimization based community detection algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 539, pp. (122937-1) (122937-15), 2020, doi: 10.1016/j.physa.2019.122937.
- [19] D. A. Abduljabbar, S. Z. M. Hashim and R. Sallehuddin, "An enhanced evolutionary algorithm for detecting complexes in protein interaction networks with heuristic biological operator," In *International Conference on Soft Computing and Data Mining*, pp. 334–345, Springer, Cham, 2020, doi: 10.1007/978-3-030-36056-6_32.
- [20] D. A. Abduljabbar, S. Z. M. Hashim, and R. Sallehuddin, "An enhanced evolutionary algorithm with local heuristic approach for detecting community in complex networks," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 20, pp 2452–2466, 2019.
- [21] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, pp. 39–43, Ieee, October 1995, doi: 10.1109/MHS.1995.494215.
- [22] K. Deb, "Multi-objective optimization using evolutionary algorithms," New York, NY, USA:Wiley, vol. 16, 2001.
- [23] A. J. Mohammed, K. I. Ghathwan, and Y. Yusof, "Optimal robot path planning using enhanced particle swarm optimization algorithm. *Iraqi Journal of Science*, vol. 61, no. 1, pp: 178–184, 2020, doi: 10.24996/ij.s.2020.61.1.20.
- [24] H. A. Chachan and F. H. Ali, "Using Non-dominated Sorting Particle Swarm Optimization Algorithm II for Bi-objective Flow Shop Scheduling Problems", *Iraqi Journal of Science*, , Vol. 62, No. 1, pp: 275–288, 2021, doi: 10.24996/ij.s.2021.62.1.26.
- [25] A. Abdollahpouri, S. Rahimi, S. M. Majd, and C. Salavati, "A modified particle swarm optimization algorithm for community detection in complex networks," In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 11–27, Springer, Cham, 2018, doi: 10.1007/978-3-319-99740-7_2.
- [26] X. Cai, Y. Shi, Y. Zhu, Y. Qiao and F. Hu, "An algorithm Q-PSO for community detection in complex networks," In *2017 16th international symposium on distributed computing and applications to business, engineering and science (DCABES)*, pp. 76–79, IEEE , October 2017, doi: 10.1109/DCABES.2017.23.

- [27] Z. Chen, F. Liu, C. Gao, X. Li and Z. Zhang, "An enhanced particle swarm optimization based on physarum model for community detection," In *International Conference on Swarm Intelligence*, pp. 99-108, Springer, Cham, July 2017, doi: 10.1007/978-3-319-61833-3_11.
- [28] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan and L. Jiao, "Greedy discrete particle swarm optimization for large-scale social network clustering," *Information Sciences*, vol. 316, pp. 503-516, 2015, doi: 10.1016/j.ins.2014.09.041.
- [29] Z. Shi, Y. Liu and J. Liang, "PSO-based community detection in complex networks," In *2009 Second international symposium on knowledge acquisition and modeling*, vol. 3, pp. 114-119, IEEE, 2009, doi: 10.1109/KAM.2009.195.
- [30] D. Xiaodong, W. Cunrui, L. Xiangdong and L. Yanping, "Web community detection model using particle swarm optimization," In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 1074-1079, IEEE, 2008, doi: 10.1109/CEC.2008.4630930.
- [31] M. Tasgin, A. Herdagdelen and H. Bingol, "Community detection in complex networks using genetic algorithms," *arXiv preprint arXiv:0711.0491*, 2007, doi: 10.48550/arXiv.0711.0491.
- [32] S. Kaur, S. Singh, S. Kaushal and A. K. Sangaiah, "Comparative analysis of quality metrics for community detection in social networks using genetic algorithm," *Neural Network World*, vol. 26, no. 6, pp. 625-641, 2016, doi: 10.14311/NNW.2016.26.036.
- [33] I. S. Dhillon, Y. Guan and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551-556, August 2004, doi: 10.1145/1014052.1014118.
- [34] J. Fagnan, A. Abnar, R. Rabbany and O. R. Zaiane, "Modular networks for validating community detection algorithms," *arXiv preprint arXiv:1801.01229*, 2018, doi: 10.48550/arXiv.1801.01229.
- [35] L. Danon, A. Diaz-Guilera, J. Duch and A. Arenas, "Comparing community structure identification," *Journal of statistical mechanics: Theory and experiment*, vol. 2005, no. 09, pp. P09008, 2005, doi: 10.1088/1742-5468/2005/09/P09008.
- [36] D. J. MacKay, "Information theory, inference, and learning algorithms," vol. 7. Cambridge university press, Cambridge, 2003.
- [37] W. Ying, H. Jalil, B. Wu, Y. Wu, Z. Ying, Y. Luo and Z. Wang, "Parallel conical area community detection using evolutionary multi-objective optimization," *Processes*, vol. 7, no. 2, pp. 111, 2019, doi: 10.3390/pr7020111.
- [38] C.H. Mu, J. Xie, Y. Liu, F. Chen, Y. Liu and L.C. Jiao, "Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks," *Applied Soft Computing*, vol. 34, pp. 485-501, 2015, doi: 10.1016/j.asoc.2015.05.034.