ISSN: 0067-2904

A Review of Community Detection Based on Modularity Optimization in Complex Networks

Samaa F. Ibraheem[1,2] *, Basad Al-Sarray [3]
[1]*Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq*
[2]*Department of Applied Science, University of Technology, Baghdad, Iraq*
[3]*Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq*

Received: 24/10/2022 Accepted: 5/5/2023 Published: 30/5/2024

Abstract

Community detection is one of the most interesting issues nowadays, especially for complex networks. The main problem is to divide these networks into partitions called communities, which are characterized by dense connections inside each part but sparse connections between them. Most networks in the real world display a community structure that must be detected and recovered. There are many approaches (techniques) for detecting communities, which may be arranged in classes according to various bases, like the operational method or the adopted definition of community. However, just a few desired algorithms are applied and adopted for identifying communities, like optimization of quality functions. This review highlights the existing modularity-based community detection methods. These methods are computational approaches based on optimization since they maximize the objective function modularity for each possible partitioning. Also, more attention was paid to demonstrating the quality functions that measure the goodness of these partitions, including the modularity function and its various expressions for different types of networks, which currently appear to be the most promising. In this review, computations are made for partitioning and detecting communities of different networks using the convexified modularity maximization algorithm (CMM), and then these partitions are measured using various quality functions. In addition, a derivation of an augmenting Lagrange multiplier is introduced to optimize the solution, which is implemented by the alternating direction multipliers method (ADMM) algorithm. So, such performance will help researchers find the best methods and choose a suitable quality function relevant to future work.

Keywords: Community detection, modularity maximization, ADMM algorithm.

مراجعة في كشف المجتمعات بناءً على تحسين النمطية في الشبكات المعقدة

سماء فؤاد ابراهيم[1،2*]، بسعاد علي حسين[3]
[1]قسم علوم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق
[2]قسم العلوم التطبيقية، الجامعة التكنولوجية، بغداد، العراق
[3]قسم علوم الحاسوب، كلية العلوم، جامعة بغداد، بغداد، العراق

*Email: Samaa.F.Ibraheem@uotechnology.edu.iq

الخلاصة

يعد كشف المجتمعات أحد أكثرالقضايا إثارة للاهتمام في الوقت الحالي، خاصة بالنسبة للشبكات المعقدة. تكمن المشكلة الرئيسية في تقسيم هذه الشبكات إلى اجزاء تسمى مجتمعات، والتي تتميز بوجود اتصال كثيف داخل كل جزء، ولكن اتصالات نادرة بينها. وغالبا ما تظهر معظم الشبكات في العالم الحقيقي هيكلاً مجتمعياً يجب اكتشافه واستعادته. هناك العديد من الأساليب (التقنيات) للكشف عن المجتمعات التي يمكن ترتيبها في فئات وفقًا لأسس مختلفة، مثلا الطريقة التشغيلية، أو التعريف المعتمد للمجتمع. ومع ذلك، عادة ما يتم تطبيق عدد قليل جدا من الخوارزميات المطلوبة والمعتمدة لتحديد المجتمعات مثل امثلية دوال الجودة. في هذه المراجعة يتم تسليط الضوء على أساليب كشف المجتمعات القائمة على الوحدات النمطية. هذه الاساليب هي نهج حسابية تستند الى الامثلية حيث يتم تحقيق اقصى قدر من دالة الهدف modularity لكل تقسيم ممكن. ايضا تم اعطاء تفصيل للدوال التي تقيس مدى جودة هذه التقسيمات بما في ذلك دالة modularity وصيغها المختلفة لأنواع مختلفة من الشبكات، التي تبدو حاليًا أكثر وعداً. في هذه المراجعة يتم اجراء حسابات لتقسيم وكشف المجتمعات في شبكات مختلفة باستعمال خوارزمية (CMM) ثم يتم قياس هذه التقسيمات باستعمال دوال الجودة المطروحة. بالإضافة الى ذلك، يتم تقديم اشتقاق لمضاعف لاكرانج لجعل الحل امثل والذي يتم تنفيذه باستعمال الطريقة المتعددة للمضاعفات بالاتجاه المتناوب (ADMM). بهذا الطرح سوف نساعد الباحثين على ايجاد افضل الطرق واختيار دالة جودة مناسبة للعمل المستقبلي.

1. Introduction

The employment of social networks has greatly increased in the last few years, where people are more likely to form groups based on their similarities in specific things like common interests, backgrounds, hobbies, etc. The partition of a network is needed to identify these groups. This partition is called clustering or community detection, and these groups are called communities or clusters, with the property that each community, which is represented as a subgraph, has elements that are more likely to be similar than other elements in the rest of the communities. In other words, the main problem is to divide a community into partitions (communities), which are characterized by dense connections inside each part but a sparse connection between them. Many community detection methods in social networks and other types of networks are presented; some of them are traditional methods like graph partitioning [1] [2], hierarchical clustering [3] [4], fuzzy clustering [5], partitional clustering such as k-means clustering with its extensions [6] [7], and spectral clustering [8] [9]. whereas other methods involve the optimization of some quality function, like the modularity function [10] [11], and another technique based on statistical inference that aims to deduce a data set's properties, like techniques based on Bayesian inference [12], block modeling [13], model selection [14], and information theory. So this review focuses on demonstrating and illustrating the quality functions, especially the modularity function and its usage as a function that wants to be optimized. The remaining part of this study is organized as follows: Section 2 gives a brief review of existing community detection methods and their classification. The most common quality functions and their implementation on given data sets are introduced in Section 3, along with a comparative study of different forms of the modularity function for different types of graphs. Section 4 poses the problem of optimizing the modularity function with recently used approaches. Section 5 takes the convex optimization of the problem and its implementation given in Section 6. Finally, Section 7 summarizes and concludes the article.

2. Community Detection

The main task in network community detection is to identify the communities of these networks. This identification is important for many reasons. It helps us understand network functionality. In the social network, for instance, community detection helps to discover

people with a common interest and keeps them tightly connected [15]. Despite the substantial interest of the scientific community over the last few years, no commonly accepted solution for detecting communities exists yet. Thus far, several techniques for the optimization of community detection have been introduced in the scientific literature. They may be arranged in various classes based on many bases, like the operational method [16], the specific definition of community by covering the meta-definition of community [17], etc. This section provides an overview of various techniques for the identification of communities that are categorized according to their mathematical topics. For more knowledge, read [18] and [19]. The community detection methods in graphs that represent static networks (Figure 1) can be classified into four broad categories as follows [20]:

1- Spectral methods: an approach that uses the spectral properties of the graph to detect clusters [21]. These methods are not reliable in very sparse networks because of the shapeless eigenvalue separation and the expensive computations.
2- Statistical inference methods: They adopt a normal approach to fitting data to a generative network model based on statistical inference. Stochastic block models (SBMs) are considered the most familiar generative model for working with communities and networks. [22][23].
3- Methods based on optimization: The idea behind these methods is to try to maximize or minimize some quality functions that give the community structure.
4- Dynamic clustering: these methods use the running dynamics of the networks, like a random walk, diffusion, and spin dynamics, to detect community structure in networks. They seek to hit spin configurations that return the optimum Hamiltonian distance. Currently, there is a lot of effort being put into enhancing the methods for analyzing temporal networks because of the growing availability of time-stamped data from networks [24]. In particular, much attention is given to the problem of detecting dynamic communities, and clustering algorithms that are involved in static networks may be implemented for dynamic networks [25] [26].
5-

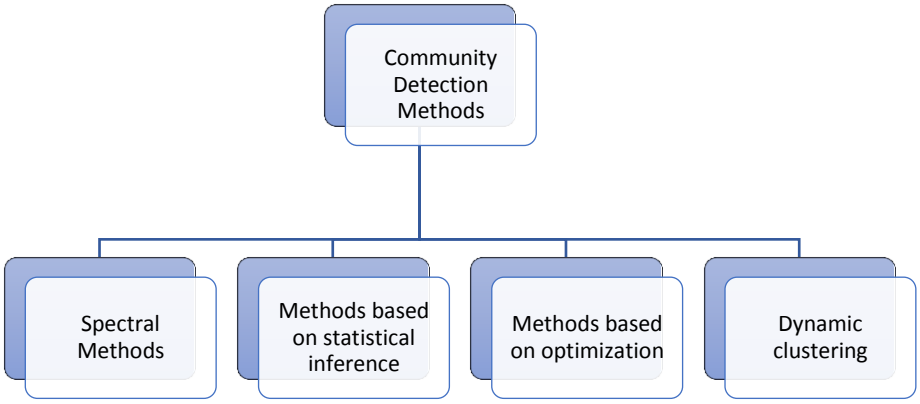

Figure 1: Categories of community detection methods in static network

3. Quality Functions

Quality functions are functions that assign a number to each partition, so they are needed to indicate the quality of clustering (overall possible clustering) to gain good clusters (partitions). In other words, quality functions are measuring functions that can express the goodness of a partition, keeping in mind that the best partitions depend on the used definition of community and the utilized quality function. There are many quality functions, but the most commonly used are mentioned below:

Given a graph *G* with *V* vertices, the partition of *G* divides *V* into *K* sets V_k, *k=1, …, K*, such that $V = V_1 \cup V_2 \ldots \cup V_k$ and $|V_k|$ gives the size of the set V_k. Let $A = [a_{ij}]$ be the adjacency matrix of *G*, the definitions of the most commonly used quality functions are listed below:

- Modularity: Given its details and development in subsection B.
- Local Density: Given by Eq. 1:

$$Q_{LD} = \frac{1}{2}\sum_{k=1}^{K}\left[\frac{\sum_{i\in V_k}^{k}\sum_{j\in V_k}^{k} a_{ij}}{|V_k|^2}\right] + 1 - \sum_{k=1}^{K}\left[\frac{\sum_{i\in V_k}^{k}\sum_{j\notin V_k}^{k} a_{ij}}{|V_k|\,|V-V_k|}\right] \quad (1)$$

- Global Density: Given by Eq. 2:

$$Q_{GD} = \frac{1}{2}\left[\frac{\sum_{k=1}^{K}\sum_{i\in V_k}\sum_{j\in V_k} a_{ij}}{\sum_{k=1}^{K}|V_k|^2} + 1 - \frac{\sum_{k=1}^{K}\sum_{i\in V_k}\sum_{j\in V-V_k} a_{ij}}{\sum_{k=1}^{K}|V_k|\,|V-V_k|}\right] \quad (2)$$

- Local Weighted Density: Given by Eq. 3:

$$Q_{LD_w} = \sum_{k=1}^{K}\frac{|V_k|}{2|V|}\left[\frac{\sum_{i\in V_k}\sum_{j\in V_k} a_{ij}}{|V_k|^2} + 1 - \frac{\sum_{i\in V_k}\sum_{j\in V-V_k} a_{ij}}{|V_k|\,|V-V_k|}\right] \quad (3)$$

- Distance Based Quality Function: Given by Eq. 4:

$$Q_{DB} = \frac{1}{|V|^2}\,\|A - A_V\| \quad (4)$$

where $\|.\|$ is a matrix norm and A_V Eqal 1 or 0 refers to whether *i* and *j* belong to the same community or not.

- Node Membership Quality Function: Given by Eq. 5:

$$Q_{NM} = \frac{1}{2|V|}\sum_{i\in V}\left[M(i,V_{(i)}) + 1 - M(i,V-V_{(i)})\right] \quad (5)$$

where the node membership is defined as subscript $M(i,V) = \frac{1}{|V|}\sum_{j\in V} a_{ij}$ and $V_{(i)}$ indicates the community that the node *i* belongs to. Table 1 gives the implementation of these functions on different data sets.

Unfortunately, quality functions are practical to discuss partitioning for networks with the same number of clusters, whereas the comparison of partitions of networks with different numbers of clusters is not explicit and can lead to vagueness, as discussed separately by [27], and the partitions in clusters of roughly equal size are most likely modularity values, as shown by [28]. Moreover, the problem of determining which clusters are best for a specific partitioning is still open. It is worth noting that many efforts were made to deal with such a problem, for instance, by building a quality function starting with the quality expression of a single community [29] and gaining the best partition in a confirmed number of clusters. Although, for the general case, using optimization of quality functions for identifying communities is still adopted [30–32].

The following subsection will give more detailed computations and various mathematical expressions to the first quality function given by Newman and Girvan since it gives the partitioning quality of networks into communities, so it solves the community detection problem [10]. For more details about the other quality functions and their calculations, see [33] and [34].

3.1 Network Representation

Networks describe interactions between entities. Basically, any mathematical object involving points and connections between them may be called a graph. Based on this concept, a network that has n entities with m interactions can be mapped to the graph *G*(*V*, *E*) with *n* vertices and *m* edges. An edge (*u*, *v*) joins the vertices *u* and *v*, and it is usually abbreviated to *uv*. The set of vertices (*V*) and the set of edges (*E*) in complex networks have formed a graph with non-trivial topological features [35]. A subgraph of a graph *G* is already a graph where its vertices and edges are subsets of *V* and *E* respectively. The maximum number of edges in

the simple graph equals $n(n-1)/2$, where in this case G is called a complete graph. A graph is usually represented using matrices such that the graph G with n vertices (G of order n) has an $n \times n$ adjacency matrix A, whose entries are in formula 6:

$$a_{ij} = \begin{cases} 1 & i,j \in E \\ 0 & otherwise \end{cases} \quad (6)$$

In directed graphs, one distinguishes between two edges, (u, v) and (v, u), i.e., $uv \neq vu$, whereas in weighted graphs, a positive number is given to each edge uv, which is denoted by W_{uv}. The degree of $v_i \in V$ is the number of edges indicated on v_i; it is denoted by d_{vi} or simply d_i, equivalently, it represents the cardinality of vertices that are adjacent to v_i, i.e. $d_i = \sum_{j=1}^{n} a_{ij}$. For a graph G with m edges, there are $2m$ half-edges, in total, it equals the sum of all vertices' degrees, so $2m = \sum_{i=1}^{n} d_i$. The sequence of degrees is the list of the vertices' degrees, $\{d_1, d_2, \cdots, d_n\}$. In directed graphs, for each vertex, there are two types of degrees: in-degree and out-degree, denoted by $d_i(in)$ and $d_i(out)$, respectively. The strength is the analogue of degree on weighted graphs. In many real networks, the graphs that represent them are weighted, in both cases of oriented edges [36] and [37].

3.2 Modularity as a Quality Function

Depending on the representation of the network, the modularity function, which is denoted by Q, can be defined as follows: It is a function that measures the strength of the division of a network into communities (subgraphs) by taking into account the degree of distribution, such that networks with a high score of modularity have dense connections among the nodes within a cluster but sparse connections between nodes in different clusters.

To construct modularity, Neman et al. proposed to find the fraction of edges that exist between vertices within the same subgraph, and then subtract from it the fraction of such expected edges that would exist if edges were randomly positioned regardless of vertex subgraph. In mathematical terms, the total number of edges that run between vertices of the same cluster defined by Kronecker delta ($\delta(C_i, C_j)$ is a function that yields 0 or 1 if vertices i and j are in the same community or not) is given in eq. (7):

$$\sum_{edges(i,j)} \delta(C_i, C_j) = \frac{1}{2} \sum_{ij} a_{ij} \delta(C_i, C_j) \quad (7)$$

where C_i denotes the community of vertex i. The expected number of edges between vertices if edges are placed at random, in null term model P_{ij} , is as formula (8):

$$\frac{1}{2} \sum_{ij} P_{ij} \delta(C_i, C_j) \quad (8)$$

Taking the difference between the two numbers gives an expression for the difference between the actual and expected number of connections in the network. The fraction of the resulting expression, after being divided by a number of edges m, gives the modularity Q. Many expressions describe modularity Q according to the type of network (graph), but they all have the same idea that a large value of Q indicates a good community structure. Below are the most common forms.

• The general expression: The sum covers all pairs of vertices. Here, the null model term (P_{ij}) is a random graph that was introduced in [38], then independently by Erdos and Renyi [39], that has a degree distribution of Poisson, which is a mismatch of the distributions that had been found in real networks, expression 1 (Table 1).

• The classic expression: Here, the null model was taken with a similar degree distribution to the original graph to be more compatible with the function and structure of real networks [40], so the term P_{ij} is computed as follows: Set the degree of v_i so that the number of half edges starts at v_i. So to make an edge in the graph, two half edges need to be connected. If that connection can be made by picking half edges uniformly, the probability of randomly picked edges being between the two vertices will be equal to eq. (9):

$$\frac{d_i}{2m}.\frac{d_j}{2m}+\frac{d_j}{2m}.\frac{d_i}{2m}=\frac{d_i d_j}{2m^2} \tag{9}$$

Because the probability of picking a half edge of node v_i is $\frac{d_i}{2m}$, where the denominator is all half edges in the graph. Now, since there are *m* edges, each of which has an expected value of $\frac{d_i d_j}{2m^2}$, then the expected number of edges between *vi* and v_j is given in (10):

$$P_{ij}=m\frac{d_i d_j}{2m^2}=\frac{d_i d_j}{2m} \tag{10}$$

Hence, the difference between the actual number of edges between vertices v_i and v_j and the expected number of edges between them is $a_{ij}-\frac{d_i d_j}{2m}$ so expression 1 becomes expression 2 (Table 1).

• Equivalent form: Considering that the graph (community) was partitioned into k subgraphs (communities), the last form can be written in a different equivalent way. In fact, the inputs to the sum are taken from vertex pairs (v_i, v_j) that belong to the same subgraph, so these inputs can be grouped with each other and the sum over the pairs of vertices rewritten as a sum over the subgraphs. So modularity *Q* becomes expression 3 (Table 1). Here, the first term of each summation represents the edges' fraction of the graph within community *C*, while the other term gives the fraction of edges that would be expected to be found within community *C* if edges were inserted at random in the network, keeping in mind that the degree sequence matches the original graph.

• Matrix form: define the modularity matrix *B* as eq. (11):

$$B_{ij}=a_{ij}-\frac{d_i d_j}{2m} \tag{11}$$

then the classical form can be expressed as expression 4 (Table 1). since if the graph is partitioned into two clusters C_1 and C_2, then the vector *s* (column vector) will be represented as $s_i = +1$ or $s_i = -1$ if vertex i belongs to C_1 or C_2 respectively, so Kronecker delta will equal to $(1/2)(s_i s_j + 1)$ [41].

• For undirected weighted networks, two modularity expressions were introduced, expressions 5 and 6 (Table 1). The first is expressed in terms of vertex degree over all vertices in the graph besides Kronecker delta, while the second takes the computations over each subgraph, i.e., in terms of the total degree of C. In both forms, W is the total weight of all of the edges in the graph. W_{ij} is the actual weight of the edges existing between vertices *i* and *j* in the original graph, s_i and s_j are the strengths of vertex *i* and *j* respectively (the weight sum of the edges that incident on *i* and *j*); hence, $\frac{s_i s_j}{2m}$ is the expected weight of the edge (*i, j*) in the null model. In the second form, *sc* is the total strength of the community's vertices, and W_c is the total weight of the internal edges of *c*. These computations run over all *k* communities.

• For directed graphs, modularity *Q* was defined with entries d_i (out) and d_i (in) in the second expression's term of expression 2, where $d_i(in)$ and $d_i(out)$ indicate the number of edges that begin and end at *i* respectively, so modularity *Q* becomes expression 7 (Table 1).

• For weighted directed graphs, modularity *Q* is defined with an expression similar to the previous one but with weighted edges, i.e., a_{ij} is replaced by the actual weight W_{ij} and *m* is replaced by the total weight *W*; see expression 8 (Table 1).

• In the overlapping communities case (subgroups can be joined), for unweighted and undirected groups, modularity Q is extended using the numbers O_i and O_j, which are the numbers of subgroups containing nodes i and j, respectively [16], which gives expression 9 (Table 1).

Table 1: Various modularity expressions for different types of graphs

No.	Expression	Directed	Weighted	Overlapping	Ref.
1	$Q = \frac{1}{2m}\sum_{ij}(a_{ij} - P_{ij})\delta(C_i, C_j)$	N	N	N	[10]
2	$Q = \frac{1}{2m}\sum_{ij}\left(a_{ij} - \frac{d_i d_j}{2m}\right)\delta(C_i, C_j)$	N	N	N	[42]
3	$Q = \sum_{c=1}^{k}\left[\frac{m_c}{m} - \left(\frac{d_c}{2m}\right)^2\right]$	N	N	N	[10]
4	$Q = \frac{1}{4m}s^T Bs$	N	N	N	[41]
5	$Q = \frac{1}{2w}\sum_{ij}\left(W_{ij} - \frac{s_i s_j}{2w}\right)\delta(C_i, C_j)$	N	Y	N	[42]
6	$Q = \sum_{c=1}^{k}\left[\frac{W_c}{W} - \left(\frac{s_c}{2w}\right)^2\right]$	N	Y	N	[16]
7	$Q = \frac{1}{2m}\sum_{ij}\left(a_{ij} - \frac{d_i(out)d_j(in)}{2m}\right)\delta(C_i, C_j)$	Y	N	N	[43]
8	$Q = \frac{1}{2w}\sum_{ij}\left(W_{ij} - \frac{W_i(out)W_j(in)}{2W}\right)\delta(C_i, C_j)$	Y	Y	N	[44]
9	$Q = \frac{1}{2m}\sum_{ij}\frac{1}{O_i O_j}\left(a_{ij} - \frac{d_i d_j}{2m}\right)\delta(C_i, C_j)$	N	N	Y	[45]

3.3 Modularity Range

Modularity can be either positive or negative, where positive values refer to the possible existence of community structure [41]. So, one can seek community structure totally by studying the partitions of networks that take positive, and best to be maximum, values of modularity. Modularity is basically dependent on the distribution of the edges and their number rather than on the number of vertices; also, isolated nodes have no impact on modularity [46], and this can be obviously seen in the third expression by Newman. For more details, this will be illustrated in three cases below, including the upper and lower bounds of modularity. Note that m_c is the number of edges within subgroup C, d_c is the degree of subgroup C, $d_c(int)$ and $d_c(ext)$ are the internal and external degrees (respectively) of vertices within C, then each subgroup will contribute to Q by formula (12):

$$\frac{m_c}{m} - \left(\frac{d_c}{2m}\right)^2 \tag{12}$$

Now, since $d_c = d_c(int) + d_c(ext) = 2m_c + d_c(ext)$ the above contribution can be rewritten as (13):

$$\frac{m_c}{m} - \left(\frac{2m_c + d_c(ext)}{2m}\right)^2 = \frac{m_c}{m} - \left(\frac{m_c}{m} + \frac{d_c(ext)}{2m}\right)^2 \tag{13}$$

Case 1: The trivial partition where all vertices belong to only one subgraph, i.e., k = 1, which is the graph itself. In this partitioning, there is a guarantee that there are zero external edges. So this partition has a value of zero for modularity, since $m_c = m$ and $d_c = 2m$ which yields $Q = 0$ [45].

Case 2: Suppose one has two subgraphs. The contribution of a subgraph is minimized when mc is zero and $d_c(ext)$ is as large as possible i.e., $d_c(ext) = m$. In this case, each subgraph

will distribute by −1/4 which yields $Q = -1/2$ where modularity reaches its lower bound, as Brandes et al. mentioned in their theorem [46].

Case 3: Suppose a partition with k subgraphs. The first term in (12) corresponds to the internal density of edges in the subgraph, while the second corresponds to the expected density of edges in the random graph null model, as mentioned earlier. The difference in the sum would indicate the amount of non-randomness in subgraph *C*. The higher the difference, the better the partition, and high values of *Q* give confidence in the non-random location of edges within *C*. In this case, by definition, C is considered a community if it satisfies the inequality (14).

$$\left[\frac{m_c}{m} - \left(\frac{d_c}{2m}\right)^2\right] > 0 \tag{14}$$

Based on this principle, if all subgraphs (*C*) of the partition are communities, i.e., satisfy (14), the value of *Q* due to this partition is positive. Besides, each summand can't exceed the first term $\frac{m_c}{m}$, which makes modularity a bounded function. So the inequality (15) is deduced:

$$\sum_{c=1}^{k}\left[\frac{m_c}{m} - \left(\frac{d_c}{2m}\right)^2\right] \leq \sum_{c=1}^{k}\frac{m_c}{m} = \frac{1}{m}\sum_{c=1}^{k} m_c \tag{15}$$

Now that the maximum value of m_c does not exceed m ($mc = m$ when all edges of the graph are internal edges for the subgraphs with zero external edges), the last term will not exceed 1 [29]. Mathematically, it is represented as an inequality (16):

$$\frac{1}{m}\sum_{c=1}^{k} m_c \leq 1 \tag{16}$$

Table 2 shows data sets of some common real networks with their modularity values that agree with the given range [47]. Note that, for the trivial case where all vertices are in a single subgroup (the group without partitioning), for instance, 100% of edges go between the same subgroup vertices, and that result is expected. There is no other place for edges to be located. The difference between the two terms is then equal to zero, which informs us that the only existing partition is trivial partitioning. In addition to that, when the fraction of edges among vertices of the same community is notably greater than what was expected by chance, the measure of *Q* will give a score greater than zero. Briefly, modularity is strictly less than 1 (in fact, it belongs to [-1/2, 1]) and takes a negative number if a network has fewer edges connecting vertices of the same type than what would be expected by chance and a positive if there are more. Normally, in the real world, modularity does not fulfill the value $Q = 1$, even for an ideally mixed network, if every vertex of the same type is connected only. So it is considered a dimensionless quantity between 0 and 1 for networks [48] and [49].

Table 2: Standard Newman-Girvan modularity value for data sets of some of the common networks [47]

Network	Vertices No.	Edges No.	Newman Modularity
Karate	34	78	0.0962
Dolphin	62	158	0.3458
American Football College	115	613	0.5290
Facebook	3958	84241	0.0224
Political Blogs	1107	9537	0.0019
Protein Protein	2284	6644	0.0111
AS-Level Internet	6444	11284	0.0336
Chesapeaka Synthetic	39	170	-0.1188
Delaunay Synthetic	1024	3056	-0.2246
Twitter	2623	21000	0.0092

4. Modularity Maximization

Modularity is oftentimes used in optimization methods for detecting community structure in networks; see Table 3. Modularity maximization is considered NP-hard [46] [50] [51]; in spite of that, indeed, modularity becomes the best-known and most studied topic for clustering networks [52] [53]. It is in light of the idea that there is no expectation that a random graph will have a community structure, so the possible existing communities are gained by balancing between the actual density of edges and the expected density that would appear in the communities if the vertices were connected regardless of community structure. As mentioned earlier, this expected edge density is subject to the selected null model. In other words, the choices of the null model term allow us to symbolize specific different network structure's features, like correlations [54], bi-partiteness [55] [56], space embeddedness [57], signed edges [58], etc.

Table 3: Modularity optimization based method algorithms

Algorithm/Author	Author's strategy	Ref.
Greedy techniques: It is a method for agglomerative hierarchical clustering. Groups of vertices are collected to establish communities of larger size, in which modularity increases with each process of merging.		
Girvan–Newman(GN) algorithm (2002)	Detecting communities by thoughtfully deleting edges that have high edge betweenness, with $O(n)^3$ as computational complexity in sparse networks	[59]
Newman (2004)	Finding communities in large-scale networks by combining communities in pairs, aiming to optimize modularity, is fun. It runs in a time $O(n^2)$ on sparse networks, which is fast.	[60]
Clauset, Newman and Moore (2004)	Exploiting some shortcuts in optimization regarding Newman's idea [58] and earning communities with greedy algorithms in more complicated networks is also performed much more quickly in running time $O(n \log^2 n)$.	[32]
Blondel, Guillaume & Lambiotte (Louvain algorithm) (2008)	Developing the function of modularity to multi-level modularity optimization that is better than modularity-based algorithms in time complexity per square of the network's size. Its computational complexity is O(nlogn), but the optimal solutions are local with respect to community merging.	[61]
Schuetz and Caflisch (2008)	Each iteration allows for the merger of more than one community pair, which reduces the danger of having local optima trapped in modularity and results in more reasonable partitions.	[62]
Ye, Hu and Yu (2008)	The initial partition is the movement of vertices between the communities, which is not allowed without decreasing Q, so modules that were found show stronger association among members than what was found by other methods.	[63]
Xiang, Chen and Zhou(2009)	Optimization was significantly improved by starting the hierarchical agglomeration from an arrangement obtained by combining the original isolated vertices into communities with larger sizes rather than from the individual vertices.	[64]
Noack and Rotta (2009)	Achieving higher-quality modularity by applying refinement strategies based on greedy agglomeration with local search at many steps.	[53]
Waltman and Nees Eck (2013)	The modular optimization process is achieved twice. So each individual community reaches a local optimal value. This algorithm works well for medium- and small-sized networks.	[65]
Simulated annealing: a global search optimization algorithm that makes use of randomness within the search to seek the global optimum of a function		
Guimer`a and Amaral (2005)	This technique uses two types of moves: local moves done by randomly shifting a single vertex from one cluster to another, and global moves consisting of mergers and splits of communities. Although it gives a very close value to the true modularity maximum, it is slow.	[66]
A. F. Yaqoob and B. Al-Sarray (2020)	Using the Tabu Search method with fuzzy c-mean (FCM) in different settings to detect communities in many types of networks by maximizing the objective functions.	[67]

Extremal optimization techniques: It is based on local variable optimization, expressing the contribution of each unit of the system to the global function.		
Boettcher and Percus (2001)	Using probabilistic selection, in which vertices are graded based on their fitness values, is based on the optimization of local variables.	[68]
Duch and Arenas (2005)	Modularity is represented as a sum that runs over all vertices such that the local modularity of any vertex equals that of the corresponding term in this sum. In a time of $O(n^2 \log n)$, the algorithm finds a good approximation of the modularity maximum.	[69]
Spectral optimization: uses the eigenvalues and eigenvectors of the modularity matrix to optimize modularity.		
Ruan and Zhang (2007)	This method is a developed form of the K-cut method for community detection in large networks. The method is implemented for all clusters of the first partition for a small range of k. It can automatically detect different types of tumors without any prior knowledge.	[70]
Wang, Shen and Ouyang (2008)	Community vectors were used to obtain high modularity partitions. The algorithm works very well when the networks are split into groups, but no more than four groups	[71]
Richardson and Mucha (2009)	Obtaining graph tripartitions with large modularity using the modularity matrix and its leading pair of eigenvectors.	[72]

5. Clustering Optimization for Community Detection

Mathematical optimization plays a great role in finding the optimal solution to practical problems in many fields like economics, medicine, engineering, data science, and artificial intelligence, so this article focuses on applying this methodology in data mining, especially clustering and therefore community detection. Recently, modularity maximization techniques have taken a different turn in an attempt to relax the modularity function by adding tuning parameters or using convex programming relaxation and semi-definite relaxation methods. For instance, [30] presents the convex modularity function for DCSBM, whereas [73] presents the semidefinite relaxation for block models. In the future, one may take advantage of these relaxations to obtain better results. For further reading, see [74].

However, two types of clustering methods that are based on optimization methods are introduced. The first is the classic method, the K-means method, followed by the more recently used method, convex clustering for community detection.

5.1 Center Based Clustering

The idea behind center-based clustering is to collect the given data (samples) in groups in such a way that, after choosing a number of centers belonging to the data, each point in the data is assigned to its closest center [75]. This way, the total distance of the samples from their nearest centers is minimal. Several distance measure types can be considered for different types of data, for instance, the Manhattan distance, the Bregman distance, the Mahalanobis distance, and the Euclidean distance as a special case of the elliptic norm.

The most classical center-based clustering method is K-means clustering, which assigns each point x_i in the data to the corresponding cluster a_j such that every point in each cluster has a minimal squared Euclidean distance from its center. According to that, each point x_i located on its j_{th} cluster is represented by a z vector of n elements $z_i=k$ $k=1,\ldots,K$, eq. (17):

$$d(x_i, a_j) = \min_{1\le j\le k} \left\| x_i - a_j \right\|^2 \tag{17}$$

This problem is considered an optimization problem applied to the total distance of sample x_i as eq. (18):

$$\min F(a_1, a_2, \ldots, a_k) = \sum_{i=1}^{n} \min_{1\le j\le k} \left\| x_i - a_j \right\|^2 \tag{18}$$

In general, the minimum of convex functions is not convex, so the *K*-means method, given in algorithm 1 [75], is also a non-convex problem in general.

Algorithm1: K-means method

Input: an n by d matrix of data x, no. of clusters (k)
Output: a k by d matrix A, labelling vector Z of n elements

Step1: Select a matrix of K Rows as centers from X randomly
Step 2: calculate the distance from each center aj to each input x_i. Set assignment matrix S_{ij} i=1, …, n and j=1, ..., k
Step 3: assign each x_i to the cluster j that has min sij in i-th row $z_i=j$
Step 4: Update the Center for each cluster by the mean of its points so:

$$a_j = \frac{\sum x_i}{|z_i|}\ ,\ z_i = j, \qquad j = 1, \dots, \mathrm{k}$$

Step 5: Repeat steps three and four until convergence.

In fact, K-means the method is good as an initial method since it is easy to use with a given K (known). Although it is suffering from that, there is no guarantee of an optimal solution; besides, it is sensitive to the initialization of centers in clusters, where it may stick in the local optimum and not converge to the global optimum. So to alleviate this case, many different initializations of centers may be used. Also, the performance of K-means on spherical data with equal radii is more stable than working on other types.

5.2 Convex Optimization

Convex optimization is at the heart of mathematical optimization since it provides the global maximum or minimum of a convex function defined over convex sets; i.e., in convex optimization, any locally optimal point is a globally optimal point, so one avoids the non-convex case, which has multiple feasible and multiple local minima within the given region. Moreover, the set of optimal points in the convex problem is itself a convex set. Convex relaxation has a great role since it has the robustness and computational power to enhance computationally and statistically the solutions to community detection problems [74–77]. So recent work has turned to the use of semidefinite programming in convex optimization to give a convex relaxation to these problems and therefore an efficient solution.

Convex relaxation can be derived in different ways. Usually, for community detection, these derivations give rise to convex programming of similar forms. This section focuses on the convexified modularity maximization proposed by [30] that applies under the SBMs as follows:

Given the observed network data represented by its adjacency matrix A, the central issue is to derive the modularity function of community structure under the (p-q) planted partition model (SBM) of partition matrix X. In this model, X is an n by n symmetric matrix where X_{ij}=1 when $c(i)=c(j)$ (i.e vertex i and j belong to the same community) and X_{ij}=0 otherwise, hence X_{ii}=1 for i=1, …, n. Moreover, dealing with a special case of this model, suppose $1>p>q>0$ where $p=Pr(A_{ij}=1)$ when I and j belong to the same partition and $q= Pr(Aij=1)$ otherwise. Note that, this formed a K by K probability matrix $B=Pr(A_{ij})$ with p in the diagonal and q in the off diagonal elements of B. The standard form of the modularity function represented by X is defined as eq. (19):

$$Q_X = \sum_{ij} \left(a_{ij} - \frac{d_i d_j}{2m} \right) X_{ij} \tag{19}$$

The main goal is maximizing Q_X by finding the optimal X that achieves that goal, where X is one of all possible partition matrices subscripts for $P_{n,k}$ for a network of n nodes that are partitioned into K communities. The form (19) as known is generalized by adding the tuning parameter λ because of the resolution limit [78], so (19) can be rewritten in generalized matrix form as formula (20):

$$\max_X \langle X, A - \lambda dd^T \rangle \quad subject\ to\ X \epsilon \bigcup_k P_{n,k} \tag{20}$$

The optimization problem (20) was convexified [30] since the partition matrix X is positive semidefinite and its elements lie between 0 and 1, with 1 in its diagonal. Therefore, the convexified modularity maximization problem is given as problem (21).

$$\max_X \langle X, A - \lambda dd^T \rangle \quad subject\ to\ X \succcurlyeq 0,\ \ 0 \le X \le J,\ \ diag(X) = 1 \tag{21}$$

where J is a matrix with all entries equal to one.

The last problem is a semi-definite program, so it can be penalized by adding an additional term in the objective function to convert it to a minimization problem and then solve it; thus, it becomes (22):

$$\min\ \langle X, \alpha I_n - A + \lambda dd^T \rangle \quad \text{subject to } X \succcurlyeq 0\ ,\ 0 \le X \le J,\ \ diag(X) = 1 \tag{22}$$

which is equivalent to problem (23):

$$\min\ \langle X, E \rangle \quad \text{subject to } X \succcurlyeq 0 \quad 0 \le X \le J,\ \ diag(X) = 1 \tag{23}$$

where $E = \alpha I_n - A + \lambda\, dd^T$ and α is a tuning parameter that amounts to the trace penalization of X. Practically, it can be a small constant or zero [76], and it is usually used to recover a low-rank matrix structure in the SDP relaxation literature (see [77-80]). Therefore, the only need is to tune the parameter λ, which has a statistical meaning since it depends on p and q that relate to the specific SBM model.

To solve the problem, one needs to use one of the optimization methods for solving SDP, which is the alternating direction method of multipliers (ADMM), introduced in [81] and used in [30].

Now define the function $f{:}A \rightarrow R^+$ as formula (24)

$$f(a \in A) = \begin{cases} 0 & a \in A \\ +\infty & a \notin A \end{cases} \tag{24}$$

Notice that, f is convex as long as A is a convex set, so (23) can be written as Eq. 25:

$$\min_{X,Z} f(X \succcurlyeq 0) + f(0 \le Z \le J_n) + \langle X, E \rangle \quad \text{subject to } X = Z \tag{25}$$

Note: The set of positive semidefinite matrices is a convex set.

Define the augmented LaGrange of this optimization problem as Eq. 26:

$$L_p(X, Z; \Lambda) = f(X \succcurlyeq 0) + f(0 \le Z \le J_n) + \langle X, E \rangle + \frac{\rho}{2} \|X - Z + \Lambda\|_F^2 \tag{26}$$

So there are two steps: The first is minimizing (26) with respect to X (Λ and Z are fixed), hence the gradient of L_p with respect to X is given by Eq. 27:

$$\nabla_X L_p(X, Z; \Lambda) = \rho(X - Z + \Lambda) + E \tag{27}$$

Setting the gradient eq. (27) to zero gives Eq. 28:

$$X^{l+1} = \frac{-E}{\rho} + Z^l - \Lambda^l \ \text{ with } X \succcurlyeq 0 \tag{28}$$

So (26) will have a solution in the form of Eq. 29:

$$X^{l+1} = \left(Z^l - \Lambda^l - \frac{E}{\rho} \right)_+ \tag{29}$$

where $(X)_+$ indicates positive semidefinite of the Eigen value decomposition of X, i.e., $(X)_+ = V \sum_+ V^T$.

The second step is to minimize (26) with respect to Z (Λ and X are fixed), hence the gradient of L_p with respect to Z is given by Eq. 30:

$$\nabla_Z L_p(X, Z; \Lambda) = \rho(X - Z - \Lambda)(-1) \tag{30}$$

Setting the gradient in (30) to zero gives Eq. 31:

$$Z^{l+1} = X^{l+1} + \Lambda^l \text{ with } 0 \leq Z \leq J_n \tag{31}$$

which has a solution of a closed form, Eq. 32:

$$Z = \min(\max(X + \Lambda, 0), J_n) \tag{32}$$

which keeps Z between zero and one. Also, Λ is updated to $\Lambda^l + (X^{l+1} - Z^{l+1})$ using the iteration method to solve the problem with Z_0 =0 and Λ_0 =0 as initial values, and simply ρ=1 may be chosen for (29). Therefore, ADMM solves (25) and is described in algorithm 2.

Algorithm 2: ADMM for convexified modularity maximization method

Input: initial values Z_0 =0 and Λ_0 =0, #iteration=l
output: X

step1: For k=1 to l
step2: $X_{k+1} = \left(Z_k - \Lambda_k - \frac{E}{\rho}\right)_+$
step3: $Z_{k+1} = \min(\max(X_{k+1} + \Lambda_k, 0), J_n$
step4: $\Lambda_{k+1} = \Lambda_k + (X_{k+1} - Z_{k+1})$
end

In the field of convex optimization for community detection approaches, many studies are given to achieve the optimal solution, especially under planted partition models. Recently, various theoretical properties of convex optimization methods, given in Table 4, have been studied in depth, particularly when dealing with statistical models for community detection.

Table 4: Theoretical properties of some convex optimization methods

Properties	Ref.
strong consistency with a growing K (number of communities)	[79], [82]
strong consistency: sharp threshold under sparse networks	[83], [84], [85]
weak consistency	[86], [87]
non-trivial recovery	[88], [89], [90]
robustness against outlier nodes	[79], [91], [92]
consistency under degree-corrected models	[31]
consistency under weak assortativity	[93], [94]

6. Experimental part

Many different algorithms have been proposed for network community detection, as mentioned in previous sections. Each of these has its own properties depending on the nature of the network as well as the domain of the problem. This section introduces an implementation of two different methods: the first is one of the most commonly used computational methods, the fast algorithm given by [60], while the second is the convexified modularity maximization algorithm (CMM), which uses semidefinite programming.

The fast algorithm for community detection is implemented as follows: Beginning by determining the partition of some different networks with the data set in Table 2, then measuring the goodness of that partition using quality functions defined in Section 3, the results are listed in Table 5.

Table 5: Results of quality function for fast partitioning algorithm

Network	modularity	Local density	Global density	Distance based	Node membership
Karate	0.3807	0.6206	0.6096	0.3062	0.2668
Dolphin	0.4923	0.6015	0.5947	0.2706	0.2179
American Football College	0.5682	0.7218	0.6920	0.1542	0.4587
Facebook	0.7856	0.5670	0.5371	0.1301	0.1344
Political Blogs	0.4407	0.5232	0.5154	0.4197	0.0484
Protein Protein	0.5709	0.5244	0.5077	0.1279	0.0516
AS-Level Internet	0.6285	0.5062	0.5022	0.0937	0.0139
Chesapeaka Synthetic	0.249	0.6323	0.6249	0.2978	0.3191
Delaunay Synthetic	0.732	0.5170	0.5143	0.2011	0.0349
Twitter	0.5820	0.5520	0.5170	0.1478	0.1111

The results of implementing CMM using the ADMM algorithm are listed in Table 6 below. It computes the modularity function to plant partitions on various networks, real and synthetic, given in Table 2, with K=2 using algorithm 2. Also, arbitrarily, l=200 iterations were chosen with ρ=1.

Table 6: Result modularity for partitioning using ADMM algorithm

Network	modularity
Karate	0.3715
Dolphin	0.3848
American Football College	0.2077
Facebook	0.3059
Political Blogs	0.4351
Protein Protein	0.1440
AS-Level Internet	0.1458
Chesapeaka Synthetic	0.1959
Delaunay Synthetic	0.3946
Twitter	0.2185

7. Conclusion

Modularity was utilized in the community detection problem on two levels. First of all, it quantified the goodness of a given network partition by giving the value of quantity Q even without any information about the actual communities of the network, and this is appropriate for very large networks. The second level of modularity usage in the graph splitting task was shown by algorithms for detecting communities that were based on the maximization of modularity. Although it suffered from some problems and limitations, modularity maximization was a very creative field to research. This paper summarized many community detection approaches based on modularity, as given in Table 3. Besides being explained in detail, the most fundamental topics concern the modularity function and other quality functions and their application, as given in Table 5. In this table, the values of the quality function were computed for the case of using the fast algorithm introduced by Newman, and in the case of using other methods of community detection, the same quality functions may be used. So this review was considered a good starting point for researchers to be able to find the best methods and choose a suitable quality function to delve deeper into this interesting subject.

The focus of researcher communities on dynamic approaches has been started once the concept of time and order integrated networks attributes as important characteristics of nodes (vertices) and links (edges). Meanwhile, dynamic approaches still under exploration and promise good performances, taking into consideration changes of characteristics of real-world complex network. In fact, most real-world complex networks in nature are by default dynamic, which makes sense to develop more efficient dynamic approaches, through handling dynamic communitydetection problem. As features of this work, an important challenge taking place for the dynamic approaches and attracting attention of researchers either in combining existing methods or in exploring limits of others. We focus on the next work on adaptation of existing methods to achieve community detection challenge for dynamic evolving networks.

References

[1] B.W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 291–307, 1970.

[2] E. R. Barnes, "An Algorithm for Partitioning the Nodes of a Graph," *SIAM J. Algebr. Discret. Methods*, vol. 3, no. 4, pp. 541–550, 1982, doi: 10.1137/0603056.

[3] J. F. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition, Springer, 2009.

[4] S. K. Mohammed, M. M. Taha, E. M. Taha, and M. N. A. Mohammad, "Cluster Analysis of Biochemical Markers as Predictor of COVID-19 Severity," *Baghdad Sci. J.*, vol. 19, no. 6, pp. 1423–1429, 2022, doi: 10.21123/BSJ.2022.7454.

[5] S. A. S. Al-Sabbah, B. A. R. Qasim, and A. M. Shareef, "Using the Hierarchical Cluster Analysis and Fuzzy Cluster Analysis Methods for Classification of Some Hospitals in Basra," *Baghdad Sci. J.*, vol. 18, no. 4, pp. 1212–1217, 2021, doi: 10.21123/BSJ.2021.18.4.1212.

[6] A. D. Abbood, A. A. Hasan, R. K. Everson, and A. Attea, "A Multi-Objective Evolutionary Algorithm with Neighbor Node Centrality for Community Detection in Complex Networks", 2022. available at: https://europepmc.org/article/ppr/ppr572860.

[7] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceeding Fifth Berkeley Symp. Math. Stat. Probab.*, vol. 1, pp. 281–297, 1967.

[8] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslov. Math. J.*, vol. 23, no. 298, 1973.

[9] A. Jahanbani, "Lower bounds for the energy of graphs," *AKCE Int. J. Graphs Comb.*, vol. 15, no. 1, pp. 88–96, 2018, doi: 10.1016/j.akcej.2017.10.007.

[10] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 69, no. 2 2, pp. 1–16, 2004, doi: 10.1103/PhysRevE.69.026113.

[11] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 74, no. 3, pp. 1–19, 2006, doi: 10.1103/PhysRevE.74.036104.

[12] R. L. Winkl, "An Introduction to Bayesian Inference and Decision," Probabilistic Publishing, 2nd edition, 2003.

[13] V. Batagelj, A. Ferligoj, and P. Doreian, *Generalized blockmodeling*, vol. 23, no. 4. Cambridge university press, 2005. doi: 10.1017/cbo9780511584176.

[14] K. P. B. D. R. Anderson, Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd Edition, Springer, 2002.

[15] F. Hassan and S. F. Behadili, "Modeling Social Networks using Data Mining Approaches-Review," *Iraqi Journal of Science*, vol. 63, no. 3, pp. 1313–1338, 2022, doi: 10.24996/ijs.2022.63.3.35.

[16] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3–5, pp. 75–174, 2010, doi: 10.1016/j.physrep.2009.11.002.

[17] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Stat. Anal. Data Min.*, vol. 4, no. 5, pp. 512–546, 2011, doi: 10.1002/sam.10133.

[18] A. Karatas and S. Sahin, "Application Areas of Community Detection: A Review," *Int. Congr. Big Data, Deep Learn. Fight. Cyber Terror. IBIGDELFT 2018 - Proc.*, no. December 2018, pp. 65–70, 2019, doi: 10.1109/IBIGDELFT.2018.8625349.

[19] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, 2016, doi: 10.1016/j.physrep.2016.09.002.

[20] P. Ponveni and J. Visumathi, "A Review on Community Detection algorithms and Evaluation Measures in Social Networks," *2021 7th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2021*, pp. 1892–1899, 2021, doi: 10.1109/ICACCS51430.2021.9441868.

[21] U. Von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007, doi: 10.1007/s11222-007-9033-z.

[22] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Soc. Networks*, vol. 5, no. 2, pp. 109–137, 1983, doi: 10.1016/0378-8733(83)90021-7.

[23] E. Abbe and C. Sandon, "Community Detection in General Stochastic Block models: Fundamental Limits and Efficient Algorithms for Recovery," *Proc. - Annu. IEEE Symp. Found. Comput. Sci. FOCS*, vol. 2015-Decem, pp. 670–688, 2015, doi: 10.1109/FOCS.2015.47.

[24] Al-Dayyeni, Amenah Dahim Abbood. *Community Detection in Complex Networks*. 2018. PhD Thesis. University of Exeter.

[25] S. Fortunato and C. Castellano, "Community structure in graphs," *Comput. Complex. Theory, Tech. Appl.*, vol. 9781461418, pp. 490–512, 2012, doi: 10.1007/978-1-4614-1800-9_33.

[26] B. A. Attea and H. S. Khoder, "A new multi-objective evolutionary framework for community mining in dynamic social networks," *Swarm Evol. Comput.*, vol. 31, no. August, pp. 90–109, 2016, doi: 10.1016/j.swevo.2016.09.001

[27] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 1, pp. 36–41, 2007, doi: 10.1073/pnas.0605965104.

[28] S. Fortunato, "Quality functions in community detection," *Noise Stochastics Complex Syst. Financ.*, vol. 6601, p. 660108, 2007, doi: 10.1117/12.726703.

[29] S. Sobolevsky, R. Campari, A. Belyi, and C. Ratti, "General optimization technique for high-quality community detection in complex networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 90, no. 1, 2014, doi: 10.1103/PhysRevE.90.012811.

[30] Y. Chen, L. I. Xiaodong, and J. Xu, "Convexified modularity maximization for degree-corrected stochastic block models," *Ann. Stat.*, vol. 46, no. 4, pp. 1573–1602, 2018, doi: 10.1214/17-AOS1595.

[31] X. He, R. Zhang, and B. Zhu, "A Generalized Modularity for Computing Community Structure in Fully Signed Networks," *Complexity*, vol. 2023, 2023, doi: 10.1155/2023/8767131.

[32] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.*, vol. 70, no. 6, p. 6, 2004, doi: 10.1103/PhysRevE.70.066111.

[33] M. Mitalidis and L. Pitsoulis, "Manual for the Community Detection Toolbox v . 0 . 9," no. February, 2014. https://usermanual.wiki/Document/Manual.861148111/help

[34] N. Gaumont, F. Queyroi, C. Magnien, and M. Latapy, "Expected nodes: A quality function for the detection of link communities," *Stud. Comput. Intell.*, vol. 597, pp. 57–64, 2015, doi: 10.1007/978-3-319-16112-9_6.

[35] G. Chartrand, "*A first course in graph theory*". *Courier Corporation*, 2013.

[36] N. Alotaibi and D. Rhouma, "A review on community structures detection in time evolving social networks," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5646–5662, 2022, doi: 10.1016/j.jksuci.2021.08.016.

[37] R. van der Hofstad, "Random graphs and complex networks," *Random Graphs and Complex Networks*, vol. 1, pp. 1–321, 2016, doi: 10.1017/9781316779422.

[38] R. Solomonoff and A. Rapoport, "Connectivity of random nets," *Struct. Dyn. Networks*, vol. 9781400841, pp. 27–37, 1951, doi: 10.1515/9781400841356.27.

[39] P. Erdös and A. Rėnyi, "on Random Graph," *Publ. Math.*, vol. 6, pp. 290–297, 1959.

[40] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. SUPPL. 1, pp. 2566–2572, 2002, doi: 10.1073/pnas.012582999.

[41] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. U.*

S. A., vol. 103, no. 23, pp. 8577–8582, 2006, doi: 10.1073/pnas.0601602103.
[42] M. E. J. Newman, "Analysis of weighted networks," *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.*, vol. 70, no. 5, p. 9, 2004, doi: 10.1103/PhysRevE.70.056131.
[43] E. A. Leicht and M. E. J. Newman, "Community structure in directed networks," *Phys. Rev. Lett.*, vol. 100, no. 11, pp. 1–4, 2008, doi: 10.1103/PhysRevLett.100.118703.
[44] A. Arenas, J. Duch, A. Fernández, and S. Gómez, "Size reduction of complex networks preserving modularity," *New J. Phys.*, vol. 9, 2007, doi: 10.1088/1367-2630/9/6/176.
[45] H. Shen, X. Cheng, K. Cai, and M. Bin Hu, "Detect overlapping and hierarchical community structure in networks," *Phys. A Stat. Mech. its Appl.*, vol. 388, no. 8, pp. 1706–1712, 2009.
[46] U. Brandes *et al.*, "On Modularity Clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 2, pp. 172–188, 2008.
[47] A. F. Yaqoob, "Community detection in big social networks using multiple soft computing techniques," *Master thesis*, College of Science, University of Baghdad, 2019.
[48] M. E. J. Newman, *Networks An Introduction*. OXFORD university press, 2018.
[49] E. Ziv, M. Middendorf, and C. H. Wiggins, "Information-theoretic approach to network modularity," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 71, no. 4, pp. 1–9, 2005, doi: 10.1103/PhysRevE.71.046117.
[50] U. Brandes *et al.*, "Maximizing Modularity is hard," ArVix, September 2006. https://doi.org/10.48550/arXiv.physics/0608255
[51] U. Brandes *et al.*, "On Modularity: NP-Completeness and Beyond," no. 001907, p. 33, 2006.
[52] S. M. Mula and G. Veltri, "A new measure of modularity density for community detection," 2019, [Online]. Available: http://arxiv.org/abs/1908.08452
[53] A. Noack and R. Rotta, "Multi-level algorithms for modularity clustering," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5526 LNCS, pp. 257–268, 2009, doi: 10.1007/978-3-642-02011-7_24.
[54] M. MacMahon and D. Garlaschelli, "Community detection for correlation matrices," *Phys. Rev. X*, vol. 5, no. 2, pp. 1–35, 2015, doi: 10.1103/PhysRevX.5.021006.
[55] M. J. Barber, "Modularity and community detection in bipartite networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 76, no. 6, pp. 1–9, 2007, doi: 10.1103/PhysRevE.76.066102.
[56] S. S. Abdul-ameer, "An Evolutionary Bi - clustering Algorithm for Community Detection in Social Networks", *Iraqi Journal of Science*, vol. 57, no. 3, pp. 2111–2120, 2015.
[57] P. Expert, T. S. Evans, V. D. Blondel, and R. Lambiotte, "Uncovering space-independent communities in spatial networks," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 108, no. 19, pp. 7663–7668, 2011, doi: 10.1073/pnas.1018962108.
[58] V. A. Traag and J. Bruggeman, "Community detection in networks with positive and negative links," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 80, no. 3, pp. 1–6, 2009, doi: 10.1103/PhysRevE.80.036115.
[59] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 12, pp. 7821–7826, 2002, doi: 10.1073/pnas.122653799.
[60] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.*, vol. 69, no. 6, p. 5, 2004, doi: 10.1103/PhysRevE.69.066133.
[61] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, 2008, doi: 10.1088/1742-5468/2008/10/P10008.
[62] P. Schuetz and A. Caflisch, "Multistep greedy algorithm identifies community structure in real-world and computer-generated networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 78, no. 2, pp. 1–7, 2008, doi: 10.1103/PhysRevE.78.026112.
[63] Z. Ye, S. Hu, and J. Yu, "Adaptive clustering algorithm for community detection in complex networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 78, no. 4, pp. 1–6, 2008, doi: 10.1103/PhysRevE.78.046115.
[64] B. Xiang, E. H. Chen, and T. Zhou, "Finding community structure based on subgraph similarity," *Stud. Comput. Intell.*, vol. 207, pp. 73–81, 2009.
[65] L. Waltman and N. J. Van Eck, "A smart local moving algorithm for large-scale modularity-based community detection," *Eur. Phys. J. B*, vol. 86, no. 11, pp. 1–33, 2013, doi:

10.1140/epjb/e2013-40829-0.
[66] R. Guimerà and L. A. N. Amaral, "Cartography of complex networks: Modules and universal roles," *J. Stat. Mech. Theory Exp.*, no. 2, pp. 1–13, 2005.
[67] A. F. Yaqoob and B. Al-Sarray, "Community detection under stochastic block model likelihood optimization via tabu search –Fuzzy C-mean method for social network data," *Iraqi Journal of Science*, vol. 61, no. 10, pp. 2695–2704, 2020, doi: 10.24996/ijs.2020.61.10.27.
[68] S. Boettcher and A. G. Percus, "Optimization with extremal dynamics," *Phys. Rev. Lett.*, vol. 86, no. 23, pp. 5211–5214, 2001, doi: 10.1103/PhysRevLett.86.5211.
[69] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 72, no. 2, pp. 2–5, 2005.
[70] J. Ruan and W. Zhang, "An efficient spectral algorithm for network community discovery and its applications to biological and social networks," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, no. November 2007, pp. 643–648, 2007, doi: 10.1109/ICDM.2007.72.
[71] G. Wang, Y. Shen, and M. Ouyang, "A vector partitioning approach to detecting community structure in complex networks," *Comput. Math. with Appl.*, vol. 55, no. 12, pp. 2746–2752, 2008, doi: 10.1016/j.camwa.2007.10.028.
[72] T. Richardson, P. J. Mucha, and M. A. Porter, "Spectral tripartitioning of networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 80, no. 3, pp. 1–12, 2009.
[73] A. A. Amini and E. Levina, "On semidefinite relaxations for the block model1," *Ann. Stat.*, vol. 46, no. 1, pp. 149–179, 2018, doi: 10.1214/17-AOS1545.
[74] X. Li, Y. Chen, and J. Xu, "Convex Relaxation Methods for Community Detection," *Stat. Sci.*, vol. 36, no. 1, pp. 2–15, 2021, doi: 10.1214/19-STS715.
[75] X. Li, L. Cai, J. Li, C. K. W. Yu, and Y. Hu, "A survey of clustering methods via optimization methodology," *J. Appl. Numer. Optim.*, vol. 3, no. 1, pp. 151–174, 2021.
[76] T. T. Cai and X. Li, "Robust and computationally feasible community detection in the presence of arbitrary outlier nodes," *Ann. Stat.*, vol. 43, no. 3, pp. 1027–1059, 2015.
[77] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," *Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011*, no. 1, pp. 1–9, 2011.
[78] R. Vidal, Y. Ma, and S. S. Sastry, "Robust principal component analysis," *Interdiscip. Appl. Math.*, vol. 40, pp. 63–122, 2016, doi: 10.1007/978-0-387-87811-9_3.
[79] Y. Chen, A. Jalali, S. Sanghavi, and H. Xu, "Clustering partially observed graphs via convex optimization," *J. Mach. Learn. Res.*, vol. 15, pp. 2213–2238, 2014.
[80] Y. Chen, S. Sanghavi, and H. Xu, "Clustering sparse graphs," *Adv. Neural Inf. Process. Syst.*, vol. 3, pp. 2204–2212, 2012.
[81] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010, doi: 10.1561/2200000016.
[82] Y. Chen and J. Xu, "Statistical-Computational Phase Transitions in Planted Models : The High-Dimensional Setting," *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 2, pp. 244-252, 2014.
[83] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 471–487, 2016, doi: 10.1109/TIT.2015.2490670.
[84] B. Hajek, Y. Wu, and J. Xu, "Achieving exact cluster recovery threshold via semidefinite programming," *IEEE Int. Symp. Inf. Theory - Proc.*, vol. 2015-June, pp. 1442–1446, 2015.,
[85] A. S. Bandeira, "Random Laplacian Matrices and Convex Relaxations," *Found. Comput. Math.*, vol. 18, no. 2, pp. 345–379, 2018, doi: 10.1007/s10208-016-9341-9.
[86] A. Perry and A. S. Wein, "A semidefinite program for unbalanced multisection in the stochastic block model," *2017 12th Int. Conf. Sampl. Theory Appl. SampTA 2017*, pp. 64–67, 2017, doi: 10.1109/SAMPTA.2017.8024399.
[87] O. Guédon and R. Vershynin, "Community detection in sparse networks via Grothendieck's inequality," *Probab. Theory Relat. Fields*, vol. 165, no. 3–4, pp. 1025–1049, 2016.
[88] Y. Fei and Y. Chen, "Exponential error rates of SDP for block models: Beyond grothendieck's inequality," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 551–571, 2019.
[89] A. Montanari and S. Sen, "Semidefinite programs on sparse random graphs and their application

to community detection," *Proc. Annu. ACM Symp. Theory Comput.*, vol. 19-21-June, pp. 814–827, 2016, doi: 10.1145/2897518.2897548.

[90] Z. Fan and A. Montanari, "How well do local algorithms solve semidefinite programs?," *Proc. Annu. ACM Symp. Theory Comput.*, vol. Part F1284, no. June, pp. 604–614, 2017.

[91] S. Mei, T. Misiakiewicz, A. Montanari, and R. I. Oliveira, "Solving SDPs for synchronization and MaxCut problems via the Grothendieck inequality," Annual Conference Computational Learning Theory *(2017)*, vol. 65, pp. 1–40, 2017.

[92] A. Moitra, W. Perry, and A. S. Wein, "How robust are reconstruction thresholds for community detection?," *Proc. Annu. ACM Symp. Theory Comput.*, vol. 19-21-June, pp. 828–841, 2016, doi: 10.1145/2897518.2897573.

[93] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan, "Learning communities in the presence of errors," *J. Mach. Learn. Res.*, vol. 49, no. June, pp. 1258–1291, 2016.

[94] B. Yan, P. Sarkar, and X. Cheng, "Provable estimation of the number of blocks in block models," *Int. Conf. Artif. Intell. Stat. AISTATS 2018*, vol. 84, pp. 1185–1194, 2018.