Community Detection in Networks using Bio-inspired Optimization: Latest Developments, New Results and Perspectives with a Selection of Recent Meta-Heuristics

Eneko Osaba[a,*], Javier Del Ser[a,b,c], David Camacho[d], Miren Nekane Bilbao[b], and Xin-She Yang[e]

[a] *TECNALIA Research & Innovation, 48160 Derio, Spain*
[b] *University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain*
[c] *Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain*
[d] *Universidad Politecnica de Madrid, 28031 Madrid, Spain*
[e] *School of Science and Technology, Middlesex University, London, United Kingdom*

Abstract

Detecting groups within a set of interconnected nodes is a widely addressed problem that can model a diversity of applications. Unfortunately, detecting the optimal partition of a network is a computationally demanding task, usually conducted by means of optimization methods. Among them, randomized search heuristics have been proven to be efficient approaches. This manuscript is devoted to providing an overview of community detection problems from the perspective of bio-inspired computation. To this end, we first review the recent history of this research area, placing emphasis on milestone studies contributed in the last five years. Next, we present an extensive experimental study to assess the performance of a selection of modern heuristics over weighted directed network instances. Specifically, we combine seven global search heuristics based on two different similarity metrics and eight heterogeneous search operators designed ad-hoc. We compare our methods with six different community detection techniques over a benchmark of 17 Lancichinetti-Fortunato-Radicchi network instances. Ranking statistics of the tested algorithms reveal that the proposed methods perform competitively, but the high variability of the rankings leads to the main conclusion: no clear winner can be declared. This finding aligns with community detection

*Corresponding author: TECNALIA. Parque Tecnologico, Ed. 700, 48160 Derio, Bizkaia, Spain. Telephone: (+34) 946 430 850. Fax: (+34) 901 760 009

Email address: `eneko.osaba@tecnalia.com` (Eneko Osaba)

tools available in the literature that hinge on a sequential application of different algorithms in search for the best performing counterpart. We end our research by sharing our envisioned status of this area, for which we identify challenges and opportunities which should stimulate research efforts in years to come.

Keywords: Bio-inspired Computation, Community Detection, Network Partition, Evolutionary Computation, Swarm Intelligence

---

1. Introduction

Since the ground-breaking advent of Social Networks, a spectrum of tools and methods have been developed in the last decade for excerpting insights from the multiple interrelations between their users [16]. Knowledge that can be extracted by using these methods ranges from the evaluation of the influence of a specific node in the whole network (*centrality*), to enriched ways of visualizing graphs or the discovery of shortest paths between groups of nodes. As can be drawn from the related literature, all such knowledge can be exploited for a myriad of practical objectives, such as the inference of radicalization risk [96, 95, 52, 46], the identification of child abuse [179] or the detection of impersonation [172].

Among the valuable information that can be extracted from Social Networks, the detection of communities within their constituent nodes is one of the most frequently addressed tasks in the related literature stream. Specifically, a *community* refers to a group of elements which meet the general principles of strong intra-community connectivity (i.e. members of the same community are strongly tied to each other) and weak inter-community connectivity (nodes belonging to different partitions are loosely connected). Such a measure of connectivity varies depending on the characteristics of the network at hand (e.g. weighted, multiple edges, directed and/or with self-loops), thereby reformulating the quantification of the cohesiveness of any community. In this regard, different efficient metrics have been proposed in the literature for evaluating the quality of a partition of a given network. Each of these metrics relies on the aforementioned connectivity principles from different perspectives, yielding a partition quality indicator that eventually serves as a optimization fitness function. As such, alternatives such as Surprise [2], Permanence [29], or the renowned Girvan-Newman coefficient of modularity [133] are arguably among the most recurrently utilized partition metrics. From an algorithmic standpoint, the literature has been also rich in regards to different optimization strategies for discovering partitions by maximizing the aforementioned partition quality indicators. Interestingly under the scope of this paper, meta-heuristic techniques with biological inspiration at their core have emerged as efficient algorithmic means to undertake community partition problems over network instances of relative complexity, with Evolutionary Algorithms as the most resorted optimization techniques in last years [143].

Interestingly, activity around bio-inspired optimization has been particularly active in proposing modern solvers which, by mimicking new biological phenomena, have been empirically proven to outperform traditional optimization methods in a plethora of application scenarios. This is the case, for instance, of the Bat Algorithm (BA, [185]), Firefly Algorithm (FA, [184]) or Cuckoo Search (CS, [186]), three exemplifying bio-inspired solvers that have spurred a flurry of research since their inception. A similar rationale can be drawn when it comes to physical processes observed in Nature, which have forged the more general family of nature-inspired optimization techniques: the Water Cycle Algorithm (WCA, [42]), for instance, is a good exponent of this alternative algorithmic branch. From this angle, several comprehensive surveys dealing with applications of bio- and nature-inspired optimization techniques have been reported to date, with diverse levels of coverage and depth [188, 187, 47]. However, to the best of our knowledge the current literature lacks a deep analysis synthesizing the state of the art around the application of sophisticated bio-inspired heuristics to community partition problems in networks, elaborating on the transition from the use of classical methods – e.g. Genetic Algorithms (GA, [53]), Particle Swarm Optimization (PSO, [87]) and the like – to the progressive adoption of modern meta-heuristics as the ones exemplified above. Definitely, a systematic study on the design trends and challenges lying at this intersection is needed, for the community to gather research efforts in niches of opportunity and value within this vibrant area.

1.1. Objective and Contribution

The motivation introduced above is reflected in the goal of the work presented in this manuscript. To begin with, we delve into the state of the art around the use of bio-inspired optimization approaches for community partition in complex networks, critically examining some of the most important works in recent times. Specifically, we focus our attention on community detection algorithms relying on nature- and bio-inspired meta-heuristic solvers published in the last five years (2015-2019). This baseline literature study serves as a stepping stone towards the second contribution of this work: an experimental benchmark composed by a selection of modern bio-inspired solvers for finding communities in weighted directed graphs. This class of networks has been far less studied than other graphs despite its straightforward applicability to real-world scenarios, such as the design of network protocols [113], relationships in the control structure of financial networks [13], trade imbalance relationships between importers and exporters [158], or interactions in social network analysis [103]. To this end, we give design rationale on how to adapt search-based heuristic operators to the specific characteristics of this problem, yielding a portfolio of 8 heterogeneous search operators based on diverse design principles (from ad-hoc heuristics to blind movement patterns). Furthermore, we resort to the *label-based representation* [75] for solution encoding, and adopt the Hamming Distance and the Normalized Mutual Information (NMI) as metrics to evaluate the similarity between different candidate partitions. The diversity of the benchmark is augmented by also considering different global search mechanisms for the benchmark, based on a selection of seven meta-heuristic schemes: the aforementioned WCA, BA, FA, and CS, along with off-the-shelf and hybrid solvers hinging on classical techniques from bio- and nature-inspired computation: PSO and Evolutionary Simulated Annealing (ESA, [190]). Finally, the benchmark also comprises a population-based variant of a classical non-biologically-inspired heuristic (Population-based Variable Neighborhood Search, PVNS, [175]) for the sake of completeness with respect to the wide field of random search heuristics. We note that some of them, such as WCA or ESA, have never been applied to community finding in weighted directed graphs. Part of the remaining ones (BA, FA and CS) have been less studied in the literature in comparison with other heuristic counterparts. We give in this manuscript a thorough description on how each of the 19 implemented solvers has been adapted to efficiently solve the modeled problem, along with deep details of the considered operators and functions, and a justification of their expected benefits in terms of convergence.

With the aim of comparatively assessing their performance, results obtained over 29 synthetically generated network instances of diverse size are discussed. The comparison is made on the basis of their capability to discover their true partition, which is known given the particular procedure used for generating the networks. In addition, the convergence behavior of the best solvers is also analyzed to gain an intuition of the performance gaps noted among algorithms. Finally, the significance of such performance gaps is statistically verified by means of the Friedman's non-parametric and Holm's post-hoc tests. Results reveal that BA with heuristic operators and using NMI as its distance function dominates the benchmark with statistical relevance.

Moreover, we have conducted an additional set of tests in order to certify that bio-inspired computation schemes can perform competitively with respect to other established techniques for community detection. To this end, we have compared the results obtained by our four best bio-inspired solvers with six different community detection methods from the literature. This third experimentation has been performed over 17 diverse LFR instances [94] composed by 100 to 600 nodes. Results of this last experimentation reveal that bio-inspired approaches present a promising performance, being especially effective in instances with high values of their topological mixing coefficient. Finally, we draw attention to a number of identified research trends that are grasping the attention of researchers in this area. In light of these trends, a prospect of challenges and niches is outlined, along with possible technical directions that could be pursued by the community to advance over the current state of the art.

The present work is an extension of the preliminary work reported in [136], yet comprising new contributed material that justifies its novelty and soundness. The first is the review of the state of the art conducted as an introduction to the experimental benchmark, in which we thoroughly examine recently proposed contributions and coherently relate each other towards identifying commonalities and trends. With this new content, the reader not only follow the state-of-the-art developments of bio-inspired computation and community detection in networks, but also gain greater insights into different methods to be compared in the second part of this paper. From the algorithmic perspective, the research scope has been extended over [136] by considering three additional meta-heuristic schemes: PSO, CS and BA. Furthermore, one additional distance metric has been developed, and a set of new heuristic movement operators has been implemented. In overall, 15 new algorithmic configurations have been added to the experimentation. Besides that, additional network instances of increased size have been considered in the experimentation. It is also worth mentioning that the analysis of the results goes beyond the statistical assessment of the partition quality values attained by each scheme, to include as well as a convergence analysis of the best techniques in the benchmark. Moreover, an experimental comparison to 6 community detection methods from the literature has been conducted over 17 LFR instances. Finally, an equally important novel contribution is our personal envisioned status of this area, which we present in the form of challenges and open opportunities that remain insufficiently addressed to date. The above three different aspects are the main contributions of our present paper, which makes it more comprehensive and complete than our previous works.

The rest of the paper is structured as follows: Section 2 elaborates on the first contribution of the paper by analyzing the state of the art of the central topic of the paper. In Section 3 the problem of detecting communities in weighted directed networks is mathematically formulated, whereas the considered heuristic solvers and their implementation details are described in Section 4. The experimental setup is detailed in Section 5, along with a discussion on the obtained results. Research opportunities for the area are highlighted in Section 6. Finally, Section 7 concludes the paper with a general outlook for the wide audience.

2. Recent Work in Community Detection using Bio-inspired Meta-heuristics

A short glimpse to the related literature reveals the increasing importance of the community detection field in the scientific panorama. As mentioned in the introduction, the widespread societal impact of Social Networks lit the wick of the growing interest in this area, strongly linked to the valuable knowledge that can be drawn from community structures. This statement is buttressed by the amount of comprehensive surveys published in recent times, such as [90] which focuses on multi-layer networks. This kind of graphs are comprised of multiple interdependent sub-networks, each representing a different aspect of the interactions between nodes. It is also interesting to mention the thorough review in [150] related to dynamic networks. Wider is the approach of the overview recently contributed in [80], stressing on community detection algorithms for disjoint and overlapping communities, along with related multidisciplinary applications. In this same sense disjoint communities are also the focus of the work in [35], mainly dived for techniques for inferring non-overlapping communities in large-scale real-world undirected and directed networks. Especially interesting for the scope of the present paper is the review recently published by Pizzuti in [143]. In that research, the author describes in depth Evolutionary Computation methods to unveil community structures in networks. Special attention is paid to solution representation (*encoding*) strategies and popular partition quality indicator functions adopted as fitness metrics to be optimized. This survey also examines different problem formulations in this regard, from multiple and single objectives to different graph topologies, such as dynamically evolving, multidimensional and signed graphs.

It is interesting to mention that our research work presented in this paper builds upon prior contributions [143] in several novel directions, among which the most remarkable one is the practical study and experimentation conducted and discussed in Sections 4 and 5. Furthermore, we place special attention on connecting the insights and findings drawn from the novel experimentation study with our prospects on the confluence between bio-inspired optimization and community detection in networks. To this end, we thoroughly describe several challenges and open opportunities that should guide the activity around this intersecting research avenue in the next years. Additional recent review works dealing with community detection can be found in [26, 199, 14, 88].

We build upon the momentum of this research area as evinced above to critically examine recently published advances. Algorithmically speaking, many contributions have so far gravitated on the development of different approaches to find communities towards implicitly or explicitly optimizing one of the aforementioned partition quality indicators. This is the case, for instance, of iterative greedy methods capable of inferring a hierarchy of communities in a constructive fashion, similarly to agglomerative hierarchical clustering techniques [20]. A technique of this kind is also used in [155], which emphasizes on large social networks retrieved from the Stanford Network Analysis Project [99]; specifically, 973 ego networks from Twitter and 10 from Facebook, with the number of nodes ranging from 5 to 60 050, and from 10 to 1045, respectively. It is also worth-mentioning the findings reported in [183], in which the Girvan-Newman modularity (the same quality indicator function as the one adopted in our study) is used as the objective function for a two-step optimization technique called DiMod, composed by two mathematical programming models that rely on differently rearranged versions of the modularity to stress on different features of the underlying community structure. Another two-stage solver is proposed in [76], in which the community finding is done over signed networks. In that paper, a Symmetric Nonnegative Matrix Factorization-based Propagation method is proposed. The first stage of this approach is to carry out a symmetric nonnegative matrix factorization on its positive part, and associate each node with an initial group indication vector. The second phase conducts a diffusion process to guarantee that these indication respect the topology of the entire network and preserve their initial values at the same time. Logically, nodes in the same community have similar indication vectors, while they differ in vertices that most likely reside in different communities. Thus, final indication vectors provide a satisfactory partition of the graph, and can be employed to assign elements into communities. Finally, we want to spotlight one last related work, which application is based on bipartite large-scale networks [166]. Bipartite graphs can be divided into two disjoint groups, $\mathcal{G}_\top$ and $\mathcal{G}_\perp$, such that every link connects a node in $\mathcal{G}_\top$ to one in $\mathcal{G}_\perp$. Authors of that research introduce an algorithm called ComSim, which is based on a similarity measure between nodes exploiting the bipartite connections. The proposed method seeks cycles of links maximizing the similarity between vertices, defining in this manner the core of the discovered communities.

Beyond the ad-hoc heuristics for community finding reviewed above, a growing strand of literature currently gravitates on the application of bio-inspired meta-heuristic methods directly adopting a partition quality indicator as their objective function. Examples are many, each focusing on assorted combinations of network instances, metric functions and algorithmic approximations. Arguably, Genetic Algorithms (GAs) are among the most recurrently explored ones for discovering communities in networks of different characteristics. In [66], for example, a novel generational GA is proposed, which is guided by the modularity index, and which introduces efficient initialization strategies and search operators. An additional GA-based method is proposed by Said et al. in [153], which introduces a Clustering Coefficient-based GA which not only detects cohesive groups from dense networks, but also identifies communities in sparse graphs. The main philosophy behind this proposal is to use a social network analysis measure to generate the initial population [177]. Another GA-based approach is presented in [39] for similar purposes, in which authors adopt label propagation for creating the initial population, and conduct an anti-destructive one-way crossover. Moreover, for improving the search efficiency, authors implement a node-local optimization strategy as a means to perform a tailored mutation process over evolved solutions. Specially interesting is the work in [67], published recently. This contribution proposes a GA comprising two different novel ingredients: 1) a strategy based on local structural similarity and roulette wheel selection for the generation of the initial population; and 2) a new mutation operator based on label propagation and local structural similarity. The efficiency of this GA-based community detection algorithm has been tested over synthetic and real-world networks, and compared to additional state-of-the-art methods.

Other interesting works were previously reported in [74] and [161]. The main contribution of the first one is the development of a multi-individual ensemble learning-based crossover function, which builds an offspring through the use of a hierarchical agglomerative clustering approach. On the other hand, [161] proposed the adaptation of the well-known two-point crossover, confirming its promising performance also for this context. The algorithmic approach proposed in [161] was later extended by Morada and Parsa in [125] by a novel local search strategy to improve the accuracy of the algorithm and to speed up its convergence. Additional works can be found in [102] and [168].

Besides GA, the history of bio-inspired meta-heuristics for community partition has also placed other techniques under its spotlight, PSO or Ant Colony Optimization (ACO):

- Regarding PSO, one of the most influential works can be found in [25], in which a discrete PSO was developed for finding communities in signed networks through the optimization of the signed modularity. A similar research is proposed by the same corresponding author in [24], in that case focused on large-scale social network clustering. Moreover, in [146] a multi-objective solver relying on a modified variant of this scheme is implemented, namely, MOPSO-Net. The main novelty of this proposed solver hinges on the modification of the particles' moving strategy, which is endowed with elements from genetically inspired operators (i.e., crossover and mutation). In [55], another discrete multi-objective PSO, termed MODPSO, is proposed. The main novel ingredients of MODPSO is the use of a specific solution encoding, as well as the redefinition of the velocity concept that drives the search of this meta-heuristic solver. Objectives to minimize in this latter work coincide with those in [146]. An alternative multi-objective formulation of the community detection problem is tackled in [107] through a PSO based solution. In this case, graphs under study are the above mentioned signed networks. Finally, a PSO-based approach is proposed in [100] for solving the same multi-objective community finding problem. The network clustering algorithm implemented in this work is referred to as quantum-behaved discrete multi-objective PSO, with parallelization and the automatic determination of the number of communities as novel contributions with respect to preceding literature.

- Likewise, several ACO-based schemes have emerged in recent years, mainly due to the suitability of this particular global search meta-heuristic to undertake problems in graphs.Two of the first adaptations of this meta-heuristic algorithm were presented in [32] and [73]. Several studies have been developed thereafter. We focusing our attention in recent works,such as [62]. In this study two ACO-based variants are proposed for being applied to ego networks, where the central node of the graph (*ego*) represents the focal user under study. Thus, methods to be developed aim to automatically determine the different users that compose groups or circles of interest around the ego node. Both ACO-based techniques in [62] differ in the source of information used to perform the community finding task. While one of them employs the knowledge drawn from the topology of the graph, the second ACO takes into account the information contained in the user profile. Further stimulating research can be found in [205], focused on the detection of overlapping communities in complex networks. Although the technique proposed in this paper (named as AntCBO) shares many similarities with other ACO-based approaches from the literature, a point of novelty resides in its post-processing phase, which is executed to naturally achieve a final overlapping community structure.

More recently, the community has started to explore the suitability of modern nature-inspired meta-heuristics for community detection in graphs. One of these methods is BA, as exposed by works such as [72] and [162]. Particularly relevant are [120] and [207], which focus on the application of BA over dynamic social networks, and in which a multi-objective community finding problem is formulated. Specifically, BA simultaneously optimizes the modularity density and the NMI as objective functions in both references. FA has also been adapted for dealing with community finding problems: in [37] the behavioral patterns of fireflies are emulated in the genotype of the community partition problem, rather than applying the FA operators on a numerically encoded representation of its search space. It is also interesting to mention that, in that paper, the metric to optimize is the recently proposed Surprise [2], which assesses how statistically unlikely a given clustering arrangement is with respect to a random network featuring the same distribution of nodes per cluster. Another interesting work can be read in [79], in which a FA-based solver is shown to outperform other bio-inspired solvers such as GA and ACO applied to a small number of real-life networks. In this case, the metric to be maximized is the Girvan-Newman modularity which, in light of our bibliographic analysis, appears to be the *de facto* fitness choice. Artificial Bee Colony (ABC, [85]) is another modern bio-inspired solver also applied to community partition problems. In [68] the proposed solver automatically defines the optimal number of partitions of the network, thanks to the inherent multi-agent nature of the ABC solver. Other exemplifying works dealing with ABC for complex network partitioning can be found in [176, 129]. Finally, we acknowledge the exploration of CS-based heuristics for community partition presented in [203, 202], which rounds up the review of the background literature targeted in this section.

Finally, we pause at several recent works where assorted bio-inspired meta-heuristics have been adapted for the community detection problem. This is the case of [64], which explores the efficiency of a method never used before for community detection: the Fireworks Algorithm. The main characteristics that authors used to develop a competitive method are new initialization strategies and new mutation functions, both based on the label propagation strategy to speed up the convergence. Two years after that study, Messaoudi and Kamel retake the Fireworks Algorithm for community finding in Social Network context [119]. In that case, the solver is endowed with an Affinity Propagation approach for initialize the population, and a double-step mutation procedure. Another algorithmic scheme which has been recently adapted for same purposes is the Sine-Cosine meta-heuristic, developed and presented in [200]. Also significant is the contribution introduced in [30] in which the adequacy of the Chemical Reaction evolutionary method is explored. Main contribution of the algorithm implemented by authors in that study is the use of two encoding schemes for the same solver: the locus-based representation and the vector-based encoding. The main reason for this dual encoding approach is the enhancement of the exploration capability of the overall search algorithm. For testing the quality of the proposed method, the performance of the Chemical Reaction based approach is compared to that featured by standard community detection algorithms, such as Louvain or GA-Net.

Table 1: Recent literature showcasing the use of bio-inspired optimization for different community finding problem variants.

Community partition problem	Optimization criterion	Evolutionary Computation	Swarm Intelligence	Other bio-inspired solvers	Example of adopted criteria
Non-overlapping communities (partitional graph clustering)	Single-objective	[66], [66], [39], [128], [9], [127], [81], [105], [200], [125], [194], [74], [161], [102], [168], [78], [67]	[136], [72], [162], [37], [79], [68], [176], [129], [65], [61], [9], [119], [64], [32], [73], [24], [25]	[136], [9], [23], [57], [86]	$\max$ Modularity $\max$ Surprise $\max$ Extended Modularity
	Multi-objective	[182], [70], [34], [193], [51], [121], [30], [167], [142]	[146], [55], [100], [202], [82], [54], [58], [111], [126], [107]	[57]	$\max$ intra-link strength *versus* $\min$ inter-link strength
Overlapping communities (fuzzy graph clustering)	Single-objective	[157], [101], [27], [41]	[205], [4], [157], [77], [198]	[164], [89], [147], [159],[45]	$\max$ Leicht's Modularity [98] $\max$ Fuzzy Modularity [130]
	Multi-objective	[195], [178], [110], [191], [17], [174]	[104], [8]	[109]	$\max$ intra-link strength *versus* $\min$ inter-link strength
Time-evolving networks (dynamic community detection)	Single-objective	[141], [134], [33], [115]	[192], [164], [165], [15]	[206]	$\max$ Modularity $\max$ Conductance $\max$ Expansion $\max$ Internal Density
	Multi-objective	[12], [174], [154], [48], [7]	[120], [207], [83], [203]	[59], [50], [208], [204]	$\max$ Cluster Accuracy *versus* $\min$ Clustering Drift
Other problems (attributed, semantic, multiplex...)	Single-objective	[114], [144], [1]	[62], [28], [69], [122]	–	$\max$ Semantic Modularity (SimQ) $\max$ Eigenvector Centrality
	Multi-objective	[108], [123], [5], [6], [1]	–	[149], [56]	$\max$ Modularity *versus* Homogeneity

As evinced by the above references, community detection has been tackled by the community in many different ways, using a wide variety of solving approaches over different formulations of the underlying optimization problem. The recent literature is really huge and spans beyond the brief excerpt provided in this section. However, we note that the main propeller of this bustling research activity is the ever-growing number of metaphor-based solvers witnessed in the field of bio-inspired computation in recent times, such as Japanese Tree Frog Algorithm [61], Parliamentary Optimization Algorithm [4] or Penguins Search Optimization [65]. We will later revolve on the implications of this noted upsurge of literature in

Section 6. Finally, we summarize and complement in Table 1 the bio-inspired related literature reviewed in this section, which evidences the relevance of the field in the current scientific community.

3. Problem Statement

As has been introduced previously, we concentrate our analysis on the detection of communities in weighted directed networks. Diverse inference problems have been tackled in the literature for this class of networks, as they model complex real-world scenarios [22, 63]. The practical applicability of the findings derived from our investigations, and the relative scarcity of contributions dealing with this class of networks, are the reasons why we consider weighted directed graphs in this research work. In this line of reasoning, we note that other simpler graph types – assuming no weights and/or directedness – have been contemplated in the literature, grasping a notably higher interest than more complex network classes. We thereby conclude that the consideration of both weighted and directed edges in our problem formulation sheds light on a research niche that remains insufficiently studied to date.

Once the election of this kind of graphs has been justified, we start by mathematically modeling the weighted network as a graph $\mathcal{G} \doteq (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the group of $|\mathcal{V}| = V$ vertices or nodes of the network, $\mathcal{E}$ denotes to the set of connecting edges or links, and $f_{\mathcal{W}} : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}^{+}$ corresponds to a function assigning a non-negative weight to the edge connecting every pair of nodes. We also assume the absence of self loops in the graph ($f_{\mathcal{W}}(v, v) = 0$ $\forall v \in \mathcal{V}$), and that $f_{\mathcal{W}}(v, v') = 0$ if nodes v and v' are not connected. Furthermore, we define $f_{\mathcal{W}}(v, v') \equiv w_{v,v'}$, leading to a $V \times V$ adjacency matrix $\mathbf{W}$ given by $\mathbf{W} \equiv \{w_{v,v'} : v, v' \in \mathcal{V}\}$ and fulfilling that the trace $\text{Tr}(\mathbf{W}) = 0$ as per the lack of self loops assumed before. In addition, asymmetry is assumed in the graph edges, thus, $w_{v,v'}$ is not necessarily equal to $w_{v',v}$.

Considering this notation, the main problem of detecting communities in a graph $\mathcal{G}$ can be understood as the partition of the set of nodes $\mathcal{V}$ into a number of non-empty, disjoint groups, each with a non-fixed size. We refer as M to the number of groups or communities of partition $\widetilde{\mathcal{V}} \doteq \{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$, such that $\cup_{m=1}^{M} \mathcal{V}_m = \mathcal{V}$ and $\mathcal{V}_m \cap \mathcal{V}_{m'} = \emptyset \; \forall m' \neq m$ (i.e. no overlapping communities). Therefore, we can represent the community to which node v belongs as $\mathcal{V}^v \in \widetilde{\mathcal{V}}$. It is interesting to highlight here that the size of partitions is not restricted to any minimum or maximum fixed value.

The weighted directed nature of the links within the graph imposes a reformulation of the classical in-degree and out-degree values to yield the so-called input and output *strengths*, which are defined as [132]:

$$s_v^{in} = \sum_{v' \in \mathcal{V}} w_{v',v}, \qquad s_v^{out} = \sum_{v' \in \mathcal{V}} w_{v,v'}, \tag{1}$$

specifically, input and output strengths are the sum of the weights of the incident (outgoing) edges to (from) node v. These values represent both the weighted nature and the directivity of the $\mathbf{W}$ adjacency matrix. This is the reason because they play a crucial role in the measurement of the adequacy of communities, in a similar fashion to the in-degree and out-degree in unweighted and directed graphs.

Extending this logic further, a redefinition of the well-known *modularity* for undirected graphs introduced in [131, 98] can be used for measuring the quality of a specific partition $\widetilde{\mathcal{V}}$. To this end, we define a binary function $\delta : \mathcal{V} \times \mathcal{V}$ in $\{0, 1\}$ in which $\delta(v, v') = 1$ if $\mathcal{V}^v = \mathcal{V}^{v'}$ as per the partition $\widetilde{\mathcal{V}}$ (and 0 otherwise). With this introduced function the modularity for weighted directed graphs is given by:

$$Q(\widetilde{\mathcal{V}}) \doteq \frac{1}{|\sum_{\mathbf{W}}|} \sum_{v \in \mathcal{V}} \sum_{v' \in \mathcal{V}} \left[w_{v,v'} \frac{s_v^{in} s_{v'}^{out}}{|\sum_{\mathbf{W}}|} \right] \delta(v, v'), \tag{2}$$

where $|\sum_{\mathbf{W}}|$ denotes the sum weight of all edges of the network. Under this redefined partition quality indicator, the *best* partition $\widetilde{\mathcal{V}}^*$ of network $\mathcal{G}$ yields as:

$$\widetilde{\mathcal{V}}^* = \arg \max_{\widetilde{\mathcal{V}} \in \mathcal{B}_V} Q(\widetilde{\mathcal{V}}), \tag{3}$$

where $\mathcal{B}_V$ denotes the group of all possible partitions of $\mathcal{V}$ elements into nonempty subgroups (i.e. the solution space of the above combinatorial problem). The specific cardinality of this set is huge, which is given by the V-th Bell number [71]. This means that by following the recursion $B_{V+1} = \sum_{v=0}^{V} \binom{V}{v} B_v$ (with $V \geq 1$ and $B_0 = 1$) [180], if we consider a network composed by $V = 20$ nodes, it can be partitioned in approximately $517.24 \cdot 10^{12}$ different manners. Consequently, if we assume that a separated evaluation of the quality of a single partition can be computed within 1 microsecond on average, we would need more than one and a half years to check all possible combinations. This situation supports the need for using heuristic methods for the efficient exploration of this solution space, which lies at the motivational core of the literature surveyed previously and the developments presented in what follows.

4. Proposed Nature-inspired Solvers

With the aim of efficiently addressing the problem stated above, several modern nature-inspired meta-heuristics have been designed. In this regard, the bio-inspired optimization realm is composed by a plethora of different methods, each inspired by different biological phenomena. To narrow our experimentation and obtain meaningful conclusions, we have chosen an excerpt of bio-inspired algorithms contributed recently in the literature. The main reason behind the selection of the particular techniques considered in this work is the excellent performance that such algorithms have shown along the years, reflected in the momentum featured by these approaches within the scientific community [38]. Thus, prior to the description of each considered method, some common design directives are next portrayed, related specifically to the encoding strategy, solution repair mechanisms and the metrics employed for measuring the difference between two candidate partitions.

One of the most important design aspects when designing heuristics is the numerical representations of a solution to the problem at hand. In our case, we embrace the *label-based representation* [75] as the solution encoding strategy for partitions evolved during the search process. This way, each potential solution is encoded as a vector $\mathbf{x} = [c_1, c_2, \ldots, c_V]$ of V integers from the range $[1, \ldots, V]$, where we recall that $V = |\mathcal{V}|$ stands for the number of nodes in the whole graph. Additionally, c_v represents the cluster label to which node v belongs. For instance, and considering a network composed by 12 nodes, a possible feasible solution could be $\mathbf{x} = [1, 2, 2, 1, 1, 2, 2, 3, 2, 3, 3, 3]$, meaning that the partition underneath is $\widetilde{\mathcal{V}} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$, with $\mathcal{V}_1 = \{1, 4, 5\}$, $\mathcal{V}_2 = \{2, 3, 6, 7, 9\}$ and $\mathcal{V}_3 = \{8, 10, 11, 12\}$ as its compounding disjoint communities.

Once an encoding strategy has been selected, a metric of similarity between two different solutions (partitions) must be devised. This similarity is the functional basis of movement strategies inherent to each of the proposed techniques. In this research work we explore two possible functions in this regard, among other additional aspects. As a result, different configurations have been considered for the seven considered meta-heuristics that will be later detailed, leading to an experimentation benchmark composed by 19 different optimization algorithms. The first similarity function is the Hamming distance, which has been already used for other combinatorial problems. Specifically, the Hamming distance $D_H(\mathbf{x}, \mathbf{x}')$ is given by the number of non-corresponding elements (highlighted in bold) between two encoded individuals $\mathbf{x}'$ and $\mathbf{x}'$. For instance, if $\mathbf{x} = [1, 2, \mathbf{2}, 1, \mathbf{2}, 2, 2, 3, 2, 3, \mathbf{1}, \mathbf{1}]$ and $\mathbf{x} = [1, 2, \mathbf{1}, 1, \mathbf{1}, 2, 2, 3, 2, 3, \mathbf{3}, \mathbf{3}]$, then

$D_H(\mathbf{x}, \mathbf{x}') = 4$. The negligible time required to compute this similarity metric clashes with a severe drawback when it is used for this specific community detection problem: the need for repairing the individuals so as to avoid ambiguities in the genotype (numerical encoding) of two phenotypically equivalent partitions. A clarifying example occurs between $\mathbf{x} = [4, 2, 2, 4, 4, 2, 2, 3, 2, 3, 3, 3]$ and $\mathbf{x}' = [7, 1, 1, 7, 7, 1, 1, 4, 1, 4, 4, 4]$, which both represent the same network partition even though their Hamming distance is maximum ($D_H(\mathbf{x}, \mathbf{x}') = 12$). To overcome this issue, a repair procedure has been developed partly inspired from the one proposed in [44]. Thanks to this procedure, which is applied to every newly created solution, ambiguities generated by partitions are efficiently resolved, leading to both of the solutions exemplified above to an unified genotype, i.e. $\mathbf{x} = [1, 2, 2, 1, 1, 2, 2, 3, 2, 3, 3, 3]$. The second distance metric used is the Normalized Mutual Information (NMI), which has been used previously for similar goals [196, 189, 11]. The NMI score quantifies the level of agreement between two community partitions, yet ignoring label permutations [173]. As such, if $\text{NMI}(\mathbf{x}, \mathbf{x}') = 1$ the partitions $\widetilde{\mathcal{V}}$ and $\widetilde{\mathcal{V}}'$ represented are equal to each other. This also means that lower values denote that they are phenotypically different to each other.

Finally, the last aspect to mention before the description of the considered meta-heuristics is the design of the movement operators used for evolving individuals during the search process. In this sense, two groups of functions have been designed, which are separately employed in different configurations of the implemented solvers:

- The first set of operators corresponds to *blind movements*, which do not exploit any heuristic knowledge of the problem. In this category, four alternatives have been considered, which are applied depending on the distance between two individuals (in the case of FA, WCA, BA and PSO), or depending on the nature of the solution (in the case of ESA, CS and PVNS). Specifically, these functions are named CE_1, CE_3, CC_1 and CC_3. For each of these operators, the subscript represents the number of randomly selected vertices, which are extracted from its corresponding community. In CE_* operators, the taken nodes are re-inserted in already existing communities, while in CC_* nodes can be inserted also into newly generated communities.

- The second category of operators corresponds to *heuristic movements*, which leverage specific knowledge about the tackled problem to select the most appropriate movement at each iteration of the search process. In this category two subcategories can be discerned: *simple* heuristics (SH_*) and *improved* heuristics (IH_*), each one composed by two components: SH_1 and SH_3 for the SH_* category, and IH_1 and IH_3 for the IH_* category. It is important to mention that these operators are based strictly on the aforementioned measure of similarity (*distance*) between individuals, as their objective is to get an individual closer to other one (in the case of BA, for instance, to the best within the whole swarm). For this reason, these operators can only be utilized on those meta-heuristics conducting distance-based movements, namely, WCA, FA, BA and PSO. Specifically, SH_* methods select uniformly at random a node of the whole *destination* individual $\mathbf{x}'$, which denotes the solution the *in-movement* solution $\mathbf{x}$ is enforced to get closer to. Then, by analyzing the c_i' value associated to the node placed in the selected position, the *in movement* individual $\mathbf{x}$ adopts in its solution the whole $\mathcal{V}'_i$ community corresponding to the c_i' value. This procedure is repeated for as many $*$ times as indicated in the name of the applied operator (SH_* or IH_*). A visual example with the SH_1 operator may help the reader understand this movement process. If the following in-movement individual $\mathbf{x}$ moves towards the destination partition $\mathbf{x}'$ given by:

$$\mathbf{x} = [1,2,2,3,2,1,2,4,1,2,4,3], \quad \mathbf{x}' = [1,2,2,1,1,2,2,3,2,\mathbf{3},3,3], \tag{4}$$

and we further consider that c_{10}' has been randomly chosen, we can see that $c_{10} = 3$ for $\mathbf{x}'$. Thus, $\mathbf{x}$ would adopt the whole community identified by label 3, resulting in a *moved* solution $\mathbf{x}''$ given by:

$$\mathbf{x}'' = [1,2,2,3,2,1,2,\mathbf{3},1,\mathbf{3},\mathbf{3},\mathbf{3}]. \tag{5}$$

Regarding the IH_* operators, the movement process also departs from the uniformly random selection of a vertex from the *destination* solution $\mathbf{x}'$. Then, the community $\mathcal{V}'_i$ to which the selected node belongs is compared to all the communities of the in-movement individual $\mathbf{x}$. Finally, the node placed in the selected position adopts the c_i of the community that shares most similarity with $\mathcal{V}'_i$. This operation is also repeated for $*$ times. Another hypothesized example is next given to clarify this process. If we consider the following partitions:

$$\mathbf{x} = [3,4,4,5,4,3,3,4,5,6,6,6], \quad \mathbf{x}' = [1,2,1,\boldsymbol{3},1,2,2,\mathit{3},2,\mathit{3},\mathit{3},\mathit{3}], \tag{6}$$

and assuming that c_4' has been randomly selected, the community to compare is $\mathcal{V}'_3 = \{4,8,10,11,12\}$ since $c_4' = 3$. In this case, the cluster belonging to $\mathbf{x}$ that shares most similarities with $\mathcal{V}'_3$ is $\mathcal{V}_4 = \{10,11,12\}$. Thus, since the

c_i value of $\mathcal{V}_4$ is 6, the solution represented by $\mathbf{x}$ would use this value in c_4, resulting in the following moved individual $\mathbf{x}''$:

$$\mathbf{x} = [3, 4, 4, \mathbf{6}, 4, 3, 3, 4, 5, 6, 6, 6]. \quad (7)$$

Now, the details of the considered global search meta-heuristics are introduced, along with an explanation how the movement operators described above are embedded in their search procedures:

4.1. Water Cycle Algorithm (WCA)

The WCA solver was originally conceived in [43] for tackling continuous optimization problems. Similarly to recent works dealing with other applications [137], a discrete adaptation of this heuristic has been done to make WCA operators efficiently deal with the discrete solution encoding of this work. Laying aside the features detailed in the beginning of this section, the most crucial mechanism to design is the way in which streams and rivers flow to their corresponding leading river or sea. In this sense, and based on the original WCA, the movement of each stream $p_{str} \in \mathcal{P}_{str}$ (where we hereafter $\mathbf{x}_n^{(t)}$ denotes the n-th solution in the swarm at generation t) towards its river $\lambda(p_{str})$ at generation $t \in \{1, \ldots, T\}$ is set to:

$$\mathbf{x}_{p_{str}}^{(t+1)} = \Psi\left(\mathbf{x}_{p_{str}}^{(t)}, \min\left\{V, \left\lfloor rand \cdot \theta \cdot Dist\left(\mathbf{x}_{p_{str}}^{(t)}, \mathbf{x}_{\lambda(p_{str})}^{(t)}\right)\right\rfloor\right\}\right), \quad (8)$$

where $rand$ is a continuous random variable uniformly distributed in $\mathbb{R}[0, 1]$, and $Dist(\cdot, \cdot) \in \{D_H(\cdot, \cdot), \text{NMI}(\cdot, \cdot)\}$ represents the similarity function that can be parameterized depending on the solver. Additionally, θ is a heuristic parameter. Furthermore, $\Psi(\mathbf{x}, Z) \in \{CE_1, CE_3, CC_1, CC_3, SH_1, SH_3, IH_1, IH_3\}$, each depending on the number of times Z this function is applied to $\mathbf{x}_{p_{str}}^{(t)}$. The best position resulting from the Z movements performed on $\mathbf{x}_{p_{str}}^{(t)}$ is chosen as the output of the operator. The same procedure is followed for the movements of a river or a stream towards the sea, just replacing $\mathbf{x}_{\lambda(p_{str})}^{(t)}$ by $\mathbf{x}_{p_{sea}}(t)$. In is important to mention that the function is selected depending on the implemented variant of the WCA solver, as will be later detailed.

With the intention of boosting the exploration capacity of the technique, the *inclination* mechanism recently proposed in [137] is also used in the developed WCA solvers for community detection. This simple but efficient mechanism provides the search methods with the intelligence for properly choosing the movement operator to use at each iteration for each individual. This decision depends on the situation of the raindrop which is about to move. Specifically, each time an individual is ready to conduct a movement in the solution space, the aforementioned *inclination* $\xi(\mathbf{x}, \mathbf{x}')$ is calculated, using as the reference the distance $Dist(\cdot, \cdot)$ between the raindrop $\mathbf{x}$ and its designated river/sea $\mathbf{x}'$. Specifically, $\xi(\cdot, \cdot)$ could be equal to $V/D_H(\cdot, \cdot)$ or $V/NMI(\cdot, \cdot)$. Thus, the larger the value of $Dist(\cdot, \cdot)$ is, the higher $\xi(\cdot, \cdot)$ should be, forcing the technique to conduct a *fast move* with a higher probability. On the other hand, if $Dist(\cdot, \cdot)$ has a small value, the inclination is made lower, suggesting that the search is near a promising area of the solution space, thereby requiring to perform a *slow move* with higher probability. In this paper, among all the considered operators, CC_*, SH_3 and IH_3 are classified as *fast moves*, whereas CE_*, SH_1 and IH_1 are regarded as *slow moves*. Last but not least, raining and evaporation procedures follow the same philosophy as in the basic WCA. In particular, the raining process performs a number R of consecutive CC_3 movements regardless the variant of the WCA solver in use.

4.2. Bat Algorithm (BA)

Similarly to what happened for WCA and in general, as it occurs in most Swarm Intelligence methods, the canonical BA was first introduced for solving continuous-variable optimization problems. For this reason, a discrete adaptation has been designed also for this second meta-heuristic. As in most of the adaptations [139], each bat in the swarm represents a feasible solution of the problem. Additionally, loudness A_i and pulse emission r_i have been considered in the same form as in the classical version of the BA. Moreover, in order to simplifying the complexity of the method, the frequency parameter f_i has not been considered. Lastly, velocity v_i has been adapted by considering $D_H(\cdot, \cdot)$ or $NMI(\cdot, \cdot)$ for measuring the similarity between two different bats. Thus, $v_i^{(t)}$ is computed as:

$$v_i^{(t)} = \text{Uniform}[1, Dist(\mathbf{x_i^{(t)}}, \mathbf{x_{best}^{(t)}})], \tag{9}$$

i.e., the value of $v_i^{(t)}$ of bat (solution) i at time step t is drawn from a uniform discrete probability distribution between 1 and the difference between the index i of this bat and that of the fittest bat in the swarm. Furthermore, the movement of a bat follows the same rule as defined in the above Eq. (8), using $v_i^{(t)}$ as the number of movements to be applied to solution $\mathbf{x}_i^{(t)}$. Finally, an inclination mechanism is also developed for the discrete versions of the BA presented in this research, which relies on the same procedure as for its WCA counterpart, using the best bat in the swarm as the reference.

4.3. Firefly Algorithm (FA)

Again, some modifications have been performed over the original version of the FA for properly dealing with the community detection problem addressed in this research. As in the BA, each firefly in the population represents a feasible solution for the problem. Furthermore, light absorption, a concept essential for adjusting fireflies' attractiveness, is considered in this discrete adaptation. The distance between two fireflies is also computed using $D_H(\cdot,\cdot)$ or $NMI(\cdot,\cdot)$, giving rise to different flavors of FA-based solvers for detecting communities. Moreover, the movement of a firefly to another brighter one is defined by the same logic shown in Expression (8). Thus, each time a firefly $\mathbf{x}$ is about to carry out a move towards another firefly. $\mathbf{x}'$, it examines $Dist(\mathbf{x},\mathbf{x}')$. If this value is greater than $V/2$, a *large* movement is performed by using CC_*, SH_3 or IH_3. Otherwise, a *short* move is made by means of the CE_*, SH_1 or IH_1 operators. This mechanism can be regarded as an adaption to embed a behavior similar to the *inclination* concept described for WCA.

4.4. Particle Swarm Optimization (PSO)

Similarly to previous solvers in this section, PSO has been already applied to discrete problems in the past [31, 201]. We rely on this previous background as a inspiration for our case study. As such, each particle represents a possible solution for the addressed problem, whereas the computation of the velocity parameter $v_i^{(t)}$, movement functions and inclination feature of WCA and BA are performed as for the previously detailed WCA and BA solvers. The movement criterion shown in Expression (8) is also implemented to drive the movement of particles inside the swarm. Likewise, $D_H(\cdot,\cdot)$ or $NMI(\cdot,\cdot)$ have been adopted as similarity functions to compare among different particles.

4.5. Cuckoo Search (CS)

CS was conceived in [186] as a structured randomized search method inspired by the combination of the holoparasite characteristics and Lévy flight foraging configurations of some cuckoo species. By virtue of its reduced number of control parameters and its relative efficiency when tackling complex optimization problems, adaptations of CS to discrete problem formulations have been particularly notable during the last year [106, 117, 19]. In our case we opt for an similar adaptation strategy to the one reported in [140] for the Traveling Salesman Problem, defining similar parameters and mechanisms. For the cuckoos movement, the same CC_*, CE_*, IH_* and SH_* functions have been considered. Depending on the CS version implemented, a random function is first assigned to each cuckoo at the beginning of the algorithm execution. Additionally, the movement of each individual is carried out by using the logic in Expression (8), using $D_H(\cdot,\cdot)$ or $NMI(\cdot,\cdot)$ as the distance function, and the best cuckoo as the reference solution.

4.6. Evolutionary Simulated Annealing (ESA)

The fifth developed global search heuristic is a population-based evolutionary version of Simulated Annealing [190]. The population-based variant of this single-point heuristic has been adopted for the sake of fairness in the comparison with the rest of methods in the benchmark. In line with this, the ESA-based schemes are endowed with the previously described movement operators. This way, each element of the population has its own randomly assigned function. Moreover, each individual has a temperature value randomly drawn from $\mathbb{R}[0.7, 1.0]$. We again rely on Expression (8) for the movement of solutions. Again, $D_H(\cdot,\cdot)$ has been used as the function to measure the distance between individuals in the population. Thereby, each solution performs a number of successive movements as per $D_H(\cdot,\cdot)$, choosing the best individual in the population as its reference. Among all movements, the most profitable one as dictated by their fitness improvement is selected. The best individual, however, performs a random number of movements between 1 and Z, which is declared to be an additional parameter to be tuned for the ESA-based heuristics.

4.7. Population-based Variable Neighborhood Search (PVNS)

The last considered meta-heuristic in this study consists of a population-based variant of the original VNS. Taking as a baseline the same design principles considered for ESA, each individual of the population is assigned a movement function randomly selected among all the available options: CE_1, CE_3, CC_1 and CC_3 for the first cases; SH_1 and SH_3 for the second group; and IH_1 and IH_3 for the last one. Then, at each iteration, each individual performs a movement based on its assigned operator, which can be replaced by a different one with probability 0.25 to promote diversity in the movement dynamics within the population.

5. Experimentation and Results

In order to evaluate the performance of the developed methods, a comprehensive experimental setup has been designed over a heterogeneous set of synthetically created network instances. Although repositories containing emulated and real network instances can be found available in the public domain, our rationale for using synthetic networks is to have full control of the true community structure underneath the network, so that supervised quality indicator statistics can be computed and used for a fair comparison. We next detail how such instances have been generated to provide intuition on our criterion to build this benchmark. Nevertheless, the set of generated instances to foster discussion has been made available to the public domain in [84] in order to stimulate future algorithmic developments.

The conducted experimentation is organized in three different phases: the first one is done over small network instances of up to $V = 75$ nodes, whereas the second and third ones deal with network instances of higher size (between $V = 100$ and $V = 600$ nodes):

- In the first experimentation phase (Subsection 5.1), the benchmark comprises 24 network composed by $V \in \{35, 50, 75\}$ elements. For each graph, a different amount of *ground of truth* communities is enforced by generating a partition of the network (with random sizes for its constituent groups $\{\mathcal{V}_m\}_{m=1}^{M}$), and then by connecting nodes belonging to different communities with probability p_{out}, and nodes within every group with probability p_{in}. The *ground of truth* community partition can be thought to be less detectable by any community detection scheme if the value of p_{out} gets close to that of p_{in}. In addition, weights $w_{v,v'}$ for each edge (v, v') have been drawn uniformly at random from $\mathbb{R}[10.0, 20.0]$ (intra-community edges) and $\mathbb{R}[0.0, 10.0]$ (inter-community edges). This network building procedure allows evaluating the performance of all implemented methods over *noisy* versions of a graph characterized by a controlled underlying community distribution. We advocate for this benchmark criterion as opposed to the common practice in the field, by which comparisons of this kind are based on the attained fitness value of each solver rather than on their capacity to infer the real community structure of the network.

- Five additional sets of network instances have been generated for the second phase of the experimentation (Subsection 5.2) by enlarging the number of nodes to $V \in \{100, 200, 300, 400, 500\}$. These network instances have been constructed with the main intention of assessing the performance of designed solvers when the dimensions of the network increase. For constructing these instance sets, the LFR algorithm described in [93] for producing directed weighted networks with overlapping communities has been used. Specifically, for all network instances we set $k = 15$ (*average degree*) and $maxk = 40$ (*maximum*). Additionally, for instances with $V \in \{100, 200\}$ the values of muw (*weight mixing parameter*), $minc$ and $maxc$ (*minimum* and *maximum community size*) are fixed to 0.1, 5 and 20, respectively. For the case with $V = 300$, on the other hand, the values of such parameters are set to $muw = 0.2$, $minc = 10$ and $maxc = 25$. For the remaining two instances we configure the instance generation algorithm in [93] with parameters $muw = 0.2$, $minc = 15$ and $maxc = 30$. The concatenation of the values for all these LFR parameters compose the label of each network referred through the presentation and discussion of the results.. Due to the high computational effort needed to solve these larger network instances, only the four most promising methods found in the first experimentation phase have been considered for this second stage. The criterion to discriminate which four methods perform best is later detailed.

- The third experimentation stage (Subsection 5.3) comprises network instances of large size and higher connectivity between clusters as per the topological mixing coefficient of the LFR benchmark generator. The rationale and coverage of this third set of experiments are later elaborated in the corresponding subsection.

As has been pointed, 19 different solvers have been implemented, which result from the allowed combinations between the seven considered global search meta-heuristic, the two similarity measures between community partitions, and the eight implemented movement operators. Table 2 summarizes the main characteristics of each of these 19 solvers. Every meta-heuristic scheme shares the same parametric configuration, meaning that e.g. all BA approaches are configured by using the same values for their control parameters, disregarding the similarity function or operators employed.

Aiming to reach statistically reliable insights on the obtained results, for each solver 15 independent runs have been executed for every network instance considered in the first experimental stage, and 10 for each larger instance in the second phase. The search process initiated at each run is ended when $V + \sum_{v=1}^{V} v = V(V+3)/2$ iterations of the algorithm at hand have been executed without any improvement of the best solution found. The population size has been established to 50 individuals for each method. In the case of WCA, the number of rivers has been set to 9 (approximately 20% of the whole population), yielding a total of 40 streams. Moreover, the maximum distance for evaporation and R have been fixed to 5% and an uniform random value from the discrete range $\mathbb{N}[0, \lfloor 0.5V \rfloor]$, respectively. In FA-based methods the value of the light absorption coefficient is configured as $\gamma = 0.95$, whereas for solvers with BA as their core meta-heuristic $\alpha = \beta = 0.98$, $A_i^0 = 1.0$ (*loudness*) and $r_i^0 = 0.1$ (*rate*). Besides that, for CS the *probability to discover an alien egg* is set to $p_a = 0.2$. These parametric values have been tuned by following the guidelines given in [137, 139, 138, 140]. Finally, ESA and PVNS have been configured as explained in Subsections 4.6 and 4.7, respectively.

Table 2: Characteristics of each implemented solver. $D_H(\cdot,\cdot)$ refers to Hamming Distance function, $NMI(\cdot,\cdot)$ refers to Normalized Mutual Information function.

Identifying label	Meta-heuristic	Similarity measure	Movement function(s)
WCA_Ham	WCA	$D_H(\cdot,\cdot)$	CC_*, CE_*
ESA_Ham	ESA	$D_H(\cdot,\cdot)$	CC_*, CE_*
FA_Ham	FA	$D_H(\cdot,\cdot)$	CC_*, CE_*
PVNS_Ham	PVNS	$D_H(\cdot,\cdot)$	CC_*, CE_*
CS_Ham	CS	$D_H(\cdot,\cdot)$	CC_*, CE_*
PSO_Ham	PSO	$D_H(\cdot,\cdot)$	CC_*, CE_*
BA_Ham	BA	$D_H(\cdot,\cdot)$	CC_*, CE_*
WCA_NMI_B	WCA	$NMI(\cdot,\cdot)$	CC_*, CE_*
FA_NMI_B	FA	$NMI(\cdot,\cdot)$	CC_*, CE_*
PSO_NMI_B	PSO	$NMI(\cdot,\cdot)$	CC_*, CE_*
BA_NMI_B	BA	$NMI(\cdot,\cdot)$	CC_*, CE_*
WCA_NMI_SH	WCA	$NMI(\cdot,\cdot)$	SH_*
FA_NMI_SH	FA	$NMI(\cdot,\cdot)$	SH_*
PSO_NMI_SH	PSO	$NMI(\cdot,\cdot)$	SH_*
BA_NMI_SH	BA	$NMI(\cdot,\cdot)$	SH_*
WCA_NMI_IH	WCA	$NMI(\cdot,\cdot)$	IH_*
FA_NMI_IH	FA	$NMI(\cdot,\cdot)$	IH_*
PSO_NMI_IH	PSO	$NMI(\cdot,\cdot)$	IH_*
BA_NMI_IH	BA	$NMI(\cdot,\cdot)$	IH_*

5.1. First Experimentation: Comparing the proposed Bio-inspired Methods

For the sake of a better readability, outcomes have been divided in four different tables. Thus, in Tables 4-7, statistics (average/best/standard deviation) of the results attained by each of the 19 solvers for the first experimentation are shown in terms of the NMI with respect to the *ground of truth* partition of every network instance. A first inspection over these values allows to glimpse a promising performance of methods such as WCA_Ham, PSO_NMI_B, BA_NMI_SH and BA_NMI_IH, which show superior average NMI scores for most cases. In fact, these methods are the best ones in their specific category. In overall, the best solver of the 19 considered ones is BA_NMI_SH. Furthermore, as could have been intuitively expected beforehand, outcomes degrade when values of p_{in} and p_{out} are made sufficiently close to each other to etch topological noise on the *ground of truth* partition. This aspect can be analyzed in instances such as $(V, M, p_{in}, p_{out}) = (50, 5, 0.6, 0.4)$ (for which the best partition found attains NMI $= 0.699$) and $(50, 5, 0.9, 0.1)$ (which is solved by 8 methods in all their runs).

Turning the discussion onto the computational efficiency of these techniques, average run times needed by each method are shown in Tables 8 and 9. At this point we note that different studies have pointed out that run times are strongly dependent on the computer resources, programming language and other practical aspects of the experimentation. Therefore, we have also conducted an analysis of the convergence behavior of the algorithms, which decouples our complexity insights from the aforementioned implementation dependence. This being said, interesting trends arise from Tables 8 and Table 9. Focusing on the methods that obtained better performance in terms of quality, namely `WCA_Ham`, `PSO_NMI_B`, `BA_NMI_SH` and `BA_NMI_IH`, it can be observed that `WCA_Ham` requires more time that the other alternatives. This conforms to expectations, because the meta-heuristic search strategy of WCA is more complex than that of BA and PSO. On the other hand, negligible differences can be found between BA and PSO, obtaining similar performance levels in this regard. Thus, execution times are not conclusive as a choice criterion for discriminating the best developed technique.

Table 3: Unadjusted and adjusted p-values obtained as a result of the application of Holm's post-hoc procedure using `BA_NMI_SH` as control algorithm. A p value lower than 0.05 means significant differences.

Index	Algorithm	Unadjusted p	p_{Holm}
1	FA_Ham	0	0
2	ESA_Ham	0.000008	0.000129
3	FA_NMI_B	0.000043	0.000687
4	FA_NMI_SH	0.000113	0.001699
5	FA_NMI_IH	0.001831	0.02563
6	WCA_NMI_SH	0.002271	0.029524
7	WCA_NMI_IH	0.002473	0.029675
8	WCA_NMI_B	0.005603	0.061635
9	PVSN_Ham	0.133487	1.334867
10	CS_Ham	0.140255	1.334867
11	BA_Ham	0.20421	1.63368
12	PSO_NMI_SH	0.23805	1.666349
13	BA_NMI_B	0.24319	1.666349
14	PSO_NMI_IH	0.248408	1.666349
15	PSO_NMI_B	0.270059	1.666349
16	PSO_Ham	0.644303	1.93291
17	BA_NMI_IH	0.847452	1.93291
18	WCA_Ham	0.857509	1.93291

Going back to the results analysis, and following the guidelines in [40, 135], two different tests have been carried out to resolve the statistical relevance of the reported performance gaps. To begin with, the Friedman's non-parametric test for multiple comparison allows proving whether differences among the results obtained by all reported methods can be declared as statistically significant. The last row of Tables 4-7 displays the mean ranking returned by this non-parametric test for each of the compared algorithms (the lower the rank, the better the performance). These results support the conclusions drawn above: `BA_NMI_SH` is the best performing method. Moreover, in their own category, `WCA_Ham`, `PSO_NMI_B`, and `BA_NMI_IH` emerge as the most promising alternatives. Besides that, the Friedman statistic obtained is 116.06, distributed according to χ^2 with 19 degrees of freedom. Furthermore, and establishing the confidence interval in 99%, being 36.191 the critical point in the χ^2 distribution with 19 degrees of freedom. Since $116.06 > 36.191$, it can be concluded that there are significant differences among the results.

Table 4: (Part 1) Obtained NMI results (average/best/standard deviation) using WCA, ESA, FA, PVNS, CS, PSO and BA. The _`Ham` suffix means that the solver uses the Hamming distance as the similarity measurement function. Best average results have been highlighted in bold.

(V, M, p_{in}, p_{out})	`WCA_Ham`	`ESA_Ham`	`FA_Ham`	`PVNS`	`CS`	`PSO_Ham`	`BA_Ham`
	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std
$(35, 4, 0.6, 0.1)$	0.526/0.526/0.000	0.515/0.526/0.010	0.521/0.547/0.010	0.526/0.526/0.000	0.522/0.526/0.010	0.526/0.526/0.000	0.525/0.526/0.003
$(35, 4, 0.9, 0.4)$	**0.876**/0.876/0.000	0.860/0.876/0.010	0.745/0.768/0.010	**0.876**/0.876/0.000	0.864/0.876/0.013	**0.876**/0.876/0.000	0.851/0.876/0.018
$(35, 7, 0.6, 0.1)$	**1.000**/1.000/0.000	0.972/1.000/0.010	0.900/0.929/0.010	**1.000**/1.000/0.000	0.973/1.000/0.021	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 7, 0.6, 0.4)$	0.807/0.807/0.000	0.827/0.863/0.010	0.800/0.828/0.010	0.806/0.807/0.010	0.828/0.855/0.017	0.806/0.807/0.003	**0.829**/0.855/0.020
$(35, 7, 0.8, 0.1)$	**1.000**/1.000/0.000	0.997/1.000/0.010	0.927/0.949/0.010	**1.000**/1.000/0.000	0.997/1.000/0.009	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 7, 0.9, 0.4)$	**1.000**/1.000/0.000	0.997/1.000/0.010	0.914/0.935/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 18, 0.6, 0.1)$	0.960/0.969/0.010	0.931/0.962/0.010	0.952/0.973/0.010	0.954/0.969/0.010	0.957/0.959/0.003	0.950/0.969/0.013	0.959/0.969/0.009
$(35, 18, 0.9, 0.4)$	0.998/1.000/0.010	0.971/0.974/0.010	0.974/0.990/0.010	0.998/1.000/0.010	0.990/0.992/1.000	0.992/1.000/0.009	0.989/1.000/0.010
$(50, 5, 0.6, 0.1)$	**1.000**/1.000/0.000	0.998/1.000/0.010	0.821/0.851/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 5, 0.6, 0.4)$	0.664/0.699/0.010	0.680/0.699/0.010	0.684/0.658/0.010	0.689/0.699/0.010	0.680/0.699/0.014	0.691/0.699/0.006	0.677/0.690/0.019
$(50, 5, 0.9, 0.1)$	**1.000**/1.000/0.000	0.996/1.000/0.010	0.825/0.905/0.030	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 10, 0.7, 0.4)$	0.971/0.972/0.000	0.971/1.000/0.010	0.893/0.908/0.010	0.972/0.972/0.010	0.970/0.977/0.006	0.972/0.972/0.000	0.970/0.972/0.004
$(50, 10, 0.9, 0.4)$	**1.000**/1.000/0.000	0.989/1.000/0.010	0.941/0.962/0.010	**1.000**/1.000/0.000	0.989/0.989/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 25, 0.6, 0.1)$	0.979/0.989/0.010	0.952/0.965/0.010	0.955/0.969/0.010	0.967/0.977/0.010	0.979/0.982/0.004	0.971/0.989/0.008	0.972/0.988/0.008
$(50, 25, 0.6, 0.4)$	0.955/0.968/0.010	0.942/0.961/0.010	0.944/0.961/0.010	0.947/0.968/0.010	0.955/0.960/0.009	0.950/0.957/0.005	0.949/0.966/0.009
$(50, 25, 0.9, 0.4)$	0.990/0.991/0.010	0.971/0.987/0.010	0.970/0.980/0.010	0.982/0.991/0.010	0.986/0.986/0.000	0.983/0.991/0.004	0.984/0.991/0.006
$(75, 8, 0.6, 0.1)$	0.987/1.000/0.010	0.959/1.000/0.010	0.828/0.844/0.010	0.971/1.000/0.010	0.939/0.991/0.025	0.982/1.000/0.014	0.935/1.000/0.022
$(75, 8, 0.8, 0.3)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	0.865/0.993/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(75, 8, 0.9, 0.4)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	0.896/0.998/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(75, 15, 0.6, 0.2)$	0.986/0.987/0.010	0.982/0.989/0.010	0.892/0.917/0.010	0.984/0.989/0.010	0.974/0.992/0.15	0.986/0.987/0.001	0.979/0.987/0.010
$(75, 30, 0.6, 0.1)$	**0.971**/0.976/0.010	0.949/0.973/0.010	0.943/0.956/0.010	0.956/0.966/0.010	0.968/0.970/0.006	0.966/0.976/0.006	0.965/0.974/0.006
$(75, 30, 0.8, 0.4)$	**0.966**/0.970/0.010	0.951/0.971/0.010	0.939/0.955/0.010	0.958/0.979/0.010	0.962/0.968/0.004	0.963/0.979/0.008	0.962/0.975/0.006
$(75, 38, 0.9, 0.1)$	**0.984**/0.993/0.010	0.972/0.981/0.010	0.972/0.979/0.010	0.973/0.981/0.010	0.981/0.983/0.002	0.982/0.987/0.003	0.980/0.993/0.007
$(75, 38, 0.9, 0.4)$	**0.985**/0.993/0.010	0.968/0.981/0.010	0.970/0.982/0.010	0.973/0.994/0.010	0.982/0.984/0.002	0.976/0.982/0.005	0.977/0.982/0.003
Friedman's non-parametric test (mean ranking)							
Rank	6.7917	13.7708	17.6458	8.9375	8.8958	7.25	8.5625

The second statistical test is the Holm's post-hoc test. For properly conducting this test, BA has been set as the control algorithm. Table 3 gathers the unadjusted and adjusted p-values obtained through the application of Holm's post-hoc procedure. From these p-values it can be stated that BA, for the first case, and FA, for the second one, are significantly better than their counterparts at a 95% confidence level, since all p values are lower than 0.05. From this statistical analysis, several interesting conclusions can be drawn. To begin with, `BA_NMI_IH` is the

Table 5: (Part 2) Obtained NMI results (average/best/standard deviation) using WCA, FA, PSO and BA. The `_NMI_B` suffix indicates the use of NMI as distance function and blind movements (CC_* and CE_*). Best average results have been highlighted in bold.

(V, M, p_{in}, p_{out})	`WCA_NMI_B`	`FA_NMI_B`	`PSO_NMI_B`	`BA_NMI_B`
	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std
$(35, 4, 0.6, 0.1)$	0.523/0.526/0.005	**0.530**/0.542/0.010	0.526/0.526/0.000	0.526/0.526/0.000
$(35, 4, 0.9, 0.4)$	0.872/0.876/0.009	0.814/0.876/0.054	**0.876**/0.876/0.000	**0.876**/0.876/0.000
$(35, 7, 0.6, 0.1)$	1.000/1.000/0.000	0.921/1.000/0.000	1.000/1.000/0.000	1.000/1.000/0.000
$(35, 7, 0.6, 0.4)$	0.809/0.825/0.004	0.825/0.866/0.023	0.807/0.807/0.000	0.812/0.838/0.000
$(35, 7, 0.8, 0.1)$	**1.000**/1.000/0.000	0.964/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 7, 0.9, 0.4)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 18, 0.6, 0.1)$	0.948/0.959/0.012	0.949/0.954/0.004	0.952/0.959/0.006	0.954/0.959/0.005
$(35, 18, 0.9, 0.4)$	0.987/0.992/0.007	0.992/0.992/0.000	0.992/0.992/0.000	0.992/0.992/0.000
$(50, 5, 0.6, 0.1)$	**1.000**/1.000/0.000	0.869/1.000/0.109	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 5, 0.6, 0.4)$	0.675/0.699/0.022	0.666/0.707/0.038	0.694/0.699/0.006	0.686/0.699/0.015
$(50, 5, 0.9, 0.1)$	0.982/1.000/0.028	0.941/0.966/0.064	0.966/0.966/0.000	0.975/1.000/0.014
$(50, 10, 0.7, 0.4)$	0.972/0.972/0.000	0.967/0.972/0.006	0.972/0.972/0.000	0.973/1.000/0.014
$(50, 10, 0.9, 0.4)$	0.989/0.989/0.000	0.979/0.989/0.037	0.989/0.989/0.000	0.989/0.989/0.000
$(50, 25, 0.6, 0.1)$	0.962/0.984/0.013	0.979/0.983/0.004	0.966/0.984/0.007	0.973/0.983/0.010
$(50, 25, 0.6, 0.4)$	0.949/0.960/0.010	0.952/0.960/0.005	0.954/0.970/0.006	**0.956**/0.961/0.007
$(50, 25, 0.9, 0.4)$	0.974/0.983/0.005	0.986/0.986/0.000	0.983/0.993/0.005	0.986/0.986/0.001
$(75, 8, 0.6, 0.1)$	0.967/0.991/0.022	0.893/0.991/0.105	0.991/0.991/0.000	0.942/0.991/0.027
$(75, 8, 0.8, 0.3)$	0.996/1.000/0.010	0.955/0.980/0.062	0.981/1.000/0.004	**1.000**/1.000/0.009
$(75, 8, 0.9, 0.4)$	**1.000**/1.000/0.000	0.924/1.000/0.093	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(75, 15, 0.6, 0.2)$	0.988/0.992/0.004	**0.992**/0.992/0.000	**0.992**/0.992/0.000	0.977/0.992/0.018
$(75, 30, 0.6, 0.1)$	0.967/0.979/0.008	0.943/0.956/0.006	0.960/0.978/0.009	0.953/0.970/0.157
$(75, 30, 0.8, 0.4)$	0.959/0.975/0.007	0.940/0.955/0.007	0.962/0.976/0.007	0.961/0.977/0.008
$(75, 38, 0.9, 0.1)$	0.970/0.987/0.006	0.972/0.979/0.003	0.977/0.987/0.005	0.976/0.989/0.008
$(75, 38, 0.9, 0.4)$	0.972/0.979/0.006	0.969/0.982/0.008	0.978/0.984/0.004	0.977/0.991/0.010
Friedman's non-parametric test (mean ranking)				
Rank	11	13.1458	8.2917	8.3958

most promising method in terms of results quality, yet the difference between its counterparts is not significant in many cases. Additionally, this table clarifies that, in general, best performing meta-heuristic schemes are PSO and BA, exhibiting a superior overall performance in all its versions. On the other hand, FA, ESA and WCA (except `WCA_Ham`, which obtains promising outcomes) have demonstrated not to be appropriate to tackle the problem tackled in this part of the paper. Finally, PVSN and CS schemes are in medium positions of the ranking, failing to perform competitively in any of the instances of the benchmark.

Table 6: (Part 3) Obtained NMI results (average/best/standard deviation) using WCA, FA, PSO and BA. The `_NMI_SH` suffix means the use of NMI as distance function and SH_* movement functions. Best average results have been highlighted in bold.

(V, M, p_{in}, p_{out})	`WCA_NMI_SH`	`FA_NMI_SH`	`PSO_NMI_SH`	`BA_NMI_SH`
	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std
$(35, 4, 0.6, 0.1)$	0.520/0.534/0.009	0.528/0.540/0.011	0.517/0.530/0.024	0.514/0.534/0.023
$(35, 4, 0.9, 0.4)$	**0.876**/0.876/0.000	**0.876**/0.876/0.000	**0.876**/0.876/0.000	**0.876**/0.876/0.000
$(35, 7, 0.6, 0.1)$	0.985/1.000/0.020	0.925/0.941/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 7, 0.6, 0.4)$	0.810/0.863/0.015	0.794/0.835/0.024	0.807/0.807/0.000	0.810/0.835/0.007
$(35, 7, 0.8, 0.1)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 7, 0.9, 0.4)$	**1.000**/1.000/0.000	0.935/0.968/0.019	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 18, 0.6, 0.1)$	0.945/0.969/0.013	0.898/0.919/0.013	0.954/0.969/0.009	**0.964**/0.969/0.003
$(35, 18, 0.9, 0.4)$	0.983/1.000/0.011	0.904/0.933/0.011	0.989/1.000/0.008	**1.000**/1.000/0.000
$(50, 5, 0.6, 0.1)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 5, 0.6, 0.4)$	0.689/0.698/0.007	0.683/0.699/0.032	**0.695**/0.699/0.005	**0.695**/0.699/0.012
$(50, 5, 0.9, 0.1)$	0.992/1.000/0.019	**1.000**/1.000/0.000	0.996/1.000/0.014	**1.000**/1.000/0.000
$(50, 10, 0.7, 0.4)$	0.972/0.972/0.000	0.972/0.972/0.000	0.972/0.972/0.000	**0.975**/1.000/0.006
$(50, 10, 0.9, 0.4)$	0.998/1.000/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 25, 0.6, 0.1)$	0.964/0.988/0.010	0.870/0.888/0.008	0.971/0.980/0.006	**0.983**/0.989/0.008
$(50, 25, 0.6, 0.4)$	0.945/0.964/0.009	0.858/0.878/0.015	0.951/0.964/0.007	0.955/0.964/0.002
$(50, 25, 0.9, 0.4)$	0.976/0.991/0.008	0.987/0.987/0.007	0.980/0.991/0.004	**0.991**/0.991/0.000
$(75, 8, 0.6, 0.1)$	0.976/0.991/0.008	0.983/1.000/0.000	0.980/0.991/0.004	0.989/0.991/0.004
$(75, 8, 0.8, 0.3)$	0.995/1.000/0.009	0.994/1.000/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.018
$(75, 8, 0.9, 0.4)$	**1.000**/1.000/0.000	0.925/1.000/0.098	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(75, 15, 0.6, 0.2)$	0.985/0.987/0.001	0.980/0.987/0.009	0.986/0.987/0.001	0.973/0.987/0.018
$(75, 30, 0.6, 0.1)$	0.959/0.974/0.008	0.954/0.967/0.008	0.962/0.967/0.003	0.956/0.976/0.003
$(75, 30, 0.8, 0.4)$	0.959/0.973/0.007	0.953/0.965/0.006	0.962/0.972/0.006	0.962/0.966/0.006
$(75, 38, 0.9, 0.1)$	0.975/0.986/0.006	0.893/0.905/0.006	0.976/0.985/0.005	0.976/0.987/0.010
$(75, 38, 0.9, 0.4)$	0.972/0.985/0.007	0.882/0.895/0.008	0.974/0.981/0.004	0.977/0.988/0.009
Friedman's non-parametric test (mean ranking)				
Rank	11.4583	12.7708	8.4167	6.5

5.2. *Second Experimentation: Scalability and Performance in Larger Networks*

Once discussed the first stage of the experimentation, we proceed by commenting on the results of a second phase by employing instances of larger size. The main objective with these tests is to go deeper in the analysis of the most promising solvers, trying to conclude which one scales best when partitioning networks. For this purpose, the best performing methods of each category have been considered: `WCA_Ham`, `PSO_NMI_B`, `BA_NMI_SH` and `BA_NMI_IH`. Similarly to the previous phase, Table 10 summarizes the average, best and standard deviation of the NMI scores achieved by every solver over each instance. This experimentation also analyzes the convergence behavior of each method. For this

Table 7: (Part 4) Obtained NMI results (average/best/standard deviation) using WCA, FA, PSO and BA. The _NMI_IH suffix denotes the use of NMI as distance function and IH_* operators. Best average results have been highlighted in bold.

(V, M, p_{in}, p_{out})	WCA_NMI_IH	FA_NMI_IH	PSO_NMI_IH	BA_NMI_IH
	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std	Avg/Best/Std
$(35, 4, 0.6, 0.1)$	0.495/0.534/0.044	0.528/0.540/0.011	0.523/0.534/0.009	0.521/0.534/0.008
$(35, 4, 0.9, 0.4)$	0.872/0.876/0.009	0.844/0.886/0.021	**0.876**/0.876/0.000	**0.876**/0.876/0.000
$(35, 7, 0.6, 0.1)$	0.997/1.000/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 7, 0.6, 0.4)$	0.811/0.845/0.011	0.820/0.863/0.018	0.808/0.825/0.005	0.819/0.863/0.015
$(35, 7, 0.8, 0.1)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 7, 0.9, 0.4)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(35, 18, 0.6, 0.1)$	0.944/0.969/0.010	0.950/0.965/0.017	0.959/0.969/0.010	**0.964**/0.969/0.006
$(35, 18, 0.9, 0.4)$	0.981/1.000/0.007	0.998/1.000/0.010	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 5, 0.6, 0.1)$	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 5, 0.6, 0.4)$	0.686/0.709/0.013	0.672/0.660/0.012	0.697/0.699/0.004	0.694/0.699/0.007
$(50, 5, 0.9, 0.1)$	0.992/1.000/0.019	0.888/0.917/0.019	1.000/1.000/0.000	**1.000**/1.000/0.000
$(50, 10, 0.7, 0.4)$	0.973/0.977/0.001	0.972/0.972/0.000	0.972/0.972/0.000	0.972/1.000/0.010
$(50, 10, 0.9, 0.4)$	**1.000**/1.000/0.000	0.980/1.000/0.020	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(50, 25, 0.6, 0.1)$	0.963/0.980/0.011	0.973/0.981/0.008	0.968/0.988/0.009	0.979/0.989/0.010
$(50, 25, 0.6, 0.4)$	0.949/0.975/0.012	0.945/0.963/0.009	0.948/0.964/0.006	0.955/0.964/0.004
$(50, 25, 0.9, 0.4)$	0.977/0.991/0.008	0.976/0.991/0.008	0.980/0.991/0.004	**0.991**/0.991/0.000
$(75, 8, 0.6, 0.1)$	**0.996**/1.000/0.007	0.943/0.975/0.017	0.960/1.000/0.019	0.941/0.977/0.016
$(75, 8, 0.8, 0.3)$	**1.000**/1.000/0.000	0.995/1.000/0.009	**1.000**/1.000/0.000	**1.000**/1.000/0.000
$(75, 8, 0.9, 0.4)$	**1.000**/1.000/0.000	0.990/1.000/0.012	0.986/0.987/0.001	**1.000**/1.000/0.000
$(75, 15, 0.6, 0.2)$	0.983/0.987/0.009	0.975/0.987/0.015	0.986/0.987/0.005	0.978/0.987/0.013
$(75, 30, 0.6, 0.1)$	0.956/0.967/0.008	0.963/0.975/0.013	0.960/0.976/0.009	0.964/0.975/0.011
$(75, 30, 0.8, 0.4)$	0.957/0.969/0.009	0.958/0.965/0.006	0.959/0.971/0.007	0.956/0.966/0.007
$(75, 38, 0.9, 0.1)$	0.969/0.987/0.007	0.970/0.980/0.012	0.972/0.980/0.005	0.980/0.986/0.010
$(75, 38, 0.9, 0.4)$	0.969/0.987/0.006	0.970/0.981/0.008	0.972/0.982/0.006	0.977/0.988/0.009
Friedman's non-parametric test (mean ranking)				
Rank	11.4167	11.5625	8.375	6.8125

goal, column t_{conv} shows the average number of fitness evaluations needed by each solver to converge under the adopted stop criterion (this value is shown in thousands). Analogously to the previous experimentation, average runtimes t_{run} are also reported, measured in seconds.

The main conclusion that can be reached after analyzing these results is that BA_NMI_SH is, again, the solver that obtain best results, followed by BA_NMI_IH and WCA_Ham. In this case, PSO_NMI_B renders a lower quality than its counterparts. Furthermore, in terms of convergence, solvers that employ heuristic movement operators (BA_NMI_IH and BA_NMI_SH) yield a much better performance than *blind* alternatives. These differences are especially remarkable in comparison with WCA_Ham, providing a great advantage for both BA approaches. For

Table 8: Run times of WCA, ESA, FA, PVNS, CS, PSO, and BA using blind movement functions, measured in seconds. The `_NMI_B` suffix indicates the use of NMI as distance function and blind movements (CC_* and CE_*). The `_Ham` suffix means that the solver uses the Hamming distance as the similarity measurement function.

(V, M, p_{in}, p_{out})	WCA_Ham	ESA_Ham	FA_Ham	PVNS	CS	PSO_Ham	BA_Ham	WCA_NMI_B	FA_NMI_B	PSO_NMI_B	BA_NMI_B
$(35, 4, 0.6, 0.1)$	2.23	0.91	10.93	1.72	1.71	1.73	2.12	7.96	14.83	4.34	3.46
$(35, 4, 0.9, 0.4)$	4.48	1.03	14.55	2.53	1.78	1.91	1.88	9.64	22.54	3.20	2.05
$(35, 7, 0.6, 0.1)$	3.63	0.91	14.47	1.95	1.43	1.44	1.75	7.35	15.96	2.95	2.04
$(35, 7, 0.6, 0.4)$	4.95	0.95	15.43	1.99	1.84	1.89	1.99	10.79	14.79	3.28	2.97
$(35, 7, 0.8, 0.1)$	3.10	0.89	16.85	1.94	1.60	1.78	2.15	7.31	17.26	2.86	2.23
$(35, 7, 0.9, 0.4)$	3.19	0.89	17.20	1.79	1.44	1.70	1.83	6.71	12.62	2.96	2.20
$(35, 18, 0.6, 0.1)$	5.79	0.99	12.66	1.98	2.23	1.84	1.48	8.81	17.90	5.39	4.73
$(35, 18, 0.9, 0.4)$	4.30	1.20	12.83	2.03	2.18	1.94	1.81	9.86	17.58	6.82	5.11
$(50, 5, 0.6, 0.1)$	12.90	6.63	19.37	7.12	8.19	6.77	7.97	20.69	25.57	16.63	10.13
$(50, 5, 0.6, 0.4)$	13.95	5.86	20.48	6.67	12.05	9.53	8.89	20.45	26.01	17.42	12.26
$(50, 5, 0.9, 0.1)$	12.80	4.94	18.36	7.81	9.55	5.45	7.46	18.95	24.66	16.01	11.79
$(50, 10, 0.7, 0.4)$	14.01	4.90	19.69	6.85	8.52	6.31	9.31	20.86	25.48	13.87	9.82
$(50, 10, 0.9, 0.4)$	10.04	4.53	18.13	6.63	7.45	5.54	10.44	17.96	23.16	14.39	7.65
$(50, 25, 0.6, 0.1)$	16.98	5.72	20.95	7.45	15.54	11.72	13.88	23.88	28.54	19.20	18.02
$(50, 25, 0.6, 0.4)$	17.86	5.13	19.29	7.40	15.66	9.23	11.73	25.04	29.43	16.98	15.22
$(50, 25, 0.9, 0.4)$	19.96	5.27	20.96	9.04	14.86	8.97	8.54	24.20	30.18	20.87	18.72
$(75, 8, 0.6, 0.1)$	84.18	36.39	90.27	34.28	70.28	53.30	67.09	91.13	98.95	74.84	42.36
$(75, 8, 0.8, 0.3)$	71.04	34.56	85.95	33.76	60.07	44.02	56.62	84.65	96.79	62.06	39.56
$(75, 8, 0.9, 0.4)$	69.43	28.94	80.03	59.60	45.65	52.49	54.35	85.29	94.28	68.23	37.54
$(75, 15, 0.6, 0.2)$	82.40	56.34	90.50	59.39	56.68	60.48	64.35	92.17	96.43	74.08	44.67
$(75, 30, 0.6, 0.1)$	93.59	59.05	98.84	36.90	82.60	75.60	68.74	91.92	99.67	82.35	79.59
$(75, 30, 0.8, 0.4)$	94.59	58.10	97.95	36.10	83.34	71.34	68.32	92.40	97.53	84.35	72.76
$(75, 38, 0.9, 0.1)$	86.25	44.30	90.89	36.49	80.21	76.77	72.42	85.70	89.77	76.09	69.82
$(75, 38, 0.9, 0.4)$	87.57	44.23	92.29	35.51	80.90	72.79	76.44	89.54	92.73	81.56	76.20

Table 9: Run times of WCA, FA, PSO, and BA using heuristic functions, measured in seconds. The `_NMI_SH` suffix means the use of NMI as distance function and SH_* movement functions. The `_NMI_IH` suffix denotes the use of NMI as distance function and IH_* operators.

(V, M, p_{in}, p_{out})	WCA_NMI_SH	FA_NMI_SH	PSO_NMI_SH	BA_NMI_SH	WCA_NMI_IH	FA_NMI_IH	PSO_NMI_IH	BA_NMI_IH
$(35, 4, 0.6, 0.1)$	6.10	9.80	3.21	4.23	5.08	9.32	4.95	3.88
$(35, 4, 0.9, 0.4)$	3.34	8.95	3.01	2.23	5.66	9.76	3.50	3.27
$(35, 7, 0.6, 0.1)$	3.93	10.23	2.33	2.34	4.41	8.02	3.42	2.44
$(35, 7, 0.6, 0.4)$	5.42	10.96	3.29	2.65	5.73	8.81	4.01	3.33
$(35, 7, 0.8, 0.1)$	4.68	10.67	2.37	5.43	3.99	9.47	2.66	2.37
$(35, 7, 0.9, 0.4)$	3.83	9.74	2.33	3.15	4.13	9.40	2.87	2.58
$(35, 18, 0.6, 0.1)$	5.73	10.62	3.96	7.50	5.67	10.39	6.28	6.75
$(35, 18, 0.9, 0.4)$	6.19	12.13	4.22	8.66	5.66	12.92	7.55	6.67
$(50, 5, 0.6, 0.1)$	17.82	25.21	11.39	8.40	15.82	23.42	18.22	12.49
$(50, 5, 0.6, 0.4)$	23.95	31.26	18.75	17.64	21.78	32.96	25.94	18.33
$(50, 5, 0.9, 0.1)$	17.90	27.52	14.89	10.62	17.80	27.10	18.78	14.65
$(50, 10, 0.7, 0.4)$	17.31	30.33	13.41	12.77	17.83	30.69	18.56	13.83
$(50, 10, 0.9, 0.4)$	17.20	28.27	10.68	14.83	16.39	33.37	16.94	11.90
$(50, 25, 0.6, 0.1)$	22.99	42.36	28.85	32.50	21.77	45.61	37.72	28.84
$(50, 25, 0.6, 0.4)$	31.95	45.05	23.95	33.25	19.87	43.07	35.75	35.94
$(50, 25, 0.9, 0.4)$	32.51	47.28	25.83	34.82	22.40	44.96	31.91	37.40
$(75, 8, 0.6, 0.1)$	90.02	97.83	78.97	42.50	83.37	94.55	86.27	77.27
$(75, 8, 0.8, 0.3)$	64.70	96.73	90.58	46.11	62.88	90.14	84.25	72.34
$(75, 8, 0.9, 0.4)$	61.89	94.21	82.93	51.31	60.02	95.34	81.50	76.57
$(75, 15, 0.6, 0.2)$	92.24	99.13	91.90	54.09	84.71	97.18	80.65	65.76
$(75, 30, 0.6, 0.1)$	93.30	101.03	94.07	72.74	77.43	98.41	83.13	81.92
$(75, 30, 0.8, 0.4)$	81.33	96.96	92.39	93.51	85.49	100.07	92.71	71.96
$(75, 38, 0.9, 0.1)$	103.68	113.56	104.13	97.53	82.67	97.28	86.65	81.06
$(75, 38, 0.9, 0.4)$	83.23	97.45	86.21	80.42	79.22	93.42	83.91	80.38

Table 10: Obtained NMI results (average/best/standard deviation) for large instances using WCA_Ham, PSO_NMI_B, BA_NMI_H1 and BA_NMI_H2. Best average results have been highlighted in bold.

	WCA_Ham					PSO_NMI_B					BA_NMI_SH					BA_NMI_IH				
Dataset	Avg	Best	Std	t_{conv}	t_{run}	Avg	Best	Std	t_{conv}	t_{run}	Avg	Best	Std	t_{conv}	t_{run}	Avg	Best	Std	t_{conv}	t_{run}
$(100, 15, 40, 0.1, 5, 20)$	**0.883**	0.897	0.021	0.820	233.12	0.820	0.989	0.025	0.688	193.02	0.862	0.904	0.032	0.403	201.25	0.841	0.898	0.027	0.362	203.95
$(200, 15, 40, 0.1, 5, 20)$	**1.0**	1.0	0.0	0.531	315.54	**1.0**	1.0	0.0	0.381	307.69	**1.0**	1.0	0.0	0.378	322.06	**1.0**	1.0	0.0	0.310	319.95
$(300, 15, 40, 0.2, 10, 25)$	**1.0**	1.0	0.0	0.895	506.21	**1.0**	1.0	0.0	0.573	517.55	**1.0**	1.0	0.0	0.640	522.69	**1.0**	1.0	0.0	0.583	532.90
$(400, 15, 40, 0.2, 15, 30)$	0.992	1.0	0.007	2.413	663.59	0.998	1.0	0.001	1.007	650.47	**0.999**	1.0	0.001	0.865	642.05	0.998	1.0	0.004	0.884	647.11
$(500, 15, 40, 0.2, 15, 30)$	0.992	1.0	0.004	3.256	802.07	0.997	1.0	0.003	1.337	796.62	**1.0**	1.0	0.0	1.051	792.74	**1.0**	1.0	0.0	1.097	804.44
Friedman's non-parametric & Holm's post-hoc tests (results quality)																				
Friedman's Rank	2.4655					2.8793					2.3103					2.3448				
p_{Holm}	1.294346					0.27992					Control Solver					1.294346				
Friedman's non-parametric & Holm's post-hoc tests (convergence behavior)																				
Friedman's Rank	3.4828					3.2069					1.7241					1.5862				
p_{Holm}	0					0.000003					0.684127					Control Solver				

this second experimentation, Friedman's non-parametric test for multiple comparison and Holm's post-hoc test have been also applied to both results' quality and convergence behavior. For these statistical tests, the outcomes of all the 29 contemplated network instances (24 previous ones and 5 *large* instances) have been considered. For this reason, and for the seek of completeness, we depict in Table 11 the convergence shown by the four solvers for the first 24 cases.

Table 11: Convergence behavior of WCA_Ham, PSO_NMI_B, BA_NMI_H1, BA_NMI_H2 for the first 24 instances. t_{conv} is the average number of evaluations needed by each solver to reach the final solution.

(V, M, p_{in}, p_{out})	WCA_Ham	PSO_NMI_B	BA_NMI_SH	BA_NMI_IH	(V, M, p_{in}, p_{out})	WCA_Ham	PSO_NMI_B	BA_NMI_SH	BA_NMI_IH
	t_{conv}	t_{conv}	t_{conv}	t_{conv}		t_{conv}	t_{conv}	t_{conv}	t_{conv}
$(35, 4, 0.6, 0.1)$	0.091	0.144	0.080	0.095	$(50, 10, 0.9, 0.4)$	0.136	0.145	0.070	0.040
$(35, 4, 0.9, 0.4)$	0.132	0.127	0.062	0.036	$(50, 25, 0.6, 0.1)$	0.652	0.610	0.262	0.404
$(35, 7, 0.6, 0.1)$	0.099	0.110	0.040	0.037	$(50, 25, 0.6, 0.4)$	0.510	0.564	0.587	0.496
$(35, 7, 0.6, 0.4)$	0.167	0.178	0.086	0.054	$(50, 25, 0.9, 0.4)$	0.383	0.532	0.130	0.117
$(35, 7, 0.8, 0.1)$	0.075	0.098	0.027	0.020	$(75, 8, 0.6, 0.1)$	0.686	0.449	0.181	0.480
$(35, 7, 0.9, 0.4)$	0.081	0.091	0.023	0.024	$(75, 8, 0.8, 0.3)$	0.267	0.452	0.234	0.206
$(35, 18, 0.6, 0.1)$	0.256	0.232	0.136	0.160	$(75, 8, 0.9, 0.4)$	0.223	0.172	0.118	0.088
$(35, 18, 0.9, 0.4)$	0.124	0.211	0.083	0.079	$(75, 15, 0.6, 0.2)$	0.609	0.374	0.348	0.385
$(50, 5, 0.6, 0.1)$	0.160	0.129	0.045	0.097	$(75, 30, 0.6, 0.1)$	1.256	0.944	1.054	0.474
$(50, 5, 0.6, 0.4)$	0.486	0.457	0.320	0.240	$(75, 30, 0.8, 0.4)$	1.647	0.996	0.332	1.367
$(50, 5, 0.9, 0.1)$	0.145	0.229	0.053	0.130	$(75, 38, 0.9, 0.1)$	1.400	1.727	0.848	0.857
$(50, 10, 0.7, 0.4)$	0.179	0.183	0.123	0.115	$(75, 38, 0.9, 0.4)$	1.863	0.870	0.795	0.645

With all this, Friedman's test also supports the conclusions drawn for the first tests, in which BA_NMI_SH was underscored as the best alternative. Furthermore, after conducting both experimentations BA_NMI_IH can be highlighted as the second-best solver, followed by WCA_Ham and PSO_NMI_B. As occurred previously, Holm's post-hoc test unveils that gaps identified in the results are not statistically significant. Anyway, the combination of these findings with the results related to the convergence behavior leads us to the claim that both BA_NMI_SH and BA_NMI_IH dominate this benchmark. To be concise, Friedman's and Holm's tests sustain these assertions, proving that heuristic methods are statistically better that WCA_Ham and PSO_NMI_B in terms of convergence behavior. These tests also discover that gaps between both BA_NMI_SH and BA_NMI_SH are not significant. Finally, regarding run times, similar conclusions can be drawn from the reported results over these large instances. That is, techniques based non PSO and BA require similar computational resources and do not exhibit significant performance differences. Once again, execution times are not decisive for choosing among the techniques.

On a closing note for this second benchmark, we conclude that BA_NMI_SH and BA_NMI_IH stand out from the rest of the solver, obtaining best outcomes in terms of results quality and convergence behavior. We can also highlight that WCA_Ham and PSO_NMI_B are promising approaches for solving the community detection problem dealt with in this study, despite they stay one step below in terms of quality and completely left behind regarding convergence behavior.

5.3. Third Experimentation: Benchmark with Community Detection Algorithms

Until now, we have analyzed the performance and running times of all the 19 implemented bio-inspired solvers, and the convergence behavior of the best techniques of each category. As a result, bio-inspired schemes have been proven to efficiently deal with the community finding problem in weighted directed networks. In any case, we carry out a final set of experiments to corroborate if bio-inspired solvers can compete in terms of results quality with 6 community detection techniques from the state of the art, all suited to deal with weighted directed networks: the well-known Louvain (Louvain) and Leiden (Leiden) algorithms whose quality function to be optimized is set to the aforementioned directed weighted modularity [20, 171], a Leiden algorithm optimizing the Surprise metric [169] (Surprise), a Reichardt and Bornholdt Potts model [148] (RB Potts), a Constant Potts Model [170] (CPM), and the well-known InfoMap algorithm [151] (InfoMap). We note that all these algorithms can cope with directed weighted networks. The six considered community detection algorithms included in this third experimental stage have been executed in the same computer as the bio-inspired methods, and for each network instance 10 independent executions have been performed so as to extract performance statistics.

In this third experimentation, 17 LFR instances have been considered. The first 5 instances are the same utilized in previous experiments. The remaining 12 instances have been generated by using the same LFR benchmark generator for weighted directed networks [93]. In this case, however, we focus on instances with moderate to high values of their topological mixing coefficient, thereby posing a more challenging task for the considered algorithms. Our research hypothesis motivating this configuration is that meta-heuristics can make a difference with respect to state-of-the-art community detection algorithms specially when the hidden cluster structure of the network is composed by notably interconnected clusters. Departing from this research hypothesis, we set the network sizes of the new instances to $V \in \{100, 200, 300, 400, 500, 600\}$, with mixing coefficients equal to 0.5 and 0.7. Obviously, NMI values with respect to the true cluster distribution of every network are expected to decrease with respect to those in previous experiments, as the higher topological mixture among clusters makes it more difficult for the algorithms to infer their true structure. The values of the rest of the parameters used for creating the LFR instances are indicated by the label of each instance, in the same format as before.

Table 12 shows the results obtained in this third experimentation. In that table, the results and standard deviations obtained by the whole group of 10 techniques is depicted. Furthermore, the table is divided into two different parts. The first one is devoted to the five instances also considered in the previous experimentation. On the other hand, the second division revolves around the 12 new generated dispersed datasets. Finally, last rows illustrate the mean ranking calculated by the Friedman's non-parametric test for all the compared algorithms (the lower the rank, the better the performance). In this regard, two different Friedman's tests have been carried out. The first one using the outcomes obtained in the 12 newly generated dispersed instances, while the second has been conducted over the whole group of network instances. For each of these statistical tests, the corresponding standard deviation computed over the considered network instances is also represented to shed light on the variability of the rankings over the benchmark.

Several interesting findings can be drawn from this third experimentation. First of all, it seems clear that state-of-the-art methods perform better in the first group of instances, inferring the true community structure of the networks in almost all the cases. However, this situation is reversed for the second group of network instances, thus validating our research hypothesis. In fact, when inspecting the Friedman's test results, `BA_NMI_IH` is on average the best performing method for these instances, followed closely by `Leiden`. It is also remarkable that `BA_NMI_SH` and `PSO_NMI_B` perform better than the state-of-the-art community detection algorithms included for comparison, except for `Leiden`.

Notwithstanding the superior performance of `BA_NMI_IH` as per its lower average ranking, a closer examination at the ranking statistics unveils that rankings are quite irregular. In other words, it is fair to claim, in light of the results, that this bio-inspired solver can dominate the rest of algorithms, but just in certain network instances. In fact, the high variability of rankings occurs concurrently for all algorithms. This observation buttresses empirically the most important insight stemming from this work: it is not possible to discriminate a method that clearly performs best for each and every kind of network. This empirical finding supports us in our prospects about the future of bio-inspired computation algorithms for community detection in graphs, which we next develop in detail.

Table 12: NMI results (average/standard deviation) obtained by `WCA_Ham`, `PSO_NMI_B`, `BA_NMI_H1`, `BA_NMI_H2` and the methods from the state of the art (`Louvain`, `Surprise`, `Leiden`, `RB Potts`, `CPM` and `InfoMap`).

Instance	`WCA_Ham`	`PSO_NMI_B`	`BA_NMI_SH`	`BA_NMI_IH`	`Louvain`	`Surprise`	`Leiden`	`RB Potts`	`CPM`	`InfoMap`
	avg/std	avg/std	avg/std	avg/std	avg/std	avg/std	avg/std	avg/std	avg/std	avg/std
$(100, 15, 40, 0.1, 5, 20)$	0.883/0.021	0.820/0.025	0.862/0.032	0.841/0.027	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000
$(200, 15, 40, 0.1, 5, 20)$	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000
$(300, 15, 40, 0.2, 10, 25)$	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000
$(400, 15, 40, 0.2, 15, 30)$	0.992/0.007	0.998/0.001	0.999/0.001	0.998/0.004	1.000/0.000	1.000/0.000	1.000/0.000	0.995/0.006	1.000/0.000	1.000/0.000
$(500, 15, 40, 0.2, 15, 30)$	0.992/0.004	0.997/0.003	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	1.000/0.000	0.994/0.005	1.000/0.000	1.000/0.000
$(100, 10, 40, 0.5, 5, 20)$	0.786/0.037	0.865/0.031	0.868/0.028	0.878/0.041	1.000/0.000	0.959/0.020	1.000/0.000	0.936/0.067	0.830/0.019	0.804/0.050
$(100, 10, 50, 0.7, 5, 20)$	0.481/0021	0.455/0.034	0.446/0.025	0.425/0.013	0.316/0.019	0.503/0.030	0.327/0.033	0.313/0.034	0.503/0.021	0.309/0.029
$(200, 10, 40, 0.5, 10, 80)$	0.628/0.017	0.648/0.015	0.626/0.011	0.633/0.021	0.630/0.000	0.625/0.022	0.625/0.022	0.643/0.004	0.520/0.014	0.590/0.080
$(200, 20, 50, 0.7, 10, 80)$	0.319/0.025	0.304/0.011	0.315/0.019	0.329/0.015	0.183/0.000	0.303/0.014	0.219/0.031	0.234/0.028	0.256/0.016	0.228/0.016
$(300, 10, 40, 0.5, 10, 100)$	0.551/0.018	0.566/0.025	0.535/0.034	0.534/0.037	0.621/0.000	0.541/0.015	0.608/0.014	0.575/0.012	0.464/0.010	0.560/0.044
$(300, 20, 50, 0.7, 10, 100)$	0.485/0.029	0.525/0.020	0.513/0.027	0.511/0.020	0.488/0.000	0.534/0.021	0.554/0.029	0.459/0.051	0.489/0.013	0.309/0.029
$(400, 10, 40, 0.5, 15, 150)$	0.469/0.024	0.496/0.020	0.499/0.028	0.500/0.018	0.597/0.008	0.446/0.020	0.622/0.019	0.573/0.019	0.412/0.009	0.501/0.042
$(400, 20, 50, 0.7, 15, 150)$	0.339/0.018	0.341/0.013	0.342/0.009	0.342/0.012	0.290/0.012	0.337/0.006	0.311/0.042	0.267/0.023	0.331/0.011	0.255/0.042
$(500, 10, 40, 0.5, 20, 200)$	0.558/0.032	0.561/0.027	0.572/0.024	0.575/0.021	0.693/0.004	0.548/0.021	0.726/0.021	0.686/0.020	0.449/0.009	0.528/0.030
$(500, 20, 20, 0.7, 20, 200)$	0.234/0.032	0.242/0.020	0.280/0.018	0.287/0.010	0.274/0.000	0.208/0.009	0.268/0.023	0.161/0.030	0.214/0.007	0.129/0.011
$(600, 10, 40, 0.5, 20, 200)$	0.429/0.031	0.433/0.045	0.466/0.029	0.460/0.037	0.525/0.000	0.435/0.009	0.501/0.102	0.541/0.050	0.407/0.007	0.396/0.052
$(600, 20, 50, 0.7, 20, 200)$	0.381/0.035	0.390/0.033	0.398/0.027	0.391/0.041	0.357/0.034	0.351/0.010	0.389/0.013	0.352/0.033	0.333/0.008	0.305/0.036
Friedman's non-parametric tests (results quality) for the 12 new instances										
Friedman's Rank/std	6.0000/2.33	4.5000/1.93	4.2917/2.15	3.95832.30	4.5417/3.34	5.8333/2.45	4.0000/2.91	5.5000/3.02	7.7083/2.56	8.667/1.86
Friedman's non-parametric tests (results quality) for all the 17 big instances										
Friedman's Rank/std	6.4706/2.79	5.3235/2.42	4.7353/2.33	4.6471/2.55	4.4412/3.25	5.3235/2.94	4.2353/2.80	5.7353/3.17	6.7059/3.74	7.3824/3.89

6. A Prospect of Research Opportunities and Open Challenges

In light of the literature review in Section 2, and the experimentation with modern bio-inspired meta-heuristics carried out in Sections 4 and 5, it is unquestionable that bio-inspired computation will play a paramount role in the challenging horizon envisioned for this field. In this context, we foresee promising research directions along diverse axis in the field of community detection with this kind of solvers, among which we pause at the following ones:

- We definitely call for a profound reflection around the computational efficiency of bio-inspired solvers when facing network instances of large size. Most reported works related to this approach have so far addressed controlled problem instances of small and medium size (in terms of number of nodes V). This is also the case of the experimental part of this manuscript, which has assessed the performance of modern heuristics over networks of up to $V = 600$ nodes.

However, the scales featured by real-world problems can get several orders of magnitude higher, which not only hinders the computational efficiency at which this family of heuristics perform, but also compromises their convergence properties. One possible workaround to this issue is to hybridize message passing methods and bio-inspired optimization algorithms. These methods have been applied in several previous studies, such as [92], in which the so-called APCOM (affinity propagation for community finding) scheme is applied to network partition problems. Another inspiring study is [197], in which the application of Belief Propagation (that can be regarded as a message passing scheme) and the so-called Cavity Method is explored; or the research in [160], which delves into such an approach for weighted community detection. We firmly believe that a step further should be taken in the development of solvers for this problem by delving into the hybridization of concepts coming from different worlds, such as the bio-inspired computation and message passing methods. Besides the straightforward benefits in terms of efficiency, exchanging messages among locally implemented heuristics over the network can be regarded as an ad-hoc, graph-sensitive flavor of distributed Evolutionary Algorithms [60], which may yield additional profits in terms of convergence speed and local optima avoidance [156]. Other inherently distributed heuristics also deserve further attention, such as Stochastic Diffusion Search [36] which, in addition, can help obtain theoretical insights on its performance as a community search algorithm thanks to its solid theoretical framework.

- In line with our postulations above, future research efforts in community partition with bio-inspired heuristics should also account for tools and frameworks arising in other disciplines of knowledge, not necessarily related to Computer Science whatsoever. For instance, an interesting research path can be discerned in regards to the Nobel-winning Matching Theory framework [112, 152], which unleashes a number of centralized and distributed matching algorithms – e.g. the renowned Gale-Shapley's Deferred Acceptance algorithm [49] – that can be adapted to enable an efficient framework for hierarchical community detection over graphs. To this end, a formulation of the utility function (*preference*) of a generic node with respect to another should be undertaken to properly establish the criterion to associate nodes to each other. Furthermore, caveats such as the eventual existence of inter-dependencies between the players preferences (*externalities*) and its consequences in terms of matching stability should be considered and resolved over the graph, for instance by circumscribing the utility computation and subsequent matching process to local contexts of the node at hand as per a measure of reachability with respect to the rest of nodes in the network. Another possible research avenue worth exploring in the future is the adoption of graph-theoretic elements that have hitherto been overseen by the heuristic community, such as the concept of group-level centrality [145], the dispersion among pair of nodes [10], or the rich club phenomenon [118], with potential implications in the discovery of hierarchical communities. Surely the ever-growing substrate of graph-theoretic measures that continuously emerge from the literature will stimulate new ad-hoc solvers trading their meta-heuristic nature for a best performance in complex community partition tasks.

- In partial consonance with the previous point, the main conclusion drawn from the third stage of our experiments (*"No community detection algorithm performs best in each and every network instance"*) concurs with the postulates of the No Free Lunch Theorem for optimization [181], and confronts a controversial trend noted in the community toward claiming that the performance of newly proposed bio-inspired meta-heuristics is superior for *a problem* rather than *for the instances considered in the study*. Generalizing the so-claimed superiority of a bio-inspired algorithm to *any* new problem instance, however, requires more exhaustive experiments than those usually reported in related studies.

  For this reason, we decidedly advocate for problem-solving strategies similar to those adopted in already existing tools for community detection, such as SurpriseMe [3]. This exemplified tool integrates a set of community detection algorithms (all hinging on the Surprise metric), which are applied sequentially for a given network instance under test. The solution scoring the highest metric value is then selected no matter which particular community detection algorithm that produced it. This search strategy implicitly acknowledges that no community detection algorithm can be declared to perform best for any network instance to be analyzed. Given the empirical results of our experiments in this overview, we encourage the community to pursue research towards the development of more bio-inspired community detection algorithms for their inclusion in this type of integrated tools, instead of misleadingly racing to achieve superior performance scores over limited experimental setups.

- Placing again the computational issues derived from processing massive networks under the spotlight, a bio-inspired paradigm of utmost applicability focuses on imprinting modifications to classical optimization heuristics aimed at coping with problem formulations comprising a high number of variables.

This portfolio of new optimization algorithms, collectively referred to as *large-scale global optimization*, can unchain unprecedented approaches to community detection in massive networks composed by thousands of nodes. In this context, methods such as the Multiple Offspring Sampling [97] or SHADEILS [124] should be explored for this optimization task. Another option to simultaneously alleviate the computational complexity of the solver and improve its performance can be found within the area of cooperative co-evolutionary algorithms [116], which could divide the community partition problem into two different subproblems (for instance, one resolving the optimal partitioning of the network, the other for the overlapping between such communities), each endowed with a sub-population evolved along generations of the search heuristic. An additional research trend related to computational efficiency would be the design and implementation of self-adaptive solvers [91], capable of modifying and selecting the most appropriate optimization criteria and/or parametric configuration of the solver during the search process. All in all, we foresee that these heuristic variants, along with their implementation over Big Data frameworks for massive data processing such as Apache Hadoop or Spark could significantly boost the adoption of bio-inspired heuristics for community partition over large-scale graphs.

- The community has lately moved forward to formulate sophisticated variants of the community detection problem, always in an attempt at reliably modeling real-life applications for which an increased insight on the unveiled community structure is sought. This rationale is indeed the core of relatively recent directions in the field, such as 1) the incorporation of multiple objectives to account for e.g. the balance between internal and external connectivity of clusters [17, 203, 207]; 2) challenging graphs instances such as dynamic networks [48] or bipartite graphs [166]; or 3) new clustering quality indicators, evolved from classical ones to better reflect certain aspects of the inferred community structure (such as Surprise, which was proposed as a well-behaved clustering metric for community distributions of varying size [2]; or Fuzzy Modularity Maximization for fuzzy community detection in overlapping networks [163]). Many other examples beyond this excerpt of alternative problem formulations still remain far less addressed in the literature (such as the discovery of motifs [18] or the partitioning of multi-layered network structures [21]), thereby opening up an uncharted research niche for the application of bio-inspired heuristics.

7. Conclusions

This manuscript has gravitated on community detection problems over networks through the perspective of bio-inspired optimization. First, we have briefly reviewed the recent history of this field, highlighting some of the most valuable works published in last years with a focus on modern bio-inspired solvers. This literature overview has been complemented by the empirical insights drawn from a comprehensive experimental study focused on detecting communities over weighted directed graphs. The discovery of optimal partitions is modeled as an optimization problem driven by an adaptation of the well-known modularity measure to accommodate the directional and weighted nature of the edges of the network. Seven different bio-inspired heuristics have been adapted to efficiently tackle the formulated optimization problem, namely, Water Cycle Algorithm, Particle Swarm Optimization, Cuckoo Search, Firefly Algorithm, Bat Algorithm, Evolutionary Simulated Annealing and Population-based Variable Neighborhood Search. Furthermore, two different similarity measures have been used as a core component of eight heterogeneous distance-based movement operators: Normalized Mutual Information (NMI) and Hamming Distance. In overall, 19 different solving schemes have been developed for the aforementioned problem, which result from combinations of the search heuristics, similarity functions and movement operators.

The performance of these techniques has been assessed and compared over a benchmark of 24 network instances of small size (from 35 to 75 nodes), as well as over five instances of larger size comprising up to 500 nodes. NMI with respect to their *ground of truth* partition has been adopted as the comparison criterion. The obtained results have revealed that BA with heuristic operators and using NMI as its similarity measure dominates the benchmark. A third experimentation has been carried out with the main goal of confirming that bio-inspired approaches can be competitive with respect to other established community detection methods. To do that, six different community detection algorithms have been used for comparison over 17 LFR big instances composed by 100 to 600 nodes.

An interesting outcome of this last set of experiments is that no clear winning algorithm outstands over the considered network instances. This finding is in consonance with the strategy followed by existing tools for community detection, which rely on the sequential application of different solvers in search for the one performing best in terms of the considered fitness function. In order to quantitatively support this statement, we have analyzed the mean and standard deviation of the rankings of all the compared algorithms. As we now conclude in the end of the experimental section of the paper, such ranking statistics confirm two interesting claims:

1. The proposed bio-inspired algorithms perform on average better than standard solvers in some network instances, particularly with highly overlapping communities.

2. Considering the high variability of the algorithms rankings across network instances, there is no clear winner in the benchmark.

This two-fold conclusion is among the research directions that conclude this overview, summarizing our envisioned future of the field. We have identified several inspiring challenges and opportunities which should congregate most of the global research efforts in the coming years. Among them, we advocate for the synergistic hybridization of solvers developed by graph theorists and experts in meta-heuristics along the years. It is in the mixture and complementarity of technical approaches from different disciplines where the community may find most of the potential to undertake network partitioning problems of unprecedented complexity. For this to occur, we eagerly call for more efforts invested in research areas such as Game Theory, message passing algorithms, distributed Evolutionary Computation, many-objective optimization and large-scale global optimization, with an emphasis on blending such disciplines in real-world network instances and problem formulations.

We unequivocally foresee an exciting future for the community in this research avenue, which we should face just like edges of a network: by connecting together multifaceted knowledge disciplines.

Acknowledgements

Eneko Osaba and Javier Del Ser would like to thank the Basque Government for its funding support through the EMAITEK program. David Camacho also thanks the Spanish Ministry of Science, Education and Competitiveness (MINECO), the European Regional Development Fund (FEDER) and the Comunidad Autonoma de Madrid for their support under grants TIN2017-85727-C4-3-P (DeepBio) and P2018/TCS-4566 (CYNAMON).

References

[1] Ahmed, M. M., Wazery, Y. M., 2019. Genetic algorithms for discovering community cores hidden within multidimensional social networks. In:

2019 International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE, pp. 142–148.

[2] Aldecoa, R., Marín, I., 2011. Deciphering network community structure by surprise. PloS one 6 (9), e24195.

[3] Aldecoa, R., Marín, I., 2013. Surpriseme: an integrated tool for network community structure characterization using surprise maximization. Bioinformatics 30 (7), 1041–1042.

[4] Altunbey, F., Alatas, B., 2015. Overlapping community detection in social networks using parliamentary optimization algorithm. International Journal of Computer Networks and Applications 2 (1), 12–19.

[5] Amelio, A., Pizzuti, C., 2013. Community mining in signed networks: a multiobjective approach. In: Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining. ACM, pp. 95–99.

[6] Amelio, A., Pizzuti, C., 2014. Community detection in multidimensional networks. In: 2014 IEEE 26th International Conference on Tools with Artificial Intelligence. IEEE, pp. 352–359.

[7] Amelio, A., Pizzuti, C., 2017. Evolutionary clustering for mining and tracking dynamic multilayer networks. Computational Intelligence 33 (2), 181–209.

[8] Amiri, B., Hossain, L., Crawford, J. W., Wigand, R. T., 2013. Community detection in complex networks: Multi–objective enhanced firefly algorithm. Knowledge-Based Systems 46, 1–11.

[9] Atay, Y., Koc, I., Babaoglu, I., Kodaz, H., 2017. Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms. Applied Soft Computing 50, 194–211.

[10] Backstrom, L., Kleinberg, J., 2014. Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook. In: Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing. ACM, pp. 831–841.

[11] Bai, L., Liang, J., Du, H., Guo, Y., 2018. A novel community detection algorithm based on simplification of complex networks. Knowledge-Based Systems 143, 58–64.

[12] Bara'a, A. A., Khoder, H. S., 2016. A new multi-objective evolutionary framework for community mining in dynamic social networks. Swarm and Evolutionary Computation 31, 90–109.

[13] Battiston, S., Glattfelder, J. B., Garlaschelli, D., Lillo, F., Caldarelli, G., 2010. The structure of financial networks. In: Network Science. Springer, pp. 131–163.

[14] Bedi, P., Sharma, C., 2016. Community detection in social networks. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 6 (3), 115–135.

[15] Bellaachia, A., Bari, A., 2011. Sfloscan: A biologically-inspired data mining framework for community identification in dynamic social networks. In: 2011 IEEE Symposium on Swarm Intelligence. IEEE, pp. 1–8.

[16] Bello-Orgaz, G., Jung, J. J., Camacho, D., 2016. Social big data: Recent achievements and new challenges. Information Fusion 28, 45–59.

[17] Bello-Orgaz, G., Salcedo-Sanz, S., Camacho, D., 2018. A multi-objective genetic algorithm for overlapping community detection based on edge encoding. Information Sciences 462, 290–314.

[18] Benson, A. R., Gleich, D. F., Leskovec, J., 2016. Higher-order organization of complex networks. Science 353 (6295), 163–166.

[19] Bibiks, K., Hu, Y. F., Li, J.-P., Pillai, P., Smith, A., 2018. Improved discrete cuckoo search for the resource-constrained project scheduling problem. Applied Soft Computing.

[20] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008 (10), P10008.

[21] Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C. I., Gómez-Gardenes, J., Romance, M., Sendina-Nadal, I., Wang, Z., Zanin, M., 2014. The structure and dynamics of multilayer networks. Physics Reports 544 (1), 1–122.

[22] Bondy, J. A., Murty, U. S. R., et al., 1976. Graph theory with applications. Vol. 290. Citeseer.

[23] Cai, Q., Gong, M., Ma, L., Jiao, L., 2015. A novel clonal selection algorithm for community detection in complex networks. Computational Intelligence 31 (3), 442–464.

[24] Cai, Q., Gong, M., Ma, L., Ruan, S., Yuan, F., Jiao, L., 2015. Greedy discrete particle swarm optimization for large-scale social network clustering. Information Sciences 316, 503–516.

[25] Cai, Q., Gong, M., Shen, B., Ma, L., Jiao, L., 2014. Discrete particle swarm optimization for identifying community structures in signed social networks. Neural Networks 58, 4–13.

[26] Cai, Q., Ma, L., Gong, M., Tian, D., 2016. A survey on network community detection based on evolutionary computation. International Journal of Bio-Inspired Computation 8 (2), 84–98.

[27] Cai, Y., Shi, C., Dong, Y., Ke, Q., Wu, B., 2011. A novel genetic algorithm for overlapping community detection. In: International conference on advanced data mining and applications. Springer, pp. 97–108.

[28] Chaitanya, K., Somayajulu, D., Krishna, P. R., 2018. A pso based community detection in social networks with node attributes. In: 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1–6.

[29] Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A., Bhowmick, S., 2014. On the permanence of vertices in network communities. In: ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 1396–1405.

[30] Chang, H., Feng, Z., Ren, Z., 2016. Community detection using dual-representation chemical reaction optimization. IEEE transactions on cybernetics 47 (12), 4328–4341.

[31] Chen, A.-l., Yang, G.-k., Wu, Z.-m., 2006. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University-Science A 7 (4), 607–614.

[32] Chen, B., Chen, L., Chen, Y., 2012. Detecting community structure in networks based on ant colony optimization. In: Proceedings of the International Conference on Information and Knowledge Engineering (IKE). The Steering Committee of The World Congress in Computer Science, Computer , p. 1.

[33] Chen, J., Liu, D., Hao, F., Wang, H., 2019. Community detection in dynamic signed network: an intimacy evolutionary clustering algorithm. Journal of Ambient Intelligence and Humanized Computing, 1–10.

[34] Cheng, F., Cui, T., Su, Y., Niu, Y., Zhang, X., 2018. A local information based multi-objective evolutionary algorithm for community detection in complex networks. Applied Soft Computing 69, 357–367.

[35] Chintalapudi, S. R., Prasad, M. K., 2015. A survey on community detection algorithms in large scale real world networks. In: 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, pp. 1323–1327.

[36] De, M. K., Slawomir, N. J., Mark, B., 2006. Stochastic diffusion search: Partial function evaluation in swarm intelligence dynamic optimisation. In: Stigmergic optimization. Springer, pp. 185–207.

[37] Del Ser, J., Lobo, J. L., Villar-Rodriguez, E., Bilbao, M. N., Perfecto, C., 2016. Community detection in graphs based on surprise maximization using firefly heuristics. In: IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 2233–2239.

[38] Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello, C. A. C., Herrera, F., 2019. Bio-inspired computation: Where we stand and what's next. Swarm and Evolutionary Computation 48, 220–250.

[39] Deng, K., Liu, X., Li, W., 2016. Community detection in complex networks based on improved genetic algorithm and local optimization. International Journal of Grid and Distributed Computing 9 (10), 357–374.

[40] Derrac, J., García, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1 (1), 3–18.

[41] Dickinson, B., Valyou, B., Hu, W., 2013. A genetic algorithm for identifying overlapping communities in social networks using an optimized search space. Social Networking 2 (04), 193.

[42] Eskandar, H., Sadollah, A., Bahreininejad, Ardeshir, Hamdi, M., 2012. Water cycle algorithm a novel metaheuristic optimization method for solving constrained engineering optimization problems. Applied Soft Computing 110 (111), 151–166.

[43] Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M., 2012. Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures 110, 151–166.

[44] Falkenauer, E., 1998. Genetic algorithms and grouping problems. Wiley & Sons., Inc., New York, NY, USA.

[45] Fan, H., Zhong, Y., Zeng, G., 2018. Overlapping community detection based on discrete biogeography optimization. Applied Intelligence, 1–13.

[46] Ferrara, E., 2017. Contagion dynamics of extremist propaganda in social networks. Information Sciences 418, 1–12.

[47] Fister, I., Yang, X.-S., Fister, D., 2014. Cuckoo search: a brief literature review. In: Cuckoo search and firefly algorithm. Springer, pp. 49–62.

[48] Folino, F., Pizzuti, C., 2014. An evolutionary multiobjective approach for community discovery in dynamic networks. IEEE Transactions on Knowledge and Data Engineering 26 (8), 1838–1852.

[49] Gale, D., Shapley, L. S., 1962. College admissions and the stability of marriage. The American Mathematical Monthly 69 (1), 9–15.

[50] Gao, C., Chen, Z., Li, X., Tian, Z., Li, S., Wang, Z., 2018. Multiobjective discrete particle swarm optimization for community detection in dynamic networks. EPL (Europhysics Letters) 122 (2), 28001.

[51] Ghaffaripour, Z., Abdollahpouri, A., Moradi, P., 2016. A multi-objective genetic algorithm for community detection in weighted networks. In: 2016 Eighth International Conference on Information and Knowledge Technology (IKT). IEEE, pp. 193–199.

[52] Gmati, H., Mouakher, A., Gonzalez-Pardo, A., Camacho, D., 2019. A new algorithm for communities detection in social networks with node attributes. Journal of Ambient Intelligence and Humanized Computing, 1–13.

[53] Goldberg, D. E., 1989. Genetic algorithms in search, optimization and machine learning.

[54] Gong, M., Cai, Q., Chen, X., Ma, L., 2013. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. IEEE Transactions on Evolutionary Computation 18 (1), 82–97.

[55] Gong, M., Cai, Q., Chen, X., Ma, L., 2014. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. IEEE Transactions on Evolutionary Computation 18 (1), 82–97.

[56] Gong, M., Chen, X., Ma, L., Zhang, Q., Jiao, L., 2013. Identification of multi-resolution network structures with multi-objective immune algorithm. Applied Soft Computing 13 (4), 1705–1717.

[57] Gong, M., Hou, T., Fu, B., Jiao, L., 2011. A non-dominated neighbor immune algorithm for community detection in networks. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM, pp. 1627–1634.

[58] Gong, M., Ma, L., Zhang, Q., Jiao, L., 2012. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Physica A: Statistical Mechanics and its Applications 391 (15), 4050–4060.

[59] Gong, M.-G., Zhang, L.-J., Ma, J.-J., Jiao, L.-C., 2012. Community detection in dynamic social networks based on multiobjective immune algorithm. Journal of Computer Science and Technology 27 (3), 455–467.

[60] Gong, Y.-J., Chen, W.-N., Zhan, Z.-H., Zhang, J., Li, Y., Zhang, Q., Li, J.-J., 2015. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. Applied Soft Computing 34, 286–300.

[61] Gonzalez-Pardo, A., Camacho, D., 2018. Design of japanese tree frog algorithm for community finding problems. In: International Conference on Intelligent Data Engineering and Automated Learning. Springer, pp. 307–315.

[62] Gonzalez-Pardo, A., Jung, J. J., Camacho, D., 2017. Aco-based clustering for ego network analysis. Future Generation Computer Systems 66, 160–170.

[63] Gross, J. L., Yellen, J., 2005. Graph theory and its applications. Chapman and Hall/CRC.

[64] Guendouz, M., Amine, A., Hamou, R. M., 2017. A discrete modified fireworks algorithm for community detection in complex networks. Applied Intelligence 46 (2), 373–385.

[65] Guendouz, M., Amine, A., Hamou, R. M., 2018. Penguins search optimization algorithm for community detection in complex networks. International Journal of Applied Metaheuristic Computing (IJAMC) 9 (1), 1–14.

[66] Guerrero, M., Montoya, F. G., Baños, R., Alcayde, A., Gil, C., 2017. Adaptive community detection in complex networks using genetic algorithms. Neurocomputing 266, 101–113.

[67] Guo, X., Su, J., Zhou, H., Liu, C., Cao, J., Li, L., 2019. Community detection based on genetic algorithm using local structural similarity. IEEE Access, 1–17.

[68] Hafez, A. I., Zawbaa, H. M., Hassanien, A. E., Fahmy, A. A., 2014. Networks community detection using artificial bee colony swarm optimization. In: International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014. Springer, pp. 229–239.

[69] Han, X., Chen, D., Yang, H., 2019. A semantic community detection algorithm based on quantizing progress. Complexity 2019.

[70] Hariz, W. A., Abdulhalim, M. F., et al., 2016. Improving the performance of evolutionary multi-objective co-clustering models for community detection in complex social networks. Swarm and Evolutionary Computation 26, 137–156.

[71] Harris, J. M., Hirst, J. L., Mossinghoff, M. J., 2008. Combinatorics and graph theory. Vol. 2. Springer.

[72] Hassan, E. A., Hafez, A. I., Hassanien, A. E., Fahmy, A. A., 2015. A discrete bat algorithm for the community detection problem. In: International

Conference on Hybrid Artificial Intelligence Systems. Springer, pp. 188–199.

[73] He, D., Liu, J., Liu, D., Jin, D., Jia, Z., 2011. Ant colony optimization for community detection in large-scale complex networks. In: 2011 Seventh International Conference on Natural Computation. Vol. 2. IEEE, pp. 1151–1155.

[74] He, D., Wang, Z., Yang, B., Zhou, C., 2009. Genetic algorithm with ensemble learning for detecting community structure in complex networks. In: 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology. IEEE, pp. 702–707.

[75] Hruschka, E. R., Campello, R. J., Freitas, A. A., et al., 2009. A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 39 (2), 133–155.

[76] Huang, C., Hu, B., Yang, R., Wu, G., 2018. Snmfp: A two-stage approach to community detection in signed networks. Physica A: Statistical Mechanics and its Applications 510, 754–764.

[77] Huang, F., Li, X., Zhang, S., Zhang, J., Chen, J., Zhai, Z., 2017. Overlapping community detection for multimedia social networks. IEEE Transactions on Multimedia 19 (8), 1881–1893.

[78] Jami, V., Reddy, G. R. M., 2016. A hybrid community detection based on evolutionary algorithms in social networks. In: 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS). IEEE, pp. 1–6.

[79] Jaradat, A. S., Hamad, S. B., 2018. Community structure detection using firefly algorithm. International Journal of Applied Metaheuristic Computing (IJAMC) 9 (4), 52–70.

[80] Javed, M. A., Younis, M. S., Latif, S., Qadir, J., Baig, A., 2018. Community detection in networks: A multidisciplinary review. Journal of Network and Computer Applications 108, 87–101.

[81] Ji, J., Jiao, L., Yang, C., Liu, J., 2016. A multiagent evolutionary method for detecting communities in complex networks. Computational Intelligence 32 (4), 587–614.

[82] Ji, P., Zhang, S., Zhou, Z., 2019. A decomposition-based ant colony optimization algorithm for the multi-objective community detection. Journal of Ambient Intelligence and Humanized Computing, 1–16.

[83] Jiang, L., Shi, L., Liu, L., Yao, J., Yuan, B., Zheng, Y., 2019. An efficient evolutionary user interest community discovery model in dynamic social networks for internet of people. IEEE Internet of Things Journal.

[84] Joint Research Lab, 2019. Weighted directed networks dataset. `http://jrlab.science/software`.

[85] Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of global optimization 39 (3), 459–471.

[86] Karimi-Majd, A.-M., Fathian, M., Amiri, B., 2015. A hybrid artificial immune network for detecting communities in complex networks. Computing 97 (5), 483–507.

[87] Kennedy, J., 2011. Particle swarm optimization. In: Encyclopedia of machine learning. Springer, pp. 760–766.

[88] Khan, B. S., Niazi, M. A., 2017. Network community detection: A review and visual survey. arXiv preprint arXiv:1708.00977.

[89] Kianian, S., Khayyambashi, M. R., Movahhedinia, N., 2017. Fuseo: Fuzzy semantic overlapping community detection. Journal of Intelligent & Fuzzy Systems 32 (6), 3987–3998.

[90] Kim, J., Lee, J.-G., 2015. Community detection in multi-layer graphs: A survey. ACM SIGMOD Record 44 (3), 37–48.

[91] Kramer, O., 2008. Self-adaptive heuristics for evolutionary computation. Vol. 147. Springer.

[92] Lai, D., Nardini, C., Lu, H., 2011. Partitioning networks into communities by message passing. Physical review E 83 (1), 016115.

[93] Lancichinetti, A., Fortunato, S., 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E 80 (1), 016118.

[94] Lancichinetti, A., Fortunato, S., 2009. Community detection algorithms: a comparative analysis. Physical review E 80 (5), 056117.

[95] Lara-Cabrera, R., Gonzalez-Pardo, A., Camacho, D., 2017. Statistical analysis of risk assessment factors and metrics to evaluate radicalisation in twitter. Future Generation Computer Systems.

[96] Lara-Cabrera, R., Pardo, A. G., Benouaret, K., Faci, N., Benslimane, D., Camacho, D., 2017. Measuring the radicalisation risk in social networks. IEEE Access 5, 10892–10900.

[97] LaTorre, A., Muelas, S., Peña, J.-M., 2012. Multiple offspring sampling in large scale global optimization. In: 2012 IEEE Congress on Evolutionary Computation. IEEE, pp. 1–8.

[98] Leicht, E. A., Newman, M. E., 2008. Community structure in directed networks. Physical review letters 100 (11), 118703.

[99] Leskovec, J., et al., 2010. Stanford network analysis project. http://snap. stanford. edu.

[100] Li, L., Jiao, L., Zhao, J., Shang, R., Gong, M., 2017. Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering. Pattern Recognition 63, 1–14.

[101] Li, M., Liu, J., 2018. A link clustering based memetic algorithm for overlapping community detection. Physica A: Statistical Mechanics and its Applications 503, 410–423.

[102] Li, S., Chen, Y., Du, H., Feldman, M. W., 2010. A genetic algorithm with local search strategy for improved detection of community structure. Complexity 15 (4), 53–60.

[103] Li, X., Zhou, S., Liu, J., Lian, G., Chen, G., Lin, C.-W., 2019. Communities detection in social network based on local edge centrality. Physica A: Statistical Mechanics and its Applications 531, 121552.

[104] Li, Y., Wang, Y., Chen, J., Jiao, L., Shang, R., 2015. Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization. Journal of Heuristics 21 (4), 549–575.

[105] Li, Y.-H., Wang, J.-Q., Wang, X.-J., Zhao, Y.-L., Lu, X.-H., Liu, D.-L., 2017. Community detection based on differential evolution using social spider optimization. Symmetry 9 (9), 183.

[106] Li, Z., Dey, N., Ashour, A. S., Tang, Q., 2018. Discrete cuckoo search algorithms for two-sided robotic assembly line balancing problem. Neural Computing and Applications 30 (9), 2685–2696.

[107] Li, Z., He, L., Li, Y., 2016. A novel multiobjective particle swarm optimization algorithm for signed network community detection. Applied Intelligence 44 (3), 621–633.

[108] Li, Z., Liu, J., Wu, K., 2018. A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks. IEEE transactions on cybernetics 48 (7), 1963–1976.

[109] Liu, B., Wang, C., Wang, C., Yuan, Y., 2015. A new algorithm for overlapping community detection. In: 2015 IEEE International Conference on Information and Automation. IEEE, pp. 813–816.

[110] Liu, C., Liu, J., Jiang, Z., 2016. An improved multi-objective evolutionary algorithm for simultaneously detecting separated and overlapping communities. Natural Computing 15 (4), 635–651.

[111] Liu, R., Liu, J., He, M., 2019. A multi-objective ant colony optimization with decomposition for community detection in complex networks. Transactions of the Institute of Measurement and Control 41 (9), 2521–2534.

[112] Lovász, L., Plummer, M. D., 2009. Matching theory. Vol. 367. American Mathematical Soc.

[113] Lu, Z., Sun, X., Wen, Y., Cao, G., La Porta, T., 2014. Algorithms and applications for community detection in weighted networks. IEEE Transactions on Parallel and Distributed Systems 26 (11), 2916–2926.

[114] Ma, L., Gong, M., Yan, J., Liu, W., Wang, S., 2018. Detecting composite communities in multiplex networks: A multilevel memetic algorithm. Swarm and evolutionary computation 39, 177–191.

[115] Ma, X., Dong, D., 2017. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. IEEE Transactions on Knowledge and Data Engineering 29 (5), 1045–1058.

[116] Ma, X., Li, X., Zhang, Q., Tang, K., Liang, Z., Xie, W., Zhu, Z., 2018. A survey on cooperative co-evolutionary algorithms. IEEE Transactions on Evolutionary Computation, in press.

[117] Majumder, A., Laha, D., 2016. A new cuckoo search algorithm for 2-machine robotic cell scheduling problem with sequence-dependent setup times. Swarm and Evolutionary Computation 28, 131–143.

[118] McAuley, J. J., da Fontoura Costa, L., Caetano, T. S., 2007. Rich-club phenomenon across complex network hierarchies. Applied Physics Letters 91 (8), 084103.

[119] Messaoudi, I., Kamel, N., 2019. Community detection using fireworks optimization algorithm. International Journal on Artificial Intelligence Tools 28 (03), 1950010.

[120] Messaoudi, I., Kamel, N., 2019. A multi-objective bat algorithm for community detection on dynamic social networks. Applied Intelligence, 1–18.

[121] Mirsaleh, M. R., Meybodi, M. R., 2016. A michigan memetic algorithm for solving the community detection problem in complex network. Neurocomputing 214, 535–545.

[122] Mittal, R., Bhatia, M., 2019. Analysis of multiplex social networks using nature-inspired algorithms. In: Nature-Inspired Algorithms for Big Data Frameworks. IGI Global, pp. 290–318.

[123] Moayedikia, A., 2018. Multi-objective community detection algorithm with node importance analysis in attributed networks. Applied Soft Computing 67, 434–451.

[124] Molina, D., LaTorre, A., Herrera, F., 2018. Shade with iterative local search for large-scale global optimization. In: 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1–8.

[125] Moradi, M., Parsa, S., 2019. An evolutionary method for community detection using a novel local search strategy. Physica A: Statistical Mechanics and its Applications 523, 457–475.

[126] Mu, C., Zhang, J., Jiao, L., 2014. An intelligent ant colony optimization for community detection in complex networks. In: 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 700–706.

[127] Mu, C.-H., Xie, J., Liu, Y., Chen, F., Liu, Y., Jiao, L.-C., 2015. Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks. Applied Soft Computing 34, 485–501.

[128] Naeni, L. M., Berretta, R., Moscato, P., 2015. Ma-net: A reliable memetic algorithm for community detection by modularity optimization. In: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1. Springer, pp. 311–323.

[129] Naser, A. M. A., Alshattnawi, S., 2014. An artificial bee colony (abc) algorithm for efficient partitioning of social networks. International Journal of Intelligent Information Technologies (IJIIT) 10 (4), 24–39.

[130] Nepusz, T., Petróczi, A., Négyessy, L., Bazsó, F., 2008. Fuzzy communities and the concept of bridgeness in complex networks. Physical Review E 77 (1), 016107.

[131] Newman, M. E., 2004. Analysis of weighted networks. Physical review E 70 (5), 056131.

[132] Newman, M. E., 2004. Coauthorship networks and patterns of scientific collaboration. Proceedings of the national academy of sciences 101 (suppl 1), 5200–5205.

[133] Newman, M. E., Girvan, M., 2004. Finding and evaluating community structure in networks. Physical review E 69 (2), 026113.

[134] Niu, X., Si, W., Wu, C. Q., 2017. A label-based evolutionary computing approach to dynamic community detection. Computer Communications 108, 110–122.

[135] Osaba, E., Carballedo, R., Diaz, F., Onieva, E., Masegosa, A., Perallos, A., 2018. Good practice proposal for the implementation, presentation, and comparison of metaheuristics for solving routing problems. Neurocomputing 271, 2–8.

[136] Osaba, E., Del Ser, J., Camacho, D., Galvez, A., Iglesias, A., Fister, I., 2018. Community detection in weighted directed networks using nature-inspired heuristics. In: International Conference on Intelligent Data Engineering and Automated Learning. Springer, pp. 325–335.

[137] Osaba, E., Del Ser, J., Sadollah, A., Bilbao, M. N., Camacho, D., 2018. A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem. Applied Soft Computing 71, 277–290.

[138] Osaba, E., Yang, X.-S., Diaz, F., Onieva, E., Masegosa, A. D., Perallos, A., 2017. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. Soft Computing 21 (18), 5295–5308.

[139] Osaba, E., Yang, X.-S., Fister Jr, I., Del Ser, J., Lopez-Garcia, P., Vazquez-Pardavila, A. J., 2019. A discrete and improved bat algorithm for solving a medical goods distribution problem with pharmacological waste collection. Swarm and Evolutionary Computation 44, 273–286.

[140] Ouaarab, A., Ahiod, B., Yang, X.-S., 2014. Discrete cuckoo search algorithm for the travelling salesman problem. Neural Computing and Applications 24 (7-8), 1659–1669.

[141] Panizo, A., Bello-Orgaz, G., Camacho, D., 2018. A genetic algorithm with local search based on label propagation for detecting dynamic communities. In: International Symposium on Intelligent and Distributed Computing. Springer, pp. 319–328.

[142] Pizzuti, C., 2011. A multiobjective genetic algorithm to find communities in complex networks. IEEE Transactions on Evolutionary Computation 16 (3), 418–430.

[143] Pizzuti, C., 2018. Evolutionary computation for community detection in networks: a review. IEEE Transactions on Evolutionary Computation 22 (3), 464–483.

[144] Pizzuti, C., Socievole, A., 2018. A genetic algorithm for community detection in attributed graphs. In: International Conference on the Applications of Evolutionary Computation. Springer, pp. 159–170.

[145] Puzis, R., Yagil, D., Elovici, Y., Braha, D., 2009. Collaborative attack on internet users' anonymity. Internet Research 19 (1), 60–77.

[146] Rahimi, S., Abdollahpouri, A., Moradi, P., 2018. A multi-objective particle swarm optimization algorithm for community detection in complex networks. Swarm and Evolutionary Computation 39, 297–309.

[147] Rao, J., Du, H., Yan, X., Liu, C., 2016. Detecting overlapping community in social networks based on fuzzy membership degree. In: International Conference on Computational Social Networks. Springer, pp. 99–110.

[148] Reichardt, J., Bornholdt, S., 2006. Statistical mechanics of community detection. Physical Review E 74 (1), 016110.

[149] Reihanian, A., Feizi-Derakhshi, M.-R., Aghdasi, H. S., 2017. Community detection in social networks with node attributes based on multi-objective biogeography based optimization. Engineering Applications of Artificial Intelligence 62, 51–67.

[150] Rossetti, G., Cazabet, R., 2018. Community discovery in dynamic networks: a survey. ACM Computing Surveys (CSUR) 51 (2), 35.

[151] Rosvall, M., Bergstrom, C. T., 2008. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences 105 (4), 1118–1123.

[152] Roth, A. E., Sotomayor, M., 1992. Two-sided matching. Handbook of game theory with economic applications 1, 485–541.

[153] Said, A., Abbasi, R. A., Maqbool, O., Daud, A., Aljohani, N. R., 2018. Cc-ga: A clustering coefficient based genetic algorithm for detecting communities in social networks. Applied Soft Computing 63, 59–70.

[154] Samie, M. E., Hamzeh, A., 2017. Community detection in dynamic social networks: A local evolutionary approach. Journal of Information Science 43 (5), 615–634.

[155] Sánchez-Oro, J., Duarte, A., 2018. Iterated greedy algorithm for performing community detection in social networks. Future Generation Computer Systems 88, 785–791.

[156] Sanders, P., Schulz, C., 2012. Distributed evolutionary graph partitioning. In: 2012 Proceedings of the Fourteenth Workshop on Algorithm Engineering and Experiments (ALENEX). SIAM, pp. 16–29.

[157] Sarswat, A., Jami, V., Guddeti, R. M. R., 2017. A novel two-step approach for overlapping community detection in social networks. Social Network Analysis and Mining 7 (1), 47.

[158] Schiavo, S., Reyes, J., Fagiolo, G., 2010. International trade and financial integration: a weighted network analysis. Quantitative Finance 10 (4), 389–399.

[159] Shi, C., Cai, Y., Fu, D., Dong, Y., Wu, B., 2013. A link clustering based overlapping community detection algorithm. Data & Knowledge Engineering 87, 394–404.

[160] Shi, C., Liu, Y., Zhang, P., 2018. Weighted community detection and data clustering using message passing. Journal of Statistical Mechanics: Theory and Experiment 2018 (3), 033405.

[161] Shi, C., Wang, Y., Wu, B., Zhong, C., 2009. A new genetic algorithm for community detection. In: International Conference on Complex Sciences. Springer, pp. 1298–1309.

[162] Song, A., Li, M., Ding, X., Cao, W., Pu, K., 2016. Community detection using discrete bat algorithm. IAENG International Journal of Computer Science 43 (1), 37–43.

[163] Su, J., Havens, T. C., 2014. Quadratic program-based modularity maximization for fuzzy community detection in social networks. IEEE Transactions on Fuzzy Systems 23 (5), 1356–1371.

[164] Sun, H., Jie, W., Loo, J., Wang, L., Ma, S., Han, G., Wang, Z., Xing, W., 2018. A parallel self-organizing overlapping community detection algorithm based on swarm intelligence for large scale complex networks. Future Generation Computer Systems 89, 265–285.

[165] Sun, H., Jie, W., Sauer, C., Ma, S., Han, G., Wang, Z., Xing, K., 2017. A parallel self-organizing community detection algorithm based on swarm intelligence for large scale complex networks. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). Vol. 1. IEEE, pp. 806–815.

[166] Tackx, R., Tarissan, F., Guillaume, J.-L., 2017. Comsim: A bipartite community detection algorithm using cycle and nodes similarity. In: International Conference on Complex Networks and their Applications. Springer, pp. 278–289.

[167] Tahmasebi, S., Moradi, P., Ghodsi, S., Abdollahpouri, A., 2019. An ideal point based many-objective optimization for community detection of complex networks. Information Sciences.

[168] Tasgin, M., Bingol, A., 2006. Communities detection in complex networks using genetic algorithms. In: Proc. of the European Conference on Complex Systems (ECSS06).

[169] Traag, V. A., Aldecoa, R., Delvenne, J.-C., 2015. Detecting communities using asymptotical surprise. Physical Review E 92 (2), 022816.

[170] Traag, V. A., Van Dooren, P., Nesterov, Y., 2011. Narrow scope for resolution-limit-free community detection. Physical Review E 84 (1), 016114.

[171] Traag, V. A., Waltman, L., van Eck, N. J., 2019. From louvain to leiden: guaranteeing well-connected communities. Scientific reports 9.

[172] Villar-Rodriguez, E., Del Ser, J., Gil-Lopez, S., Bilbao, M. N., Salcedo-Sanz, S., 2017. A meta-heuristic learning approach for the non-intrusive detection of impersonation attacks in social networks. International Journal of Bio-Inspired Computation 10 (2), 109–118.

[173] Vinh, N. X., Epps, J., Bailey, J., 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research 11 (Oct), 2837–2854.

[174] Wan, X., Zuo, X., Song, F., 2018. A decomposition based multiobjective evolutionary algorithm for dynamic overlapping community detection. In: International Conference on Bio-Inspired Computing: Theories and Applications. Springer, pp. 338–350.

[175] Wang, X., Tang, L., 2009. A population-based variable neighborhood search for the single machine total weighted tardiness problem. Computers & Operations Research 36 (6), 2105–2110.

[176] Wang, Z., Zhao, X., Wen, P., Xue, J., Hu, C., 2016. Community detection in complex networks using improved artificial bee colony algorithm. In: International Conference on Communications, Information Management and Network Security. Atlantis Press.

[177] Watts, D. J., Strogatz, S. H., 1998. Collective dynamics of small-worldnetworks. nature 393 (6684), 440.

[178] Wen, X., Chen, W.-N., Lin, Y., Gu, T., Zhang, H., Li, Y., Yin, Y., Zhang, J., 2017. A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. IEEE Transactions on Evolutionary Computation 21 (3), 363–377.

[179] Westlake, B. G., Bouchard, M., 2016. Liking and hyperlinking: Community detection in online child sexual exploitation networks. Social science research 59, 23–36.

[180] Wilf, H. S., 2005. generatingfunctionology. AK Peters/CRC Press.

[181] Wolpert, D. H., Macready, W. G., et al., 1997. No free lunch theorems for optimization. IEEE transactions on evolutionary computation 1 (1), 67–82.

[182] Wu, P., Pan, L., 2015. Multi-objective community detection based on memetic algorithm. PloS one 10 (5), e0126845.

[183] Yang, L., Silva, J., Papageorgiou, L., Tsoka, S., 2016. Community structure detection for directed networks through modularity optimisation. Algorithms 9 (4), 73.

[184] Yang, X.-S., 2010. Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-Inspired Computation 2 (2), 78–84.

[185] Yang, X.-S., 2010. A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp. 65–74.

[186] Yang, X.-S., Deb, S., 2009. Cuckoo search via lévy flights. In: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on. IEEE, pp. 210–214.

[187] Yang, X.-S., He, X., 2013. Bat algorithm: literature review and applications. International Journal of Bio-Inspired Computation 5 (3), 141–149.

[188] Yang, X.-S., He, X., 2013. Firefly algorithm: recent advances and applications. International Journal of Swarm Intelligence 1 (1), 36–50.

[189] Yang, Z., Algesheimer, R., Tessone, C. J., 2016. A comparative analysis of community detection algorithms on artificial networks. Scientific Reports 6, 30750.

[190] Yip, P. P., Pao, Y.-H., 1995. Combinatorial optimization with use of guided evolutionary simulated annealing. IEEE Transactions on neural networks 6 (2), 290–295.

[191] Yuxin, Z., Shenghong, L., Feng, J., 2017. Overlapping community detection in complex networks using multi-objective evolutionary algorithm. Computational and Applied Mathematics 36 (1), 749–768.

[192] Zadeh, P. M., Kobti, Z., 2015. A multi-population cultural algorithm for community detection in social networks. Procedia Computer Science 52, 342–349.

[193] Žalik, K. R., Žalik, B., 2017. Multi-objective evolutionary algorithm using problem-specific genetic operators for community detection in networks. Neural Computing and Applications, 1–14.

[194] Žalik, K. R., Žalik, B., 2018. Memetic algorithm using node entropy and partition entropy for community detection in networks. Information Sciences 445, 38–49.

[195] Zhang, L., Pan, H., Su, Y., Zhang, X., Niu, Y., 2017. A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection. IEEE Transactions on Cybernetics 47 (9), 2703–2716.

[196] Zhang, P., 2015. Evaluating accuracy of community detection using the relative normalized mutual information. Journal of Statistical Mechanics: Theory and Experiment 2015 (11), P11006.

[197] Zhang, P., Moore, C., 2014. Scalable detection of statistically significant communities and hierarchies, using message passing for modularity. Proceedings of the National Academy of Sciences 111 (51), 18144–18149.

[198] Zhang, Q., Ge, M., Fu, J., 2018. Pso combined with lpa for the detection of overlapping community. In: 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, pp. 2825–2830.

[199] Zhao, Y., 2017. A survey on theoretical advances of community detection in networks. Wiley Interdisciplinary Reviews: Computational Statistics 9 (5), e1403.

[200] Zhao, Y., Zou, F., Chen, D., 2019. A discrete sine cosine algorithm for community detection. In: International Conference on Intelligent Computing. Springer, pp. 35–44.

[201] Zhong, Y., Lin, J., Wang, L., Zhang, H., 2018. Discrete comprehensive learning particle swarm optimization algorithm with metropolis acceptance criterion for traveling salesman problem. Swarm and Evolutionary Computation.

[202] Zhou, X., Liu, Y., Li, B., 2016. A multi-objective discrete cuckoo search algorithm with local search for community detection in complex networks. Modern Physics Letters B 30 (07), 1650080.

[203] Zhou, X., Liu, Y., Li, B., Li, H., 2017. A multiobjective discrete cuckoo search algorithm for community detection in dynamic networks. Soft Computing 21 (22), 6641–6652.

[204] Zhou, X., Liu, Y., Li, B., Sun, G., 2015. Multiobjective biogeography based optimization algorithm with decomposition for community detection in dynamic networks. Physica A: Statistical Mechanics and its Applications 436, 430–442.

[205] Zhou, X., Liu, Y., Zhang, J., Liu, T., Zhang, D., 2015. An ant colony based algorithm for overlapping community detection in complex networks. Physica A: Statistical Mechanics and its Applications 427, 289–301.

[206] Zhou, X., Yang, K., Xie, Y., Yang, C., Huang, T., 2019. A novel modularity-based discrete state transition algorithm for community detection in networks. Neurocomputing.

[207] Zhou, X., Zhao, X., Liu, Y., 2018. A multiobjective discrete bat algorithm for community detection in dynamic networks. Applied Intelligence 48 (9), 3081–3093.

[208] Zhou, Y., Wang, J., Luo, N., Zhang, Z., 2016. Multiobjective local search for community detection in networks. Soft Computing 20 (8), 3273–3282.