Combining Bio-inspired Meta-Heuristics and Novelty Search for Community Detection over Evolving Graph Streams

Eneko Osaba
Tecnalia Research & Innovation,
48160, Derio, Spain
eneko.osaba@tecnalia.com

Javier Del Ser
University of the Basque Country
(UPV/EHU), 48013 Bilbao, Spain
javier.delser@tecnalia.com

Angel Panizo
Universidad Autonoma de Madrid,
Madrid, Spain
angel.panizo@uam.es

David Camacho
Universidad Autonoma de Madrid,
Madrid, Spain
david.camacho@uam.es

Akemi Galvez
Toho University, Funabashi, Japan
University of Cantabria, Spain
galveza@unican.es

Andres Iglesias
Toho University, Funabashi, Japan
University of Cantabria, Spain
iglesias@unican.es

ABSTRACT

Finding communities of interrelated nodes is a learning task that often holds in problems that can be modeled as a graph. In any case, detecting an optimal partition in a graph is highly time-consuming and complex. For this reason, the implementation of search-based meta-heuristics arises as an alternative for addressing these problems. This manuscript focuses on optimally partitioning dynamic network instances, in which the connections between vertices change dynamically along time. Specifically, the application of Novelty Search mechanism for solving the problem of finding communities in dynamic networks is studied in this paper. For this goal, this procedure has been embedded in the search process undertaken by three different bio-inspired meta-heuristic schemes: Bat Algorithm, Firefly Algorithm and Particle Swarm Optimization. All these methods have been properly adapted for dealing with this discrete and dynamic problem, using a reformulated expression of the modularity coefficient as its fitness function. A thorough experimentation has been conducted using a benchmark composed by 12 synthetically created instances, with the main objective of analyzing the performance of the proposed Novelty Search mechanism when facing this problem. In light of the outperforming behavior of our approach and its relevance dictated by two different statistical tests, we conclude that Novelty Search is a promising procedure for finding communities in evolving graph data.

CCS CONCEPTS

• **Theory of computation → Bio-inspired optimization**; *Random search heuristics*; *Theory of randomized search heuristics*; • **Mathematics of computing** → *Evolutionary algorithms*;

KEYWORDS

Bio-inspired computation, Novelty Search, Evolving Graphic Streams, Community Detection

ACM Reference Format:
Eneko Osaba, Javier Del Ser, Angel Panizo, David Camacho, Akemi Galvez, and Andres Iglesias. 2019. Combining Bio-inspired Meta-Heuristics and Novelty Search for Community Detection over Evolving Graph Streams. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3319619.3326831

1 INTRODUCTION

These days, a remarkable amount of methods and tools can be found in the literature for efficiently excerpting insights from the heterogeneous interrelations between different elements in a network. Undoubtedly, the impactful influence that social networks have in the current society is the main reason of this recently exploded boiling activity [3]. The information that can be obtained by the use of these methods is truly diverse, ranging from enriched ways for network visualization and routing of efficient paths amidst a pair of nodes, to the assessment of the influence of a specific node in the whole network (*centrality*). All this gained insights have been used for many practical goals recently, such as child abuse detection [53] or the evaluation of radicalization risk [2, 38].

Focused in the above mentioned Social Networks, the finding of communities within a network is probably one of the most valuable and recurrent tasks, as evinced by the recent literature [21, 50]. In this regard, a *community* is group of nodes which fulfill the principles of weak inter-connectivity (a low connectivity with members belonging to other communities) and robust intra-connectivity (solid links between vertices of the same partition). Moreover, the characteristics of these parameters lead to the construction of heterogeneous network (directed, weighted or dynamic, among other variants), quantifying the cohesiveness of any projected partition. Furthermore, different metrics have been formulated by the scientific community for efficiently evaluating the quality of any proposed solution. Each of these metrics takes different assumptions, leading to a single value for assessing the quality of the community. Some of the most used examples are Surprise [1], Permanence [6] and, above all, Newman and Girvan's Modularity [32]. In the present research, the latter Modularity is considered.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326831

In this context, the main focus of this work is set on a specific type of graphs characterized by their time-evolving dynamic nature. Usually, relationships between human beings undergo changes over time. People tend to strengthen existing connections or build new ones throughout their lives, while some others are loosened or eventually broken up. Thus, if we place our attention on the relational history of a single person in a long time lapse, we will surely discover some level of dynamism. Logically, this phenomenon can be also reflected in Social Networks. In this way, dynamic graphs are special instances of networks in which the number of nodes and the links between them can suffer from modifications along time, reliably modeling the situation described above. This is the reason why a dynamic network can be also considered as an evolving graph stream, in which the evolution of the graph may occur among consecutive time steps. In this envisaged situation, the discovery of communities becomes even more involved than in the static scenario, as structural graph dynamics propagate to the communities underneath the graph itself. This noted fact unleashes a pressing need for devising efficient methods to incrementally infer communities from streaming graph instances, while incorporating mechanisms to adapt the inferred community structure to structural changes eventually occurring over time.

In this regard, a myriad of efficient methods has been proposed in recent years for solving the problem of detecting communities optimizing one of the previously pointed metrics. In line with the study presented in this paper, a rising part of the scientific community is devoting efforts towards adapting heuristic optimization methods to this problem by using one modularity metric as their objective function. Many interesting studies can be found in the recent literature, devoted to diverse approaches of algorithmic methods, graph types and quality metrics. One of the most utilized method for this purpose is the Genetic Algorithm, as can be seen in works such as [58] or [40]. Some additional techniques used in this research area that fall inside the umbrella of Evolutionary Computation and Swarm Intelligence are Ant Colony Optimization [39], Particle Swarm Optimization [43], Artificial Bee Colony [48], Bat Algorithm [18] and Firefly Algorithm [10]. Regarding dynamic networks, the amount of scientific studied related to this topic is much fewer than the one associated to stationary networks. A valuable survey focused on community finding in dynamic graphs can be found in [45]. Another interesting practical study was reported in [30], in which a multi-objective Bat Algorithm is proposed for tackling with this problem. Furthermore, Genetic Algorithms have been also used in this context, as evinced in [26] or [15]. Finally, along the history, many works have been published exploring the tackling of dynamic problems using metaheuristic methods [8, 9, 29].

In this paper we take a step further over the state of the art by elaborating on a new research direction: the application of Novelty Search mechanism for solving the problem of finding communities in dynamic networks. The Novelty Search (NS, [24]) was proposed in 2008 as a way to enhance the exploratory ability of population-based algorithmic solvers. After showing its great performance applied to several optimization problems, we hypothesize its promising performance also for this specific community finding problems. To this end, we have developed different versions of well-known Swarm Intelligence methods, namely Bat Algorithm (BA, [55]), Firefly Algorithm (FA, [54]) and Particle Swarm Optimization (PSO, [22]), and we evaluate in this manuscript the performance of these meta-heuristics embedding the NS mechanism on their basic scheme. We introduce along the paper the descriptions on how these methods have been modeled for tackling the problem at hand, and how the NS has been adapted for this discrete scenario. In order to assess the performance of each implemented solver, outcomes get over 12 synthetically generated datasets are compared and discussed, based on their capability to discover their ground-of-truth community partition. Moreover, with the aim to draw statistically robust conclusions, two different statistical tests (Friedman's and Holm's) have been carried out with the obtained outcomes.

The remaining of this manuscript is structured as follows: in Section 2, the problem of community finding in dynamic network is formulated. In Section 3, the main concepts behind NS are introduced, placing emphasis on how we have hybridized meta-heuristic solvers with this mechanism. Next, the considered heuristic solvers and important development aspects are described in Section 4. Experimentation is detailed and discussed in Section 5 and, finally, Section 6 ends this paper with concluding remarks and a prospect of future research lines.

2 PROBLEM STATEMENT

We begin by modeling the network as a graph $\mathcal{G} \doteq \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ stands for the set of $|\mathcal{V}| = V$ nodes or vertex of the network, and $\mathcal{E}$ represents the group that details the dynamic situation of the edges (or links connecting every pair of elements) over a given time frame (*time horizon*). During this time frame a set of graph snapshots is generated, each one describing the specific structure of the network at one moment in time. Thus, $\mathcal{E} \doteq \{e_1, \ldots, e_N\}$, where e_n corresponds to the group of edges at time stamp n.

It is important to highlight that the dynamism of the graphs considered on this study affects to the relations between the vertex. This means that while the connections between elements vary along the time, the number of nodes remains the same in the whole time horizon. Additionally, the weight of each edge connecting nodes v and v' is $w_{v,v'} = 1$. We also consider that $w_{v,v} = 0$ (i.e. no self-loops) and that $w_{v,v'} = 0$ whether elements v and v' are not connected. We also define an adjacency matrix $\mathbf{W}$ given by $\mathbf{W} \doteq \{w_{v,v'} : v, v' \in \mathcal{V}\}$, and satisfying $\mathrm{Tr}(\mathbf{W}) = 0$. Finally, we assume symmetry in $\mathcal{G}$, so, $w_{v,v'} = w_{v',v}$. Henceforth, weights are denoted as $w^n_{v,v'}$, representing the weight at timestamp n. This notation is used in order to contemplate the dynamism of the problem.

Therefore, the problem of detecting related communities in the network $\mathcal{G}$ is assumed in this research as the partition of the nodes group $\mathcal{V}$ into a number of non-empty, non-fixed size and disjoint sets. Considering that M is the amount of partitions $\widetilde{\mathcal{V}} \doteq \{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$, 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). Similarly, the set of partitions will be depicted as $\mathcal{V}_n$ henceforth, representing $\widetilde{\mathcal{V}_n} \doteq \{\mathcal{V}^n_1, \ldots, \mathcal{V}^n_M\}$ the set of communities detected at time stamp n.

As has been advanced in the introduction, the Newman and Girvan's Modularity metric has been employed for properly measuring the quality of a certain community. This famous formula has been widely employed in a plethora of previously published works, and its adequacy has been extensively affirmed in studies such as [5, 7, 25]. Thusly, the modularity metric for a considered community can be calculated by:

$$Q(\widetilde{\mathcal{V}_n}) \doteq \frac{1}{2|E_n|}\sum_{v,v'}\left[w^n_{v,v'} - \frac{k^n_v k^n_{v'}}{2|E_n|}\right]\delta(v,v')_n, \tag{1}$$

where k^n_v is the degree of node v, $|E_n|$ is the total amount of elements in the network, and $\delta(v,v')_n$ depicts the Kronecker delta symbol. All these values are contextualized in the time stamp n. With other words, $\delta(v,v')_n$ is a binary function $\delta : \mathcal{V}_n \times \mathcal{V}_n \mapsto \{0,1\}$, such that $\delta(v,v'_n) = 1$ if $\mathcal{V}^v_n = \mathcal{V}^{v'}_n$ as per the community set by $\widetilde{\mathcal{V}_n}$ (and 0 otherwise). We note the dependence of all this notation with the time stamp n. Thus, finding an optimal partition $\widetilde{\mathcal{V}}^*_n$ of graph $\mathcal{G}_n$ can be formulated as:

$$\widetilde{\mathcal{V}}^*_n = \arg\max_{\widetilde{\mathcal{V}_n}\in\mathcal{B}_V} Q(\widetilde{\mathcal{V}_n}), \tag{2}$$

where $\mathcal{B}_V$ denotes the set of possible partitions of $\mathcal{V}_n$ nodes into nonempty subsets (i.e. the solution space of the above combinatorial problem). The cardinality of this group is computationally intractable for an exhaustive search, whose value is given the V_n-th Bell number [17]. It is also important to remark that our main objective is to solve the above problem over the n timestamps. For reaching this objective, different degrees of similarity must be modeled and contemplated between consecutive snapshots n and $n+1$.

3 NOVELTY SEARCH (NS)

NS is a mechanism which main goal is to increase the diversity of a population-based method, by finding novel candidates in the behavioral space instead of the search space. Commonly, individual in a population tend to collapse in the same point of the solution space. On the other hand, this tendency does not appear in the behavioral space, which is measured using the well-known Euclidean distance. Thus, we can quantify numerically how novel an individual $\mathbf{x}$ is within the population/swarm at hand as:

$$\rho(\mathbf{c}) = \frac{1}{k}\sum_{i=1}^{k} d(\mathbf{c}, \boldsymbol{\mu}_i), \tag{3}$$

where $d(\cdot,\cdot)$ denotes the Euclidean distance, and k is the number of neighbor candidates selected from the subset of neighbors $\mathcal{N} = \{\mu_1, \mu_2, \ldots, \mu_k\} \subseteq \mathcal{P}$ (i.e. the neighborhood size). The value of k is problem-dependent, i.e. its specific value should be selected empirically. Furthermore, the selection of candidates is driven by the distance metric, which also depends on the problem. Although NS has showcased its efficiency in manifold studies so far [14, 16, 27], the strategy to adapt NS to a given problem remains weakly defined, and strongly subject to the problem at hand [13].

4 PROPOSED HYBRID APPROACH

For efficiently dealing with the dynamic community finding problem formulated previously, we propose to hybridize several bio-inspired meta-heuristic solvers and a NS-based diversity inducing mechanism. Prior to detailing each solver under consideration, we describe several important design aspects in what follows. These aspects are related to solution encoding, repairing mechanism and the metrics employed for comparing among different candidates.

To begin with, a *label-based* representation [20] has been adopted for encoding solutions that represent partitions arranged over a graph. Thus, each candidate is represented as a combination $\mathbf{c} = [c_1, c_2, \ldots, c_V]$ of V integers from the range $[1, \ldots, V]$, where $V = |\mathcal{V}|$ is the amount of vertices of the network. Additionally, c_v represents the community label to which element v belongs. For example, considering a network comprising $V = 12$ nodes, a feasible solutions could be $\mathbf{c} = [1,1,1,2,2,3,1,3,2,3,2,3]$, meaning that the partition obtained is $\widetilde{\mathcal{V}} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$, where $\mathcal{V}_1 = \{1,2,3,7\}$, $\mathcal{V}_2 = \{4,5,9,11\}$ and $\mathcal{V}_3 = \{6,8,10,12\}$ (thus, $M = 3$).

This representation has the disadvantage of an inherent ambiguity between the selected genotype representation (label encoding) and the phenotype. Specifically, multiple genotypes map to the same phenotype (partition). For avoiding this important problem, a repairing mechanism has been implemented, partly inspired by the one presented by Falkenauer in [12]. This procedure transforms the genotype of each generated candidate to collapse into a single representation, thereby making the relationship between the repaired genotype to its phenotype be biyective (one-to-one). For example, unrepaired solutions such as $\mathbf{c} = [4,4,4,3,3,5,4,5,3,5,3,5]$ and $\mathbf{c}' = [7,7,7,1,1,8,7,8,1,8,1,8]$ (which represent the same partition) are modified to collapse into a single individual for both candidates: $\mathbf{c} = [1,1,1,2,2,3,1,3,2,3,2,3]$.

The next important aspect to revolve around is the measure of similarity between different individuals (solutions). This measure lies at the core of movement strategies of certain meta-heuristics as the ones considered in this work. For this matter we have selected the Hamming Distance following the good performance of this choice observed in prior contributions [34]. Specifically, the Hamming Distance is computed as the number of non-corresponding elements between two solutions. For instance, if we consider $\mathbf{c} = [1,1,\mathbf{1},2,\mathbf{2},2,2,3,2,\mathbf{1}]$ and $\mathbf{c}' = [1,1,\mathbf{2},2,\mathbf{3},2,2,3,2,\mathbf{2}]$ as exemplifying partitions, their Hamming Distance equals 3.

Finally, four movement functions have been implemented for evolving solutions along the search process. These functions have been labeled as CE_1, CE_3, CC_1 and CC_3. On the one hand, the subscript depicts the number of randomly selected vertices, which are removed from its corresponding community. On the other hand, in CE_* operators, the extracted nodes are re-inserted in an already existing community, whereas in CC_* they can be introduced also in newly generated ones. At this point, it is interesting to observe that the main operator of all meta-heuristic methods is CC_1; CE_1, CE_3, and CC_3, however, compose the pool of functions that NS considers for the reinsertion of candidates. This aspect is detailed below. We now introduce the meta-heuristic algorithms under consideration:

- **BA**: The Bat Algorithm was first proposed for solving continuous optimization problems [55]. As done in recent work [37], a discrete adaptation has been introduced to accommodate the BA operators to the combinatorial nature of the problem tackled in this study. First, each bat of the swarm represents a possible solution to the problem, and both concepts of pulse emissions r_i and loudness A_i have been modeled and implemented in the same way as in the naïve BA. In order to simplify the approach, no frequency parameter has been considered. Furthermore, the value of the velocity parameter v_i has been adapted by embracing the Hamming Distance as its similarity function as $v^t_p = \text{rand}[1, D_H(\mathbf{c}_p, \mathbf{c}^{best})]$, i.e., the velocity of the p-th bat in the swarm at generation t is a random number, which follows a discrete uniform distribution between 1 and the Hamming Distance between $\mathbf{c}_p$ and the leading bat $\mathbf{c}^{best}$. At generation t, $\mathbf{c}_p$ moves towards $\mathbf{c}^{best}$ as:

$$\mathbf{c}_p(t+1) = \Psi\left(\mathbf{c}_p(t), \min\left\{V, v_p^t\right\}\right), \tag{4}$$

where $\Psi(\mathbf{c}, Z) \in \{CE_1, CE_3, CC_1, CC_3\}$ is the movement function that depends on the bat in move, each one parametrized by the amount of times Z this function is performed onto $\mathbf{c}$. After Z trials, the movement enhancing most the fitness of the modified individual is chosen as output.

- **FA**: similarly to what occurs with the BA meta-heuristic algorithm, the basic FA cannot be directly applied to the combinatorial problem considered in this work, hence some adaptations are needed. Each firefly represents a feasible solution for the problem. Regarding the light absorption parameter, it is considered in this discrete FA, taking into account its importance for properly adjusting the fireflies' attractiveness. In addition, the distance between two individuals is also computed by the Hamming Distance. Finally, the movement criterion followed by a firefly attracted by a brighter one is determined as per (4).
- **PSO**: Particle Swarm Optimization is the last method considered in our study. In this case, PSO has been already adapted to discrete problems in multiple occasions [52, 59]. Following in part this prior work, each particle in the swarm represents a possible partition for the problem. Velocity v_p is computed roughly in the same fashion as for the BA. Additionally, the movement strategy is inspired by Expression (4). Finally, Hamming Distance has been taken as the similarity measure between particles.

In this specific study, the NS mechanism has been applied identically in the three considered meta-heuristic algorithms. Previously published works [14] pointed out the need for modeling a suitable distance metric in order to properly achieve the objective of enhancing the diversity of the population. In our case, this metric is the aforementioned Hamming Distance $D_H(\cdot, \cdot)$. Furthermore, a subset $\mathcal{B}$ is maintained, in which all replaced or discarded individuals are introduced at each generation. Thus, the size of $\mathcal{B}$ is the same as the main population of the algorithm.

Conceptually, $\mathcal{B}$ contains the candidates which are more potentially *novel* and, therefore, to be inserted back into the population for diversity injection. Basically, when a trial solution $\mathbf{c}_i$ outperforms the individual which is going to replace, it is introduced in the main population, whereas the replaced solution is inserted into $\mathcal{B}$. Otherwise, if the evolved candidate is worse than its preceding version, the former is inserted directly into $\mathcal{B}$. Furthermore, once the t-th generation comes to its end, if r_{NS} (a value drawn from a normal probability distribution) is lower than the parameter $NS_P \in [0.0, 1.0]$, the NS mechanism is carried out. In this specific study, $NS_P = 0.3$, which has been fixed after a thorough empirical analysis not shown for the sake of clarity.

Furthermore, there is no clear consensus established by the related community regarding the number of individuals that should be reinserted in the population throughout the *NS* procedure, and how they replace existing solutions. Once again, practitioners of the field recommend to adapt these values and criteria depending on the problem being tackled. In this study, the number of reinserted individuals has been established to 7, which replace solutions of the population with lower fitness. Furthermore, these candidates are chosen from $\mathcal{B}$ based on their distance respect to the whole population. Namely, the 7 solutions having a greater diversity with respect to the population/swarm of the meta-heuristic solver are those chosen for reinsertion. Finally, the main contribution of the proposed NS mechanism is a novel neighborhood changing procedure. Concretely, each time a solution $\mathbf{c}$ is introduced in $\mathcal{B}$, its movement operator $\Psi(\cdot, \cdot)$ is randomly modified using a baseline pool of four different functions $\{CE_1, CE_3, CC_1, CC_3\}$. In this way, if a solution is reinserted in the population, it can explore the solution space in a different way. This simple but effective mechanism not only enhances further the diversity of the population, but also the exploratory capacity of the solver.

Once described the designed methods and how the NS mechanism has been adapted to this specific problem, it is also crucial to explain here how implemented approaches manage the transitions between the n different snapshots that compose each dataset. First, in the beginning of every execution, the main population of the algorithm is comprised completely by randomly generated individuals. Furthermore, $\mathcal{B}$ is initialized empty. After that, the main population is maintained in every snapshot transition, but $\mathcal{B}$ is reinitialized between consecutive time stamps. In other words, the population at the first generation of snapshot n is the resultant of the last generation of snapshot $n - 1$.

5 EXPERIMENTATION AND RESULTS

We assess the performance of the considered solvers for dynamic graph partitioning by means of several computer experiments over a heterogeneous set of synthetically generated graph instances. All these datasets have been built by using the DANCer framework [4, 23], aiming at covering a diverse group of practical situations in dynamic environments. The complete benchmark is comprised by 12 different instances with 100 nodes, for which the true underlying community structure behind the generated time-varying graphs (*ground of truth*) is given. Furthermore, each dataset is labeled to indicate the parameter values established for its generation, namely:

- *Size of the problem*: This value is 100 nodes in all the datasets.
- *Communities*: Number of communities that comprise the ground of truth partition.
- *Generations*: Number of generations executed in each graph snapshot $\mathcal{G}_n$.
- *Variability*: The difference that the graph suffers between two adjacent graph snapshots $\mathcal{G}_n$ and $\mathcal{G}_{n+1}$. This parameter can adopt three values: slight (variety of 5% between adjacent snapshots), medium (variety of 10%), and severe (variety of 20%).
- *Transition*: Each dataset is comprised by 30 base snapshots, divided into two different families of 15 timestamps. If *Transition* equals *abrupt*, the transition between both families is carried out directly after the 15th timestamp. Thus, these datasets are composed just by these 30 canonical snapshots. On the hand, if *Transition* is *gradual*, the transition is performed gradually, introducing a group of 10 additional snapshots, which are placed between the last timestamp of the first family and the first timestamps of the second one. Thus, these *gradual* instances are composed by 40 snapshots.

This generation approach allows evaluating the performance of the implemented methods over *noisy* versions of a network characterized by a controlled underlying community distribution. This

Table 1: Obtained NMI results (average/best) using BA, BA_{NS}, FA, FA_{NS}, PSO, and PSO_{NS}. Best average results have been highlighted in bold.

	BA		*BA_{NS}*		*FA*		*FA_{NS}*		*PSO*		*PSO_{NS}*	
Instance	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best
100_7_20_*Sli_Abr*	0.673-0.474	0.782-0.656	**0.674-0.493**	0.766-0.665	0.645-0.538	0.779-0.673	**0.687-0.564**	0.780-0.719	0.568-0.428	0.654-0.539	**0.642-0.511**	0.768-0.705
100_7_20_*Sli_Grad*	0.676-0.473	0.787-0.663	**0.691-0.490**	0.773-0.650	0.633-0.530	0.726-0.683	**0.682-0.569**	0.785-0.712	0.566-0.459	0.676-0.588	**0.644-0.570**	0.773-0.681
100_7_50_*Sli_Abr*	0.666-**0.512**	0.747-0.683	**0.676**-0.493	0.758-0.646	0.654-0.561	0.754-0.693	**0.691-0.597**	0.779-0.696	0.626-0.477	0.752-0.619	**0.668-0.554**	0.735-0.656
100_7_50_*Sli_Grad*	0.688-0.455	0.765-0.579	**0.689-0.486**	0.769-0.633	0.660-0.571	0.755-0.710	**0.684-0.594**	0.788-0.746	0.618-0.481	0.736-0.645	**0.674-0.577**	0.761-0.664
100_8_20_*Med_Abr*	0.654-0.476	0.819-0.641	**0.663-0.496**	0.837-0.613	0.605-0.489	0.775-0.706	**0.675-0.535**	0.839-0.693	0.564-0.442	0.692-0.555	**0.653-0.467**	0.852-0.661
100_8_20_*Med_Grad*	**0.674-0.459**	0.835-0.613	0.663-0.456	0.855-0.638	0.626-0.483	0.818-0.623	**0.682-0.526**	0.845-0.659	0.568-0.454	0.677-0.552	**0.645-0.489**	0.850-0.631
100_8_50_*Med_Abr*	0.701-0.469	0.861-0.603	**0.710-0.503**	0.861-0.617	0.640-0.513	0.889-0.658	**0.690-0.533**	0.847-0.636	0.620-0.458	0.798-0.642	**0.698-0.501**	0.854-0.637
100_8_50_*Med_Grad*	0.671-0.467	0.870-0.631	**0.691-0.468**	0.841-0.616	0.655-0.502	0.817-0.637	**0.688-0.517**	0.833-0.661	0.635-0.471	0.834-0.653	**0.685-0.510**	0.837-0.664
100_9_20_*Sev_Abr*	0.649-0.511	0.820-0.705	**0.650-0.530**	0.797-0.713	0.612-0.515	0.841-0.657	**0.653-0.562**	0.796-0.694	0.539-0.485	0.641-0.609	**0.601-0.496**	0.754-0.674
100_9_20_*Sev_Grad*	**0.640-0.544**	0.769-0.716	0.639-0.525	0.814-0.652	0.607-0.526	0.789-0.656	**0.649-0.565**	0.806-0.700	0.541-0.515	0.649-0.606	**0.613-0.549**	0.768-0.674
100_9_50_*Sev_Abr*	0.644-**0.544**	0.775-0.705	**0.654**-0.539	0.820-0.650	0.631-0.526	0.770-0.634	**0.671-0.560**	0.829-0.716	0.594-0.505	0.716-0.631	**0.666-0.545**	0.872-0.655
100_9_50_*Sev_Grad*	0.655-0.524	0.786-0.640	**0.665-0.538**	0.825-0.662	0.647-0.520	0.775-0.631	**0.658-0.535**	0.822-0.660	0.595-0.535	0.706-0.676	**0.660-0.563**	0.836-0.705
Friedman's non-parametric test (mean ranking)												
Rank	3.0-4.5		1.83-3.75		4.83-3.25		**1.75-1.26**		6-5.54		3.58-2.66	

specific approach is opposed to the widely followed practice consisting of the comparison done based on the fitness value. For each dataset, 10 independent executions have been conducted, with main goal of drawing statistically reliable conclusions. The population size has been fixed to 50 individuals for every solver. For the development and parameterization of these methods, the guidelines given in [34–36] have been followed. The finishing criterion of each method is strictly related to the value of both the *Transition* and *Generations* parameters of every dataset. Depending on the value of these parameters, solvers are stopped after 150, 200, 300 or 400 generations.

In Table 1, outcomes (average/best) obtained by the four solvers are shown. Each of these values are divided into two different subvalues, each one depicting separately the performance for the first and the second family. As has been mentioned, each dataset is composed by 30 canonical graph snapshots (plus 10 transitional graphs for *gradual* instances), which belong to two different families. All results are shown in terms of the Normalized Mutual Information (NMI) with respect to the *ground of truth* partition of the specific timestamp. This means that the average depicted represents the mean NMI value for all the 15 timestamps belonging to the same family. Analogously, best values indicate the maximum value reached at any timestamp of the whole family. The NMI score measures the level of agreement between two community partitions: if $\text{NMI}(\widetilde{\mathcal{V}}, \widetilde{\mathcal{V}}') = 1$ both distributions $\widetilde{\mathcal{V}}$ and $\widetilde{\mathcal{V}}'$ are equal to each other. This also means that lower values denote that there are differences between the partition output by the solver and the true community underneath.

A first analysis reveals that in both PSO and FA, NS improves considerably the outcomes of the solver, for both first and second families in all the 12 datasets. On the other hand, in the case of BA, the NS mechanism yields better performance results in 83.33% of the first families (10 out of 12), and in the 66.66% of the second families (8 out of 12). The same trend is also noted for the best results obtained for each dataset. Other interesting insights can be drawn after conducting a Friedman's non-parametric test for multiple comparison [11, 33]. The last row of Table 1 indicates the mean ranking returned by this nonparametric test for each of the compared algorithms and families (the lower the rank, the better the performance). Results obtained from this test confirm that methods using the NS mechanism not only outperform their naïve version, but also emerge as the outperforming methods. Thus, FA_NS and BA_NS arise as the best alternatives for the first family, whereas FA_NS and PSO_NS take the lead for the second family.

Additionally, the Friedman statistic for the first family is equal to 48.428. Furthermore, the confidence interval has been established in 99%, being 16.812 the critical point in a χ^2 distribution with 6 degrees of freedom. Since $48.428 > 16.812$, it can be concluded that there are significant differences among the results. In relation to the second family, the Friedman statistic test is 37.25. Again, since $37.25 > 16.812$, the same conclusion is also applicable in this case. A second statistical test has been conducted for delving in the obtained outcomes. In this case, the analysis carried out is the Holm's post-hoc test. For this analysis, FA_{NS} has been established as the control algorithm in both families. Table 2 shows the unadjusted and adjusted p-values obtained with the application of this statistical test. In this case, it is noteworthy to highlight that FA_{NS} not only outperforms its basic version in a significant way for the first family (since p value is lower than 0.05), but also *PSO* and PSO_{NS} . For the second group the conclusions are more conclusive, since FA_{NS} significantly beats all its counterparts, except PSO_{NS}.

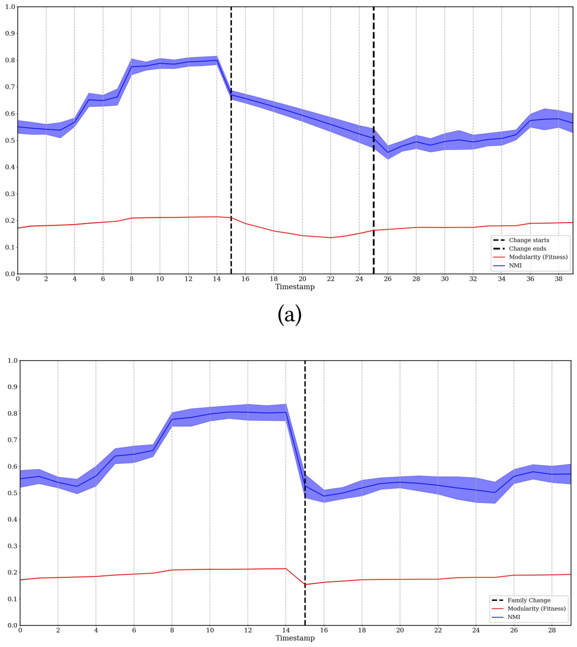
Table 2: Holm's post-hoc unadjusted and adjusted p-values using FA_{NS} as the control algorithm.

Fam1 (FA_{NS} as control)			Fam2 (FA_{NS} as control)		
Algorithm	Unadjusted p	Adjusted p	Algorithm	Unadjusted p	Adjusted p
BA	0.101707	0.203414	*BA*	0.000027	0.000106
BA_{NS}	0.913116	0.913116	BA_{NS}	0.001288	0.003803
FA	0.000054	0.000217	*FA*	0.010346	0.020691
PSO	0.0	0.0	*PSO*	0.0	0.0
PSO_{NS}	0.016377	0.049132	PSO_{NS}	0.071817	0.071814

Lastly, for a better visual assessment of the obtained outcomes, Figure 1.a and 1.b depict the evolution of the NMI score and modularity for instances 100_8_50_*Med_Grad* and 100_8_50_*Med_Abr*. Both plots illustrate the performance statistics averaged over the 10 independent runs, where it is important to observe that both types of changes in the community structure of the graph streams imply both a NMI and fitness decrease. This underscores the need for further research aimed at crosschecking how the severity and dynamics of the change relate to the amount of diversity induced by the NS mechanism.

6 CONCLUSIONS AND FUTURE RESEARCH

This research work has elaborated on the application of Novelty Search for finding communities in dynamic graph streams. For this goal, this diversity inducing mechanism has been embedded into the search algorithm of three different bio-inspired meta-heuristic solvers: Bat Algorithm, Firefly Algorithm and Particle Swarm Optimization. The discovery of partitions has been modeled as a combinatorial optimization problem, embracing the Newman and Girvan's modularity coefficient as the fitness function to be maximized. The performance of all implemented approaches has been evaluated by using a benchmark of 12 synthetic dynamic graph streams, suitably generated through the DANCer framework. A comparison between solvers has been conducted using the Normalized Mutual Information (or NMI) regarding their *ground of truth* partition. The results rendered by our experimentation permit to claim that the use of NS improves the performance of basic versions of the considered meta-heuristic algorithms. On a closing note, we conclude that the NS mechanism is a promising method for solving the problem of community detection over evolving graphs with search meta-heuristics.



(a)

(b)

Figure 1: Evolution of NMI (blue line) and modularity (red line) obtained by FA_{NS} for 100_8_50_*Med_Grad* (upper) and 100_8_50_*Med_Abr* (lower). A vertical gray line represents the time stamp at which the community structure begins to change, whereas a black vertical line denotes the point at which the community structure has changed entirely.

We plan to invest research efforts in several interesting directions rooted on this initial study. In the short term, we will adaptat additional Evolutionary Computation and Swarm Intelligence algorithms for tackling this problem, comparing the benefits of hybridizing them with a NS mechanism. Other modern solvers such as Cuckoo Search [56], Coral Reefs Optimization [46] or [31] will be considered, after showing a significant performance applied to other optimization problems [19, 41, 42, 47, 57]. In the longer term, we will explore how to modify the NS mechanism proposed in this work to graph instances composed by a massive number of nodes, for which avant-garde techniques under the family of Large-Scale Global Optimization will be investigated. Finally, in the case of recurrent changes over the stream we foresee that the amount of population diversity imprinted by NS during the meta-heuristic search should be linked to the characteristics of changes (speed and intensity) along the stream. If these characteristics could be predicted as a result of the recurrent nature of the phenomena producing it, we could actively tune the diversity induced for an adapted reaction of the search under such circumstances. Our attention will be surely focused on exploring this postulated hypothesis in the future.

ACKNOWLEDGEMENTS

E. Osaba and J. Del Ser acknowledge the financial support from the EMAITEK funds from the Basque Government. A. Iglesias and A. Galvez receive financial support from projects TIN2017-89275-R (AEI/FEDER, UE) and PDE-GIR (H2020, MSCA program, ref. 778035). A. Panizo and D. Camacho thank the Spanish Ministry of Science and Education and Competitivity (MINECO), the European Regional Development Fund (FEDER) and the Comunidad Autonoma de Madrid for their funding support through grants TIN2017-85727-C4-3-P (DeepBio) and P2018/TCS-4566 (CYNAMON).

REFERENCES

[1] Rodrigo Aldecoa and Ignacio Marín. 2011. Deciphering network community structure by surprise. *PloS one* 6, 9 (2011), e24195.

[2] Gema Bello-Orgaz, Julio Hernandez-Castro, and David Camacho. 2017. Detecting discussion communities on vaccination in twitter. *Future Generation Computer Systems* 66 (2017), 125–136.

[3] Gema Bello-Orgaz, Jason J Jung, and David Camacho. 2016. Social big data: Recent achievements and new challenges. *Information Fusion* 28 (2016), 45–59.

[4] Oualid Benyahia, Christine Largeron, Baptiste Jeudy, and Osmar R Zaïane. 2016. Dancer: Dynamic attributed network with community structure generator. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 41–44.

[5] Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. 2017. Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)* 50, 4 (2017), 54.

[6] Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. 2014. On the permanence of vertices in network communities. In *ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1396–1405.

[7] Mingming Chen, Konstantin Kuzmin, and Boleslaw K Szymanski. 2014. Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems* 1, 1 (2014), 46–65.

[8] Carlos Cruz, Juan R González, and David A Pelta. 2011. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing* 15, 7 (2011), 1427–1448.

[9] Ignacio G Del Amo, David A Pelta, Juan R González, and Antonio D Masegosa. 2012. An algorithm comparison for dynamic optimization problems. *Applied Soft Computing* 12, 10 (2012), 3176–3192.

[10] Javier Del Ser, Jesus L Lobo, Esther Villar-Rodriguez, Miren Nekane Bilbao, and Cristina Perfecto. 2016. Community detection in graphs based on surprise maximization using firefly heuristics. In *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2233–2239.

[11] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. 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 (2011), 3–18.

[12] Emanuel Falkenauer. 1998. *Genetic algorithms and grouping problems.* Wiley & Sons., Inc., New York, NY, USA.

[13] Iztok Fister, Andres Iglesias, Akemi Galvez, Javier Del Ser, and Eneko Osaba. 2018. Using Novelty Search in Differential Evolution. In *International Conference on Practical Applications of Agents and Multi-Agent Systems.* Springer, 534–542.

[14] Iztok Fister, Andres Iglesias, Akemi Galvez, Javier Del Ser, Eneko Osaba, Iztok Fister Jr, Matjaž Perc, and Mitja Slavinec. 2019. Novelty search for global optimization. *Appl. Math. Comput.* 347 (2019), 865–881.

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

[16] Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. 2015. Devising effective novelty search algorithms: A comprehensive empirical study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation.* ACM, 943–950.

[17] John Michael Harris, Jeffry L Hirst, and Michael J Mossinghoff. 2008. *Combinatorics and graph theory.* Vol. 2. Springer.

[18] Eslam A Hassan, Ahmed Ibrahem Hafez, Aboul Ella Hassanien, and Aly A Fahmy. 2015. A discrete bat algorithm for the community detection problem. In *International Conference on Hybrid Artificial Intelligence Systems.* Springer, 188–199.

[19] Xing-Shi He, Fan Wang, Yan Wang, and Xin-She Yang. 2018. Global Convergence Analysis of Cuckoo Search Using Markov Theory. In *Nature-Inspired Algorithms and Applied Optimization.* Springer, 53–67.

[20] Eduardo Raul Hruschka, Ricardo JGB Campello, Alex A Freitas, et al. 2009. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 39, 2 (2009), 133–155.

[21] Baofang Hu, Hong Wang, Xiaomei Yu, Weihua Yuan, and Tianwen He. 2019. Sparse network embedding for community detection and sign prediction in signed social networks. *Journal of Ambient Intelligence and Humanized Computing* 10, 1 (2019), 175–186.

[22] James Kennedy. 2011. Particle swarm optimization. In *Encyclopedia of machine learning.* Springer, 760–766.

[23] Christine Largeron, Pierre-Nicolas Mougel, Oualid Benyahia, and Osmar R Zaïane. 2017. DANCer: dynamic attributed networks with community structure generation. *Knowledge and Information Systems* 53, 1 (2017), 109–151.

[24] Joel Lehman and Kenneth O Stanley. 2008. Exploiting open-endedness to solve problems through the search for novelty.. In *ALIFE.* 329–336.

[25] Elizabeth A Leicht and Mark EJ Newman. 2008. Community structure in directed networks. *Physical review letters* 100, 11 (2008), 118703.

[26] Zhangtao Li and Jing Liu. 2016. A multi-agent genetic algorithm for community detection in complex networks. *Physica A: Statistical Mechanics and its Applications* 449 (2016), 336–347.

[27] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. 2015. Constrained novelty search: A study on game content generation. *Evolutionary computation* 23, 1 (2015), 101–129.

[28] Hao Lu, Mahantesh Halappanavar, and Ananth Kalyanaraman. 2015. Parallel heuristics for scalable community detection. *Parallel Comput.* 47 (2015), 19–37.

[29] Antonio D Masegosa, Enrique Onieva, Pedro Lopez-Garcia, Eneko Osaba, and Asier Perallos. 2015. An adaptive local search with prioritized tracking for Dynamic Environments. *International Journal of Computational Intelligence Systems* 8, 6 (2015), 1053–1075.

[30] Imane Messaoudi and Nadjet Kamel. 2019. A multi-objective bat algorithm for community detection on dynamic social networks. *Applied Intelligence* (2019), 1–18.

[31] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. 2014. Grey wolf optimizer. *Advances in engineering software* 69 (2014), 46–61.

[32] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 026113.

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

[34] Eneko Osaba, Javier Del Ser, David Camacho, Akemi Galvez, Andres Iglesias, and Iztok Fister. 2018. Community Detection in Weighted Directed Networks Using Nature-Inspired Heuristics. In *International Conference on Intelligent Data Engineering and Automated Learning.* Springer, 325–335.

[35] Eneko Osaba, Xin-She Yang, Fernando Diaz, Pedro Lopez-Garcia, and Roberto Carballedo. 2016. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Engineering Applications of Artificial Intelligence* 48 (2016), 59–71.

[36] Eneko Osaba, Xin-She Yang, Fernando Diaz, Enrique Onieva, Antonio D Masegosa, and Asier Perallos. 2017. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Computing* 21, 18 (2017), 5295–5308.

[37] Eneko Osaba, Xin-She Yang, Iztok Fister Jr, Javier Del Ser, Pedro Lopez-Garcia, and Alejo J Vazquez-Pardavila. 2019. A Discrete and Improved Bat Algorithm for solving a medical goods distribution problem with pharmacological waste collection. *Swarm and Evolutionary Computation* 44 (2019), 273–286.

[38] A. Panizo, G. Bello-Orgaz, A. Ortega, and D. Camacho. 2018. *Community finding in dynamic networks using a genetic algorithm improved via a hybrid immigrants scheme.* 591–598.

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

[40] Clara Pizzuti and Annalisa Socievole. 2018. A Genetic Algorithm for Community Detection in Attributed Graphs. In *International Conference on the Applications of Evolutionary Computation.* Springer, 159–170.

[41] Radu-Emil Precup, Radu-Codrut David, and Emil M Petriu. 2017. Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity. *IEEE Transactions on Industrial Electronics* 64, 1 (2017), 527–534.

[42] Radu-Emil Precup, Radu-Codrut David, Emil M Petriu, Alexandra-Iulia Szedlak-Stinean, and Claudia-Adina Bojan-Dragos. 2016. Grey wolf optimizer-based approach to the tuning of pi-fuzzy controllers with a reduced process parametric sensitivity. *IFAC-PapersOnLine* 49, 5 (2016), 55–60.

[43] Shadi Rahimi, Alireza Abdollahpouri, and Parham Moradi. 2018. A multi-objective particle swarm optimization algorithm for community detection in complex networks. *Swarm and Evolutionary Computation* 39 (2018), 297–309.

[44] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. 2009. GSA: a gravitational search algorithm. *Information sciences* 179, 13 (2009), 2232–2248.

[45] Giulio Rossetti and Rémy Cazabet. 2018. Community discovery in dynamic networks: a survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 35.

[46] S Salcedo-Sanz, J Del Ser, I Landa-Torres, S Gil-López, and JA Portilla-Figueras. 2014. The coral reefs optimization algorithm: a novel metaheuristic for efficiently solving optimization problems. *The Scientific World Journal* 2014 (2014).

[47] Sancho Salcedo-Sanz, Alvaro Pastor-Sánchez, Javier Del Ser, Luis Prieto, and Zong-Woo Geem. 2015. A coral reefs optimization algorithm with harmony search operators for accurate wind speed prediction. *Renewable Energy* 75 (2015), 93–101.

[48] Bilal Saoud. 2018. Networks clustering with bee colony. *Artificial Intelligence Review* (2018), 1–13.

[49] Cheng Shi, Yanchen Liu, and Pan Zhang. 2018. Weighted community detection and data clustering using message passing. *Journal of Statistical Mechanics: Theory and Experiment* 2018, 3 (2018), 033405.

[50] Amritpal Singh, Sahil Garg, Shalini Batra, and Neeraj Kumar. 2019. Probabilistic data structure-based community detection and storage scheme in online social networks. *Future Generation Computer Systems* 94 (2019), 173–184.

[51] Javier Torregrosa and Ángel Panizo. 2018. RiskTrack: Assessing the Risk of Jihadi Radicalization on Twitter Using Linguistic Factors. In *International Conference on Intelligent Data Engineering and Automated Learning.* Springer, 15–20.

[52] Jingyun Wang and Sanyang Liu. 2019. A Novel Discrete Particle Swarm Optimization Algorithm for Solving Bayesian Network Structures Learning Problem. *International Journal of Computer Mathematics* just-accepted (2019), 1–23.

[53] Bryce G Westlake and Martin Bouchard. 2016. Liking and hyperlinking: Community detection in online child sexual exploitation networks. *Social science research* 59 (2016), 23–36.

[54] Xin-She Yang. 2010. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation* 2, 2 (2010), 78–84.

[55] Xin-She Yang. 2010. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010).* Springer, 65–74.

[56] Xin-She Yang and Suash Deb. 2009. Cuckoo search via Lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on.* IEEE, 210–214.

[57] Xin-She Yang, Suash Deb, and Sudhanshu K Mishra. 2018. Multi-Species Cuckoo Search Algorithm for Global Optimization. *Cognitive Computation* 10, 6 (2018), 1085–1095.

[58] Krista Rizman Žalik. 2019. Evolution Algorithm for Community Detection in Social Networks Using Node Centrality. In *Intelligent Methods and Big Data in Industrial Applications.* Springer, 73–87.

[59] Yiwen Zhong, Juan Lin, Lijin Wang, and Hui Zhang. 2018. Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem. *Swarm and Evolutionary Computation* (2018).