Dynamic Partitioning of Evolving Graph Streams using Nature-inspired Heuristics

Eneko Osaba[1], Miren Nekane Bilbao[2], Andres Iglesias[3,4], Javier Del Ser[1,2], Akemi Galvez[3,4], Iztok Fister Jr.[5], and Iztok Fister[5]

[1] TECNALIA, 48160 Derio, Spain
[2] University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain
[3] Universidad de Cantabria, 39005 Santander, Spain
[4] Toho University, Funabashi, Japan
[5] University of Maribor, Maribor, Slovenia
eneko.osaba@tecnalia.com

Abstract. Detecting communities of interconnected nodes is a frequently addressed problem in situation that be modeled as a graph. A common practical example is this arising from Social Networks. Anyway, detecting an optimal partition in a network is an extremely complex and highly time-consuming task. This way, the development and application of meta-heuristic solvers emerges as a promising alternative for dealing with these problems. The research presented in this paper deals with the optimal partitioning of graph instances, in the special cases in which connections among nodes change dynamically along the time horizon. This specific case of networks is less addressed in the literature than its counterparts. For efficiently solving such problem, we have modeled and implements a set of meta-heuristic solvers, all of them inspired by different processes and phenomena observed in Nature. Concretely, considered approaches are Water Cycle Algorithm, Bat Algorithm, Firefly Algorithm and Particle Swarm Optimization. All these methods have been adapted for properly dealing with this discrete and dynamic problem, using a reformulated expression for the well-known modularity formula as fitness function. A thorough experimentation has been carried out over a set of 12 synthetically generated dynamic graph instances, with the main goal of concluding which of the aforementioned solvers is the most appropriate one to deal with this challenging problem. Statistical tests have been conducted with the obtained results for rigorously concluding the Bat Algorithm and Firefly Algorithm outperform the rest of methods in terms of Normalized Mutual Information with respect to the true partition of the graph.

Keywords: Bio-inspired computation · Nature-inspired heuristics · Evolving Graphic Streams · Community Detection

1 Introduction

With the impactful arrival of Social Networks, a remarkable number of tools and methods have been developed for excerpting information and insights from

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

the multiple interrelations between their users [1]. In this regard, the knowledge that can be drawn using these methods range from evaluating the influence of a specific node in the whole graph (*centrality*), to enriched ways of network visualizing or the finding of shortest paths amidst a pair or groups of nodes. As has been seen in late years, all this information can be used for interesting practical goals, such as the identification of radicalization risk [2, 3], or child abuse detections [4, 5].

Among all the valuable knowledge that can be inferred from these Social Networks, the detection of different communities within the nodes is probably one of the most recurrent tasks, being the main focus of lots of recently developed scientific studies. Specifically, a *community* refers to a group of nodes which meet the general principles of strong intra-connectivity (strong links between members of the same community) and weak inter-connectivity (weak connectivity with nodes belonging to other partitions). Furthermore, the redefinition of these measured parameters leads to the characterization of different networks (weighted, multiple edges, directed, self loops), quantifying the cohesiveness of any candidate community.

Additionally, diverse efficient metrics have been projected in the literature for evaluating the quality of proposed partitions. Each of these metrics take different assumptions for measuring the connectivity, yielding to a single quality value for the community. Permanence [6], Surprise [7] and Newman and Girvan's Modularity [8] are some frequently used examples. In this specific study, the last-mentioned Modularity is considered.

In terms of computational solvers, many different approaches have been proposed in recent years for finding communities towards (explicitly or implicitly) optimizing one of the aforementioned metrics. Related with the research presented in this manuscript, a growing community is currently emerging devoted to adapting well-known (or even develop new) heuristic optimization algorithms, directly adopting one modularity metric as objective function. Many of these works can be found in the recent literature, focused on assorting combinations of algorithmic approaches, network instances and quality measurement metric functions. In this context, Genetic Algorithms (GA) crop up as one of the most often employed methods for discovering partitions in networks of different characteristics [9, 10]. Besides GAs, many techniques that fall inside the umbrella of Evolutionary Computation and Swarm Intelligence have been proposed in last years, with the main goal of solving same or similar problem. Some of these solvers are the Ant Colony Optimization [11] or Particle Swarm Optimization [12]. Furthermore, interestingly for the scope of this work, a growing strand of the related literature is currently committed to the adaptation of modern nature-inspired algorithms for community detection in networks. Some examples are the Firefly Algorithm [13], Bat Algorithm [14] or Artificial Bee Colony [15].

At this point, it is worth to mention a specific case of networks, which are characterized by their dynamism. Since time immemorial, relationships between human beings tend to be different over time. In this sense, people are used to strengthen existing (or build new) relations over their whole lives, meanwhile

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

some others are weakened (or even broken up). For this reason, if we analyze the relationship history of a single person in a long enough time lapse, we will surely find some dynamism. Of course, this evolution is also reflected in Social Networks. This way, Dynamic Networks are special cases of graphs in which the number of links among nodes, the strength of these links or even the number of nodes can suffer changes along the time. Thus, a Dynamic Network can be seen as an Evolving Graph Stream, in which the evolution of the network is described step by step.

Dynamic Networks have also been the subject matter of a recently published interesting works, focused on dynamic community finding [16]. In [17], for example, a multi-objective Bat Algorithm is presented for dealing with this problem. GAs have also been occasionally adapted to this kind of graphs, as can be seen in [18] or [19]. In any case, the amount of scientific material related to this topic is much fewer than the one associated to stationary networks.

With all this, the research presented in this paper aims at taking a step further over this scarce state of the art by elaborating on several directions: 1) we face the problem of finding communities in dynamic networks, far less used than stationary ones; 2) we adopt the Hamming Distance as a metric to assess the similarity between different solutions and partitions, and 3) we evaluate these algorithmic features with a group of ad-hoc adopted nature-inspired solvers: Water Cycle Algorithm (WCA, [20]), Bat Algorithm (BA, [21]), Firefly Algorithm (FA, [22]) and Particle Swarm Optimization (PSO, [23]). Along the paper, details on how these methods have been adapted to the proposed problem are exposed, as well as a justification of their expected benefits. In order to measure the performance of each method, results reached over 12 synthetically generated datasets are compared and discussed, based on their efficiency on discovering their ground-of-truth partition. Furthermore, with the intention of drawing fair and rigorous conclusions, two different statistical tests (Friedman's and Holm's) are employed with the obtained outcomes.

The rest of the paper is structured as follows: in the next Section 2, the problem of detecting communities in dynamic networks is formulated. After that, the heuristic solvers are described in Section 3, while the experimentation is displayed in Section 4. This manuscript ends with conclusions and further work in Section 5.

2 Problem Statement

In order to properly deal with the aforementioned community detection problem, we start by modeling the dynamic network as a graph $\mathcal{G} \doteq \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ represent the group of $|\mathcal{V}| = V$ vertex or nodes of the network, whilst $\mathcal{E}$ corresponds to a set that describes the dynamic situation of the links (or edges connecting every pair of nodes) along the whole time horizon. At this point, it is interesting to clarify that the time horizon is comprised by a set of graph snapshots, each one describing the specific situation of the network at one exact moment. This

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

way, $\mathcal{E} \doteq \{e_1, \ldots, e_N\}$, corresponding e_n to the set of edges at timestamp n. In our study, we have established N in 30 and 40.

Another noteworthy point is that, while the relation between the nodes vary along the time, the number and situation of all vertex remains constant along the time horizon. Furthermore, the weight of each link connecting every pair of nodes v and v' is $w_{v,v'} = 1$. We also assume that $w_{v,v} = 0$ (i.e. no self-loops) and that $w_{v,v'} = 0$ if nodes v and v' are not connected. For notational convenience, we define an adjacency matrix $\mathbf{W}$ given by $\mathbf{W} \doteq \{w_{v,v'} : v, v' \in \mathcal{V}\}$, and fulfilling $\mathrm{Tr}(\mathbf{W}) = 0$. Finally, symmetry is always assumed in $\mathcal{G}$, meaning that $w_{v,v'} = w_{v',v}$. From now on, and in order to properly contemplate the dynamism inherent to the problem, weights are represented as $w^n_{v,v'}$, depicting the weight at timestamp n.

With all this, the problem of finding communities in the graph $\mathcal{G}$ is conceived in this study as the partition of the vertex set $\mathcal{V}$ into a number of non-empty and disjoint groups, each with a non-fixed size. Assuming that M is the number of groups 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). We can thus denote the community to which node v belongs as $\mathcal{V}^v \in \widetilde{\mathcal{V}}$. Analogously, from now on, the set of partitions will be represented as $\mathcal{V}_n$, depicting $\widetilde{\mathcal{V}_n} \doteq \{\mathcal{V}^n_1, \ldots, \mathcal{V}^n_M\}$ the group communities found at a specific timestamp n. This way, dynamism can be contemplated in this formulation.

Furthermore, as has been advanced in the introduction, the Newman and Girvan's Modularity formula has been adopted for measuring the quality of a given partition. This well-known function has been employed in myriad of works before, and its adequacy has been proven extensively [24–26]. This way, the measure of modularity for the considered community can be computed by:

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

where k^n_i is the degree of node i, $|E_n|$ is the total number of edges in the network, and $\delta(v, v')_n$ represents the Kronecker delta symbol, all of them contextualized in timestamp n. Clearly explained, $\delta(v, v')_n$ 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 partition set by $\widetilde{\mathcal{V}_n}$ (and 0 otherwise). All of them also contextualized by the timestamp n.

Therefore, detecting a *good* partition $\widetilde{\mathcal{V}}^*_n$ of the considered network $\mathcal{G}$ can be casted as:

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

denoting $\mathcal{B}_V$ the group of possible partitions of $\mathcal{V}_n$ 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 [27]. This means that if we consider a network composed by $V = 20$ nodes, it can be partitioned in approximately $517.24 \cdot 10^{12}$ different manners. Thus, considering that a separated evaluation can be computed in 1 microsecond on

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

average, we would need more than one and a half years to check all possible combinations. This situation confirms the necessity of using a heuristic method for the efficient exploration of the solution space.

Finally, it is worth-mentioning at this point that the main goal of this paper is to solve the described problem in n consecutive timestamps, trying to reach the highest precision possible. To do that, it is important to consider different degrees of similarity between adjacent graph snapshots n and $n+1$.

3 Proposed Nature-inspired Solvers

With the aim of properly deal with the above described problem, several nature-inspired methods have been proposed. Prior to the description of each considered solver, some common design aspects are described in what follows, related to the encoding strategy, solution repair mechanism and the method used for comparing different solutions.

Being one of the most important aspects while heuristic developing, it is interesting to mention that *label-based representation* [28] has been adopted for encoding purposes. This way, each solutions is codified as a permutation $\mathbf{x} = [c_1, c_2, \ldots, c_V]$ of V integers from the range $[1, \ldots, V]$, where V denotes the number of nodes in the network. Furthermore, c_v denotes the cluster label to which node v belongs to. For instance, considering a $V = 12$ network, a possible solution could be $\mathbf{x} = [1, 2, 2, 1, 1, 2, 2, 3, 2, 3, 3, 3]$, which means that the partition depicted is $\widetilde{\mathcal{V}} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$, where $\mathcal{V}_1 = \{1, 4, 5\}$, $\mathcal{V}_2 = \{2, 3, 6, 7, 9\}$ and $\mathcal{V}_3 = \{8, 10, 11, 12\}$.

With the intention of avoiding ambiguities in the representation, a repairing mechanism has been built, which is partly inspired from the one presented in [29]. Thanks to this procedure, which is applied to every newly created solution, ambiguities generated by solutions such as $\mathbf{x} = [4, 2, 2, 4, 4, 2, 2, 3, 2, 3, 3, 3]$ and $\mathbf{x} = [7, 1, 1, 7, 7, 1, 1, 4, 1, 4, 4, 4]$ (which represent the same partition) are solved, transforming both of them to the above shown $\mathbf{x} = [1, 2, 2, 1, 1, 2, 2, 3, 2, 3, 3, 3]$.

An additional important aspect of the developed methods is how the similarity between different solutions is measured. This similarity is the basis of the movement strategies inherent to each of the proposed techniques. Thus, the well-known Hamming Distance has been chosen for this purpose. This function has previously used in several studies with the same objective, as can be seen in [30], verifying its adequacy in this context. Concretely, Hamming Distance is calculated as the number of non-corresponding elements between two individuals. This way, and considering two partitions $\mathbf{x} = [1, 2, \mathbf{2}, 1, \mathbf{2}, 2, 2, 3, 2, 3, \mathbf{1}, \mathbf{1}]$ and $\mathbf{x} = [1, 2, \mathbf{1}, 1, \mathbf{1}, 2, 2, 3, 2, 3, \mathbf{3}, \mathbf{3}]$ their Hamming Distance $D_H(\mathbf{x}, \mathbf{x}')$ would equal 4.

Finally, four movement functions have been implemented for evolving individuals along the search process. The use of these operators is based on the distance between two different partitions. Specifically, these functions are named CE_1, CE_3, CC_1 and CC_3. On the one hand, the subscript represents the number of randomly selected nodes, which are extracted from its corresponding community.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

On the other hand, in CE_* operators the taken nodes are re-inserted in already existing clusters, while in CC_* nodes can be inserted also in newly generated ones.

Now, the details of the considered metaheuristics are introduced:

WCA : The Water Cycle Algorithm was firstly proposed in [31] for solving continuous optimization problems. As has been made in other scientific works [32], a discrete adaptation has been conducted for properly facing the problem addressed in this research. Regarding this approach, and laying aside the aspects mentioned in the beginning of this section, the most crucial mechanism to implement is the way in which streams and rivers flow to their corresponding leading individual. Thus, and following the same philosophy of the basic WCA, the movement of each stream $p_{str} \in \mathcal{P}_{str}$ towards its river $\lambda(p_{str})$ at each generation $t \in \{1, \ldots, T\}$ is set to:

$$\mathbf{x}^{p_{str}}(t+1) = \Psi\left(\mathbf{x}^{p_{str}}(t), \min\left\{V, \left\lfloor rand \cdot \theta \cdot D_H(\mathbf{x}^{p_{str}}(t), \mathbf{x}^{\lambda(p_{str})}(t))\right\rfloor\right\}\right), \quad (3)$$

where $rand$ is a continuous random variable uniformly distributed in $\mathbb{R}[0, 1]$, θ is a heuristic parameter. Furthermore, $\Psi(\mathbf{x}, Z) \in \{CE_1, CE_3, CC_1, CC_3\}$, each one parametrized by the number of times Z this function is applied to $\mathbf{x}$. This way, the best movement resulting from all the Z movements carried out on $\mathbf{x}$ is selected as output. The same logic is adopted for the movements of a river or a stream towards the sea, just replacing $\mathbf{x}^{\lambda(p_{str})}(t)$ by $\mathbf{x}^{p_{sea}}(t)$.

Additionally, the *inclination* mechanism recently proposed in [32] is also used in the developed discrete WCA, with the main goal of boosting the exploration ability of the method. This simple mechanism endows the algorithm with the intelligence of properly selection the movement operator to use at each iteration for every individual. This election depends on the specific situation of each raindrop. Concretely, each time an individual is about to perform a movement, the aforementioned *inclination* $\xi(\mathbf{x}, \mathbf{x}')$ is computed using as reference the $D_H(\mathbf{x}, \mathbf{x}')$ to its designated river/sea $\mathbf{x}'$. Particularly, $\xi(\cdot, \cdot)$ is equal to $V/D_H(\cdot, \cdot)$. Accordingly, the bigger $D_H(\cdot, \cdot)$ is, the higher $\xi(\cdot, \cdot)$ should be, forcing the method to perform a *fast move* with a higher probability. On the contrary, if $D_H(\cdot, \cdot)$ is small the inclination decreases, suggesting that the search is in a promising area of the solution space, and performing a *slow move* with higher probability. In the present study, four different movement functions have been considered, deeming CC_* as *fast moves*, and CE_* as *slow moves*. Finally, the evaporation and raining procedures remain in the same way as in the original WCA. Specifically, the raining process comprises a number R of consecutive CC_3 movements.

BA : As for the WCA, the classic BA was firstly introduced for solving continuous optimization problems. For this reason, a discrete adaptation has been conducted in order to correctly face the problem addressed in this paper. As in many other adaptations [33], each bat represents a feasible solution of the problem. Both concepts of loudness A_i and pulse emission r_i have been considered

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

in exactly the same form as in the basic version of the method. Furthermore, in order to simplify the complexity of the approach, frequency f_i parameter has not been deemed. Finally, the velocity v_i has been adapted, considering the Hamming Distance as measure function to evaluate the similarity between two bats. This way, $v_i^t = \text{Random}[1, D_H(\mathbf{x_i}, \mathbf{x}_*)]$. In other words, the v_i of a bat i at time step t is a random number, which follows a discrete uniform distribution between 1 and the difference between this i and the best bat of the swarm. Finally, the way in which a bat moves is determined similarly to Expression (3), using v_i as the number of movements considered. Furthermore, the inclination concept is also implemented in this discrete version of the BA, also following the same philosophy as for the WCA, and using the best bat as reference.

FA : Again, the classic FA cannot be applied directly to address discrete problems. For this reason, some modifications have been performed over the original version of the FA. As for the BA, each firefly in the swarm represents a solution for the problem. Additionally, the concept of light absorption is considered, which is essential for the adjustment of fireflies' attractiveness. As has been mentioned, the distance between two different individuals is calculated by the Hamming Distance. Finally, the movement of a firefly attracted to another brighter one is determined following the same logic depicted in Expression (3). Besides that, when a firefly is prepared to perform a movement to another firefly, it examines its distance. If it is higher than $V/2$, it can be assumed that it is far from its counterpart. Therefore, it carries out a *wide move*, using a CC_* operator. Otherwise, a *short move* is performed by a CE_* function. This mechanism has been added aiming the adapt the *inclination* functionality above described.

PSO : The last considered approach is the well-known Particle Swarm Optimization, which has been already applied to discrete problems in multiple times [34, 35]. Taking as inspiration previous discrete adaptations of the PSO, each particle also deems a feasible solution for the dealt problem. Velocity parameter v_i has been considered analogously to what has been done for the BA. Additionally, both movement operators and inclination mechanism have also been contemplated for the PSO in the same way as for the FA, WCA and BA. Finally, Hamming Distance has been taken as similarity measurement function.

4 Experimentation and Results

With the aim of properly evaluating the performance of the four developed solvers, computer experiments have been run using a heterogeneous set of synthetically generated network instances. All these instances have been created using the well accepted DANCer platform [36, 37], and with the aim of covering a diverse set of common situations in dynamic environments. Specifically, the benchmark is composed by 12 different 100-noded datasets. The name of each instance is built joining the values of five different parameters:

– *Size of the problem*: In all cases this value is 100.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

- *Communities*: The number of communities that compose the ground of truth solution.
- *Generations*: Number of generations run for each Graph Snapshot e_i.
- *Variability*: The difference between adjacent graph snapshot e_i and e_{i+1}. This parameter can adopt three different values: Slight (variety of 5% between adjacent snapshots), Medium (variety of 10%), and Dramatic (variety of 20%).
- *Transition*: Each instance is composed by 30 canonical snapshots, divided into two different families of 15 timestamps. If *Transition* takes *Abrupt* value, the transition between both families is directly made after the 15th timestamp. These instances are comprised just by these 30 snapshots. On the hand, if *Transition* takes *Gradual* value, this transition is made gradually, introducing 10 additional snapshots between the last timestamp of the first family, and the first timestamps of the second one. These datasets are finally composed by 40 snapshots.

This network construction approach allows us to measure the performance of the developed solvers over *noisy* versions of a graph characterized by a controlled underlying community distribution. This method opposes to the common practice by which the comparison is done based on the fitness value obtained by each technique. Finally, 10 independent runs have been executed for each solver and dataset, aiming at reaching statistically reliable insights. Regarding the ending criterion, it depends on both *Generations* and *Transition* parameters of the instance. Thus, depending on the values taken by these parameters, solvers end after *600*, *800*, *1500* or *2000* generations. The population size has been established in 50 for each method. In the concrete case of WCA, the number of rivers has been established in 9 (approximately 20% of the whole population), leading to a number of 40 streams. On the other hand, the maximum distance for evaporation) and R have been respectively set to 5% and a uniform random value from $\mathbb{N}[0, \lfloor 0.5V \rfloor]$. On the other hand, for FA γ=0.95. Finally, for BA α=β=0.98, A_i^0=1.0 and r_i^0=0.1. For the development and parameterization of these methods, the guidelines given in [32, 33, 38] have been followed.

Table 1. Obtained NMI results (average/best/standard deviation) using WCA, BA, FA and PSO. Best average results have been highlighted in bold.

	WCA			BA			FA			PSO		
Instance	Avg	Best	Std	Avg	Best	Std	Avg	Best	Std	Avg	Best	Std
100_7_20_*Sli_Abr*	0.617-0.439	0.724-0.632	0.029-0.036	**0.668**-0.484	0.811-0.676	0.035-0.062	0.646-**0.571**	0.739-0.682	0.038-0.034	0.573-0.460	0.665-0.608	0.024-0.0334
100_7_20_*Sli_Grad*	0.619-0.482	0.772-0.632	0.030-0.034	**0.676**-0.461	0.757-0.588	0.027-0.053	0.658-**0.571**	0.760-0.681	0.034-0.036	0.580-0.463	0.657-0.590	0.021-0.042
100_7_50_*Sli_Abr*	0.671-0.529	0.784-0.683	0.034-0.035	**0.699**-0.513	0.755-0.670	0.026-0.041	0.655-**0.577**	0.760-0.681	0.034-0.036	0.640-0.492	0.759-0.664	0.031-0.045
100_7_50_*Sli_Grad*	0.662-0.540	0.784-0.708	0.029-0.034	**0.689**-0.500	0.777-0.635	0.032-0.054	0.650-**0.580**	0.751-0.689	0.033-0.030	0.638-0.486	0.758-0.700	0.031-0.039
100_8_20_*Med_Abr*	0.619-0.445	0.860-0.648	0.029-0.040	**0.660**-0.500	0.869-0.653	0.040-0.042	0.633-**0.519**	0.811-0.618	0.030-0.033	0.561-0.474	0.698-0.597	0.027-0.030
100_8_20_*Med_Grad*	0.620-0.463	0.840-0.620	0.027-0.040	**0.654**-0.460	0.861-0.652	0.041-0.043	0.643-**0.526**	0.799-0.642	0.031-0.040	0.571-0.464	0.701-0.571	0.026-0.031
100_8_50_*Med_Abr*	0.681-0.495	0.861-0.663	0.029-0.036	**0.710**-0.483	0.870-0.635	0.036-0.035	0.640-**0.522**	0.802-0.657	0.031-0.036	0.601-0.498	0.835-0.633	0.033-0.034
100_8_50_*Med_Grad*	0.674-0.492	0.861-0.684	0.030-0.039	**0.683**-0.482	0.861-0.691	0.032-0.049	0.650-**0.524**	0.812-0.644	0.031-0.032	0.612-0.466	0.848-0.610	0.030-0.034
100_9_20_*Dram_Abr*	0.593-0.474	0.726-0.631	0.029-0.038	**0.639**-0.501	0.820-0.684	0.034-0.039	0.607-**0.569**	0.741-0.657	0.030-0.027	0.535-0.539	0.643-0.657	0.029-0.029
100_9_20_*Dram_Grad*	0.589-0.515	0.696-0.641	0.029-0.033	**0.641**-0.543	0.793-0.697	0.025-0.063	0.606-**0.570**	0.753-0.706	0.031-0.029	0.538-0.538	0.643-0.644	0.024-0.030
100_9_50_*Dram_Abr*	0.642-0.538	0.791-0.684	0.028-0.036	**0.677**-0.550	0.811-0.675	0.026-0.026	0.614-**0.559**	0.765-0.641	0.031-0.028	0.586-0.529	0.727-0.646	0.032-0.039
100_9_50_*Dram_Grad*	0.638-0.544	0.820-0.694	0.024-0.030	**0.651**-0.548	0.826-0.664	0.042-0.038	0.620-**0.553**	0.764-0.643	0.035-0.026	0.585-0.514	0.701-0.641	0.028-0.031
Friedman's non-parametric test (mean ranking)												
Rank	2.5-3.0			**1.0**-2.83			2.5-**1.0**			4.0-3.16		

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

In Table 1, outcomes (average/best/standard deviation) obtained by the four solvers are shown. Each of these values are divided into two different sub-values, 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 ones 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 depict 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 them.

A first analysis reveals that BA is the best alternative in the 100% of the cases for the first family, while the FA emerges as the best alternative in all the datasets for the second families. On the other hand, the performance of PSO is much lower than the other alternatives, while WCA is one step behind BA and FA. In this sense, we can see how WCA is much worse than the BA for the first families but similar to FA in this context. On the contrary, it is much worse than FA for second families while it obtains similar outcomes than BA. Although it may seem unintuitive, this switch in the quality of results has a logical explanation, which is based in the fact that BA is a better for the exploitation of the solution space, while FA shows a better adapting capacity thank to its enhanced exploratory ability.

Following the guidelines in [39] and [40], 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 if there are significant differences in the results obtained by all reported methods. Last row of Table 1 displays the mean ranking returned by this non-parametric test for each of the compared algorithms and families (the lower the rank, the better the performance). Thus, for the first family the Friedman statistic (distributed according to χ^2 with 4 degrees of freedom) was equal to 32.4. Furthermore, the confidence interval has been set to 99%, being 13.27 the critical point in a χ^2 distribution with 4 degrees of freedom. Since $32.4 > 13.27$, it can be concluded that there are significant differences among the results, thus BA can be regarded as the method having the lowest rank. Regarding the second family, and taking into account also the results shown in Table 1, the Friedman statistic test is 22.0. Again, since $22.0 > 13.27$, the same conclusion is also applicable in this case.

The second statistical test is the Holm's post-hoc test. For correctly conducting this test, BA has been set as the control algorithm for the first family, whilst for the second FA has been established. Table 2 gathers the unadjusted and adjusted p-values obtained through the application of Holm's post-hoc procedure. From these p-values it can be concluded that BA, for the first case, and

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

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.

Finally, and seeking for the completeness of the present research, we show in Figure 1 and Figure 2 the evolution of the NMI and modularity along the whole execution for the instance 100_7_50_*Sli_Abr*. In Figure 1 the performance of the BA is depicted, while Figure 2 is devoted to FA. Both graphs represent the performance presented over the 10 runs.

Fam1 (BA as control)			Fam2 (FA as control)		
Algorithm	Unadjusted p	Adjusted p	Algorithm	Unadjusted p	Adjusted p
WCA	0.004427	0.008853	WCA	0.000148	0.000296
FA	0.004427	0.008853	BA	0.000504	0.000504
PSO	0.0	0.0	PSO	0.000039	0.000118

Table 2. Unadjusted and adjusted p-values obtained as a result of the application of Holm's post-hoc procedure using BA and FA as control algorithms.

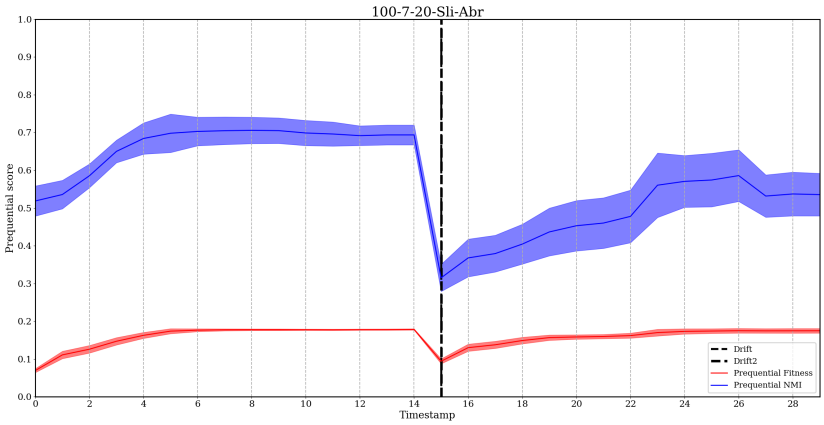

Fig. 1. Evolution of the NMI and fitness obtained by BA for 100_7_50_*Sli_Abr*. Blue line denotes the NMI. Red line depicts the modularity. Vertical grey line represents a timestamp alteration. Vertical black line points the family change.

5 Conclusions and Future Research Lines

In this study, community finding in dynamic graphs has been dealt by using four different nature-inspired meta-heuristics: Water Cycle Algorithm, Bat Algorithm, Firefly Algorithm and Particle Swarm Optimization. For this purpose, the detection of optimal partitions has been modeled as a discrete optimization problem, using an adapted Newman and Girvan's Modularity as evaluation function. All the four deemed methods have been adapted to face the particularities

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

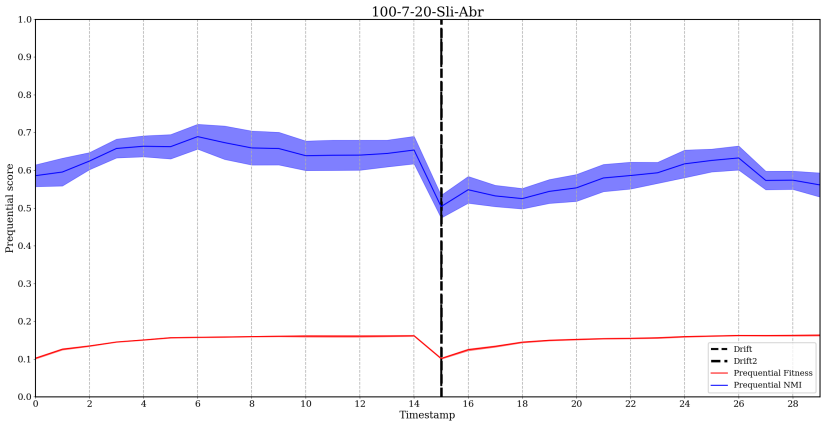

Fig. 2. Evolution of the NMI and fitness obtained by FA for 100_7_50_*Sli_Abr*. Blue line denotes the NMI. Red line depicts the modularity. Vertical grey line represents a timestamp alteration. Vertical black line points the family change.

of the solution space, such as the potential representational ambiguity of label encoding and the definition of distance between solutions to the problem. The performance of each approach has been evaluated using a benchmark composed by 12 dynamic networks, all of them comprised by 100 nodes, using as comparison criterion the Normalized Mutual Information (or NMI) regarding their *ground of truth* partition. Obtained outcomes demonstrated that BA and FA dominate over their counterparts with statistical significance.

As future work, we plan to conduct further efforts in different directions. The most imminent one is the adaption of additional nature-inspired, evolutionary and swarm intelligent methods, such as the Cuckoo Search [41] or Grey Wolf Optimizer [42]. The performance shown by these techniques applied to other optimization problems [43–46] lead us to consider them as potentially promising methods. Moreover, we will also consider larger network instances that the ones employed in this paper. Finally, we have the firm intention of exploring the hybridization of these heuristic with local search techniques, trying to mimic the operation of other heuristics found in the related state of the art, such as recently contributed message passing procedures [47] and other techniques renowned for their good scalability [48].

Acknowledgements

E. Osaba and J. Del Ser would like to thank the Basque Government for its funding support through the EMAITEK program. A. Iglesias and A. Galvez acknowledge the financial support from the projects TIN2017-89275-R (AEI/FEDER, UE) and PDE-GIR (H2020, MSCA program, ref. 778035). Iztok Fister and Iztok Fister Jr. acknowledge the financial support from the Slovenian Research Agency (Research Core Founding No. P2-0041 and P2-0057).

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

References

1. Bello-Orgaz, G., Jung, J.J., Camacho, D.: Social big data: Recent achievements and new challenges. Information Fusion **28** (2016) 45–59
2. Lara-Cabrera, R., Pardo, A.G., Benouaret, K., Faci, N., Benslimane, D., Camacho, D.: Measuring the radicalisation risk in social networks. IEEE Access **5** (2017) 10892–10900
3. Torregrosa, J., Panizo, Á.: Risktrack: Assessing the risk of jihadi radicalization on twitter using linguistic factors. In: International Conference on Intelligent Data Engineering and Automated Learning, Springer (2018) 15–20
4. Westlake, B.G., Bouchard, M.: Liking and hyperlinking: Community detection in online child sexual exploitation networks. Social science research **59** (2016) 23–36
5. Villar-Rodríguez, E., Del Ser, J., Torre-Bastida, A.I., Bilbao, M.N., Salcedo-Sanz, S.: A novel machine learning approach to the detection of identity theft in social networks based on emulated attack instances and support vector machines. Concurrency and Computation: Practice and Experience **28**(4) (2016) 1385–1395
6. Chakraborty, T., Srinivasan, S., Ganguly, N., Mukherjee, A., Bhowmick, S.: On the permanence of vertices in network communities. In: ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2014) 1396–1405
7. Aldecoa, R., Marín, I.: Deciphering network community structure by surprise. PloS one **6**(9) (2011) e24195
8. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Physical review E **69**(2) (2004) 026113
9. Žalik, K.R.: Evolution algorithm for community detection in social networks using node centrality. In: Intelligent Methods and Big Data in Industrial Applications. Springer (2019) 73–87
10. Pizzuti, C., Socievole, A.: A genetic algorithm for community detection in attributed graphs. In: International Conference on the Applications of Evolutionary Computation, Springer (2018) 159–170
11. Pizzuti, C.: Evolutionary computation for community detection in networks: a review. IEEE Transactions on Evolutionary Computation **22**(3) (2018) 464–483
12. Rahimi, S., Abdollahpouri, A., Moradi, P.: A multi-objective particle swarm optimization algorithm for community detection in complex networks. Swarm and Evolutionary Computation **39** (2018) 297–309
13. Del Ser, J., Lobo, J.L., Villar-Rodriguez, E., Bilbao, M.N., Perfecto, C.: Community detection in graphs based on surprise maximization using firefly heuristics. In: IEEE Congress on Evolutionary Computation (CEC), IEEE (2016) 2233–2239
14. Hassan, E.A., Hafez, A.I., Hassanien, A.E., Fahmy, A.A.: A discrete bat algorithm for the community detection problem. In: International Conference on Hybrid Artificial Intelligence Systems, Springer (2015) 188–199
15. Saoud, B.: Networks clustering with bee colony. Artificial Intelligence Review (2018) 1–13
16. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: a survey. ACM Computing Surveys (CSUR) **51**(2) (2018) 35
17. Messaoudi, I., Kamel, N.: A multi-objective bat algorithm for community detection on dynamic social networks. Applied Intelligence (2019) 1–18
18. Li, Z., Liu, J.: A multi-agent genetic algorithm for community detection in complex networks. Physica A: Statistical Mechanics and its Applications **449** (2016) 336–347

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

19. Folino, F., Pizzuti, C.: An evolutionary multiobjective approach for community discovery in dynamic networks. IEEE Transactions on Knowledge and Data Engineering **26**(8) (2014) 1838–1852
20. Eskandar, H., Sadollah, A., Bahreininejad, Ardeshir, Hamdi, M.: Water cycle algorithm a novel metaheuristic optimization method for solving constrained engineering optimization problems. Applied Soft Computing **110**(111) (2012) 151–166
21. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer (2010) 65–74
22. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-Inspired Computation **2**(2) (2010) 78–84
23. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of machine learning. Springer (2011) 760–766
24. Leicht, E.A., Newman, M.E.: Community structure in directed networks. Physical review letters **100**(11) (2008) 118703
25. Chakraborty, T., Dalmia, A., Mukherjee, A., Ganguly, N.: Metrics for community analysis: A survey. ACM Computing Surveys (CSUR) **50**(4) (2017) 54
26. Chen, M., Kuzmin, K., Szymanski, B.K.: Community detection via maximization of modularity and its variants. IEEE Transactions on Computational Social Systems **1**(1) (2014) 46–65
27. Harris, J.M., Hirst, J.L., Mossinghoff, M.J.: Combinatorics and graph theory. Volume 2. Springer (2008)
28. Hruschka, E.R., Campello, R.J., Freitas, A.A., et al.: A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **39**(2) (2009) 133–155
29. Falkenauer, E.: Genetic algorithms and grouping problems. Wiley & Sons., Inc., New York, NY, USA (1998)
30. Osaba, E., Del Ser, J., Camacho, D., Galvez, A., Iglesias, A., Fister, I.: Community detection in weighted directed networks using nature-inspired heuristics. In: International Conference on Intelligent Data Engineering and Automated Learning, Springer (2018) 325–335
31. Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M.: Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures **110** (2012) 151–166
32. Osaba, E., Del Ser, J., Sadollah, A., Bilbao, M.N., Camacho, D.: A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem. Applied Soft Computing **71** (2018) 277–290
33. Osaba, E., Yang, X.S., Fister Jr, I., Del Ser, J., Lopez-Garcia, P., Vazquez-Pardavila, A.J.: 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
34. Chen, A.l., Yang, G.k., Wu, Z.m.: Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University-Science A **7**(4) (2006) 607–614
35. Zhong, Y., Lin, J., Wang, L., Zhang, H.: Discrete comprehensive learning particle swarm optimization algorithm with metropolis acceptance criterion for traveling salesman problem. Swarm and Evolutionary Computation (2018)
36. Largeron, C., Mougel, P.N., Benyahia, O., Zaïane, O.R.: Dancer: dynamic attributed networks with community structure generation. Knowledge and Information Systems **53**(1) (2017) 109–151

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29

37. Benyahia, O., Largeron, C., Jeudy, B., Zaïane, O.R.: Dancer: Dynamic attributed network with community structure generator. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer (2016) 41–44
38. Osaba, E., Yang, X.S., Diaz, F., Onieva, E., Masegosa, A.D., Perallos, A.: 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
39. Derrac, J., García, S., Molina, D., Herrera, F.: 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
40. Osaba, E., Carballedo, R., Diaz, F., Onieva, E., Masegosa, A., Perallos, A.: Good practice proposal for the implementation, presentation, and comparison of metaheuristics for solving routing problems. Neurocomputing **271** (2018) 2–8
41. Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, IEEE (2009) 210–214
42. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in engineering software **69** (2014) 46–61
43. Precup, R.E., David, R.C., Petriu, E.M., Szedlak-Stinean, A.I., Bojan-Dragos, C.A.: 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
44. Precup, R.E., David, R.C., Petriu, E.M.: Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity. IEEE Transactions on Industrial Electronics **64**(1) (2017) 527–534
45. Yang, X.S., Deb, S., Mishra, S.K.: Multi-species cuckoo search algorithm for global optimization. Cognitive Computation **10**(6) (2018) 1085–1095
46. He, X.S., Wang, F., Wang, Y., Yang, X.S.: Global convergence analysis of cuckoo search using markov theory. In: Nature-Inspired Algorithms and Applied Optimization. Springer (2018) 53–67
47. Shi, C., Liu, Y., Zhang, P.: Weighted community detection and data clustering using message passing. Journal of Statistical Mechanics: Theory and Experiment **2018**(3) (2018) 033405
48. Lu, H., Halappanavar, M., Kalyanaraman, A.: Parallel heuristics for scalable community detection. Parallel Computing **47** (2015) 19–37

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_29