Community Detection in Weighted Directed Networks Using Nature-Inspired Heuristics

Eneko Osaba[1](✉), Javier Del Ser[1,2,3], David Camacho[4], Akemi Galvez[5,6], Andres Iglesias[5,6], Iztok Fister Jr.[7], and Iztok Fister[7]

[1] TECNALIA, 48160 Derio, Spain
eneko.osaba@tecnalia.com
[2] University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain
[3] Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain
[4] Universidad Autonoma de Madrid, 28049 Madrid, Spain
[5] Universidad de Cantabria, 39005 Santander, Spain
[6] Toho University, Funabashi, Japan
[7] University of Maribor, Maribor, Slovenia

Abstract. Finding groups from a set of interconnected nodes is a recurrent paradigm in a variety of practical problems that can be modeled as a graph, as those emerging from Social Networks. However, finding an optimal partition of a graph is a computationally complex task, calling for the development of approximative heuristics. In this regard, the work presented in this paper tackles the optimal partitioning of graph instances whose connections among nodes are directed and weighted, a scenario significantly less addressed in the literature than their unweighted, undirected counterparts. To efficiently solve this problem, we design several heuristic solvers inspired by different processes and phenomena observed in Nature (namely, Water Cycle Algorithm, Firefly Algorithm, an Evolutionary Simulated Annealing and a Population based Variable Neighborhood Search), all resorting to a reformulated expression for the well-known modularity function to account for the direction and weight of edges within the graph. Extensive simulations are run over a set of synthetically generated graph instances, aimed at elucidating the comparative performance of the aforementioned solvers under different graph sizes and levels of intra- and inter-connectivity among node groups. We statistically verify that the approach relying on the Water Cycle Algorithm outperforms the rest of heuristic methods in terms of Normalized Mutual Information with respect to the true partition of the graph.

Keywords: Bio-inspired computation · Community detection

1 Introduction

With the advent of Social Networks, the spectrum of tools and techniques capable of achieving insights from the interrelations between their users has increased

H. Yin et al. (Eds.): IDEAL 2018, LNCS 11315, pp. 325–335, 2018.
https://doi.org/10.1007/978-3-030-03496-2_36

considerably in the last decade [2]. The acquired knowledge by virtue of such methods range from the quantification of the level of influence of a node within the network (*centrality*) to the discovery of shortest paths between a given pair of nodes or the derivation of enriched ways to visualize a network given the weight distribution of its edges. Many of the functionalities that Social Network users enjoy nowadays build upon this algorithmic portfolio, with a late prominence noted around other practical goals (e.g. child abuse [6,28,30] or the detection of radicalization risk [17]).

In this context, inferring communities within the nodes of a given network is arguably one of the most addressed tasks in the related literature. In this regard, a *community* refers to a set of nodes that comply with the general principles of strong intra-connectivity (namely, high degree/strength of the links among nodes belonging to the community) and weak inter-connectivity (correspondingly, low degree/strength of edges between nodes belonging to different communities). These measured parameters can be redefined as per the characteristics of the network at hand (directed, weighted, multiple edges, self loops), so that they ultimately quantify the cohesiveness of any proposed partition of the network. This evaluation is rather done based on diverse metrics proposed in related studies, each composing differently how connectivity is analyzed to yield a single quality value for the partition under different assumptions, e.g. Newman and Girvan's Modularity [21], Permanence [5], Surprise [1] and others alike [4].

Algorithmically speaking, many contributions have hitherto gravitated on the development of different heuristic approaches to find communities towards – implicitly or explicitly – optimizing one of the aforementioned metrics. This is the case, for instance, of iterative greedy methods capable of inferring a hierarchy of communities in a constructive fashion, similarly to agglomerative hierarchical clustering techniques [3]. Interestingly for the scope of this work, a growing strand of literature is currently devoted to the use of heuristic optimization algorithms directly adopting one modularity metric as their objective function. Examples abound, each focusing on assorted combinations of network instances, metric functions and algorithmic approximations. Genetic Algorithms are arguably among those more recurrently explored to date for discovering communities in networks of different characteristics [10,23,27]. However, many other solvers within the Evolutionary Computation and Swarm Intelligence fields have been also employed for this same purpose: to cite a few, Differential Evolution [16], Particle Swarm Optimization [25] or Ant Colony Optimization [14][24]. More recently, the research attention has steered towards the use of modern nature-inspired solvers for community detection in graphs, such as the Firefly Algorithm [7], Bat Algorithm [13] or Artificial Bee Colony [11], among others.

The work presented in this manuscript takes a step further over the state of the art exposed above by elaborating on several new research directions: (1) we address the problem of detecting communities in weighted directed networks, far less studied than other graph instances; (2) the adoption of the Hamming distance as a measure to compute the similarity between different partitions, which can be exploited during the search process of the overall heuristic; and (3) the assessment of these algorithmic ingredients with a diversity of nature-inspired solvers: Water Cycle Algorithm (WCA, [8]), Firefly Algorithm (FA, [31]), Evolutionary Simulated Annealing (ESA, [32]) and Population based Variable Neighborhood Search (PVNS, [29]). Thorough details on how each of these methods has been tailored to benefit from the developed operators are given, along with a justification of their expected benefits in terms of convergence. In order to assess their comparative performance, results obtained over 24 synthetically generated datasets are presented and discussed on the basis of their capability to discover their ground-of-truth partition. The significance of the performance gaps found in this benchmark is verified by two statistical tests (Friedman's and Wilcoxon), from which we conclude that WCA outperforms the rest of heuristic approaches in most of the networks.

The rest of the paper is structured as follows: in Sect. 2 the problem of finding communities in weighted directed networks is mathematically formulated, whereas the heuristic solvers are described in Sect. 3. The experimentation is introduced in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Problem Statement

We begin by modeling a weighted network as a graph $\mathcal{G} \doteq \{\mathcal{V}, \mathcal{E}, f_{\mathcal{W}}\}$, where $\mathcal{V}$ denotes the set of $|\mathcal{V}| = V$ nodes or vertices of the network, $\mathcal{E}$ correspond to the set of links or edges connecting every pair of nodes, and $f_{\mathcal{W}} : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}^+$ is a function assigning a non-negative weight to the edge connecting every pair of nodes. We assume that $f_{\mathcal{W}}(v, v) = 0$ (i.e. no self loops), and that $f_{\mathcal{W}}(v, v') = 0$ if nodes v and v' are not connected. For notational convenience we define $f_{\mathcal{W}}(v, v') \doteq w_{v,v'}$, yielding a $V \times V$ adjacency matrix $\mathbf{W}$ given by $\mathbf{W} \doteq \{w_{v,v'} : v, v' \in \mathcal{V}\}$ and fulfilling $\mathrm{Tr}(\mathbf{W}) = 0$. The directed nature of the network is ensured by overriding any assumption on the symmetry of $\mathcal{W}$, e.g. $w_{v,v'}$ is not necessarily equal to $w_{v',v}$.

With the above notation in mind, the general problem of detecting communities in a graph $\mathcal{G}$ can be conceived as the partition of the vertex set $\mathcal{V}$ into a number of disjoint, non-empty groups, each with an arbitrary size. Let M denote 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). By extending this notation, the community to which node v belongs can be denoted as $\mathcal{V}^v \in \widetilde{\mathcal{V}}$.

The weighted directed nature of the network imposes a redefinition of the conventional in-degree and out-degree to the input and output *strength* of a given node of the network, which are correspondingly given by

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

namely, as the sum of the weights of the incident (outgoing) edges to (from) node v. It is important to note that these quantities reflect both the directivity and the weighted nature of adjacency matrix $\mathbf{W}$. Thereby, they should play an important role in the definition of the communities in a similar fashion to the in- and out-degree when clustering undirected, unweighted networks. Following this rationale, a measure of the quality of a given partition $\widetilde{\mathcal{V}}$ can be formulated departing from the definition of *modularity* for undirected graphs introduced in [18,20]. By defining a binary function $\delta : \mathcal{V} \times \mathcal{V} \mapsto \{0,1\}$, such that $\delta(v,v')=1$ if $\mathcal{V}^v = \mathcal{V}^{v'}$ as per the partition set by $\widetilde{\mathcal{V}}$ (and 0 otherwise), the measure of modularity in weighted directed graphs can be computed by:

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

where $|\sum_{\mathbf{W}}|$ denotes the sum weight of all edges of the network. Finding a *good* partition $\widetilde{\mathcal{V}}^*$ of a weighted directed network $\mathcal{G}$ can be then casted as:

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

where $\mathcal{B}_V$ denotes the set of possible partitions of V elements into nonempty subsets (i.e. the solution space of the above combinatorial problem). The cardinality of this set, given by the V-th Bell number [12] is huge, thus calling for the adoption of heuristics for its efficient exploration. For example, a network instance with $V = 20$ nodes can be partitioned in approximately $517.24 \cdot 10^{12}$ different ways. Provided that the above metric could be computed in 1 microsecond on average, we would need more than one and a half years to exhaustively check all possible partitions.

3 Proposed Nature-Inspired Solvers

To efficiently solve the problem posed above, several discrete solvers are proposed. Before a deep detail of each method, common design aspects such as the encoding strategy, the solution repair method and the distance to compare different solutions are first described in what follows.

The first issue to be tackled is the encoding of solutions or individuals. In this work we adopt a label-based representation [15]: each solution is represented as a permutation $\mathbf{x} = [c_1, c_2, \ldots, c_V]$ of V integers from the range $[1, \ldots, V]$, where we recall that V represents the number of nodes in the network. The value of c_v denotes the cluster label to which node v belongs. For example, assuming a $V = 10$ network, one feasible solution could be $\mathbf{x} = [1,2,1,1,2,2,2,3,3,3]$, meaning that the partition represented by this vector is $\widetilde{\mathcal{V}} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$, where $\mathcal{V}_1 = \{1,3,4\}$, $\mathcal{V}_2 = \{2,5,6,7\}$ and $\mathcal{V}_3 = \{8,9,10\}$. In order to avoid ambiguities in the representation, a repairing procedure has been developed partly inspired from the one proposed in [9]. By applying this process to every newly produced solution, ambiguities such as those between $\mathbf{x} = [4,2,4,4,2,2,2,3,3,3]$ and $\mathbf{x} = [7,1,7,7,1,1,1,4,4,4]$ (which represent the same partition) are modified to $\mathbf{x} = [1,2,1,1,2,2,2,3,3,3]$.

Another important aspect of three of the proposed methods (namely, WCA, FA and ESA) is how the similarity between the different individuals is computed, which lies at the core of the constituent operators of such methods. To this end, the well-known Hamming distance has been selected. Some studies of the literature have previously used this function for similar purposes in other combinatorial problems [22], verifying its good performance for this purpose. Specifically, this function calculates the number of non-corresponding elements of both solutions. For example, assuming two individuals $\mathbf{x} = [1, 1, 2, \mathbf{3}, 3, \mathbf{2}, 2, 3, 3, \mathbf{1}]$ and $\mathbf{x}' = [1, 1, 2, \mathbf{2}, 3, \mathbf{3}, 2, 3, 3, \mathbf{2}]$, their Hamming Distance $D_H(\mathbf{x}, \mathbf{x}')$ would equal 3.

Furthermore, four different movement operators have been developed for evolving individuals along the search process. These functions are applied depending on the distance between two individuals (in the case of FA and WCA), or depending on the nature of the solution (in the case of ESA and PVNS). These operators are called CE_1, CE_3, CC_1 and CC_3. For each of these functions, the subscript represents the number of randomly selected nodes, which are extracted from its corresponding cluster. In CE_* operators, the taken nodes are re-inserted in already existing clusters, while in CC_* nodes can be inserted also in newly generated clusters.

WCA: This solver was first conceived for solving continuous optimization problems. For this reason, an adaptation has been made in order to address combinatorial optimization problems such as the one addressed in this paper. Along with the encoding and the measurement of distance method, the most critical aspect is how rivers and streams flow to their corresponding leading raindrop. Following the philosophy of the original 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:

$$\mathbf{x}^{p_{str}}(t+1) = \Psi\left(\mathbf{x}^{p_{str}}(t), \min\left\{V, \left\lfloor rand \cdot \theta \cdot D_H(\mathbf{x}^{p_{str}}(t), \mathbf{x}^{\lambda(p_{str})}(t))\right\rfloor\right\}\right), \quad (4)$$

where θ is a heuristic parameter, $rand$ is a continuous random variable uniformly distributed in $\mathbb{R}[0, 1]$, and $\Psi(\mathbf{x}, Z) \in \{CE_1, CE_3, CC_1, CC_3\}$, each parametrized by the number of times Z this operator is applied to the raindrop $\mathbf{x}$. The best position resulting from the Z movements performed on $\mathbf{x}$ is chosen as the output of the operator. The same philosophy is followed for the movement of a stream or a river towards the sea, simply by replacing $\mathbf{x}^{\lambda(p_{str})}(t)$ by $\mathbf{x}^{p_{sea}}(t)$.

Furthermore, in order to further enhance the exploration capacity of the technique, the *inclination* mechanism recently introduced in [22] is also used in the WCA developed in this work. Thanks to this mechanism, the method intelligently selects the proper movement function to use at each iteration for each raindrop, depending on its specific situation. Particularly, each time a raindrop is ready to perform a movement, the so-called *inclination* $\xi(\mathbf{x}, \mathbf{x}')$ is calculated, using as reference the $D_H(\mathbf{x}, \mathbf{x}')$ to its designated river/sea $\mathbf{x}'$. Specifically, $\xi(\cdot, \cdot)$ is set equal to $V/D_H(\cdot, \cdot)$. Taking into account that the bigger $D_H(\cdot, \cdot)$ is, the higher $\xi(\cdot, \cdot)$ should be, a *fast move* should be enforced with a higher probability if the inclination is high. On the other hand, 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 this research, instead of having only two different functions (as occurs in [22]), four different operators are available: CC_* functions are considered *fast moves*, whereas

CE_* are deemed *slow moves*. Finally, the philosophy of both evaporation and raining concepts remain in the same way as in the basic WCA, acting similarly to a mutation operator in Genetic Algorithms. Concretely, the raining process comprises a number R of consecutive CC_3 movements.

FA: As in the case of WCA, the classic FA cannot be either applied directly to address a discrete problem. For this reason, some modifications have been included. First, each firefly in the swarm represents a solution for the problem. The concept of light absorption is also considered for this adaptation, which is crucial for the adjustment of fireflies' attractiveness. The distance between two different fireflies is represented by the Hamming Distance. Finally, the movement of a firefly attracted to another brighter firefly is determined similarly to Expression (4). At last, and emulating the concept of *inclination* introduced for the WCA, 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.

ESA: For the sake of fairness w.r.t. the rest of considered solvers, a population-based evolutionary version of the naïve SA has been used [32]. ESA counts with the same four CC_* and CE_* movement functions, meaning that each individual has its own randomly assigned operator. Furthermore, each population element has its own temperature value drawn at random from $\mathbb{R}^+[0.7, 1.0]$ and kept fixed over the entire search process. A step forward in the adaptation of this method is done by also exploiting Eq. (4) for the individual's movement, using the distance to the best individual of the population as the reference. This way, each individual performs a number of $D_H(\cdot,\cdot)$ separated movements, from which the best one is selected. Finally, the best individual performs a random number of movements between 1 and Z.

PVNS: A population-based approach of the original VNS has been designed for this problem. Following the same philosophy considered for ESA, each individual of the population has its own main movement operator, randomly selected among CE_1, CE_3, CC_1 and CC_3. At each generation, every individual of the population performs a movement using its main operator, but it may choose a different one with probability 0.25.

4 Experimentation and Results

In order to assess the performance of the above 4 heuristic solvers, computer experiments are run over a heterogeneous set of synthetically generated network instances. Specifically, the benchmark is composed by networks with $V \in \{35, 50, 75\}$ nodes. For each network, a different number of *ground of truth* communities is modeled by first creating a partition of the network (with random sizes for its constituent groups $\{\mathcal{V}_m\}_{m=1}^{M}$), and then by connecting nodes within every group with probability p_{in} and nodes of different groups with probability p_{out}. Therefore, the *ground of truth* partition should be more discriminable if p_{in} is high and p_{out} is low. Weights $w_{v,v'}$ for every edge (v, v') have been drawn uniformly at random from ranges $\mathbb{R}[0.0, 10.0]$ (inter-community edges) and $\mathbb{R}[10.0, 20.0]$ (intra-community edges). This network construction process permits to assess the performance of the proposed solvers over *noisy* versions of a graph characterized by a controlled underlying community distribution, as opposed to the common practice by which such a comparison is made based on the attained fitness value of every technique. Finally, 15 independent runs have been executed for each dataset, with the main goal of providing statistically reliable insights on the performance of every method. In relation to the ending criterion, each run finishes when there are $V + \sum_{v=1}^{V} v = V(V + 3)/2$ iterations without improvements in the best solution found. The population size is set to 50 individuals for all cases. In the specific case of WCA, the number of rivers has been set to 9 raindrops (approximately 20% of the whole population), leading to a number of streams equal to 40. On the other hand, the maximum distance for evaporation) and R have been respectively set to 5% and a uniform random value from $\mathbb{N}[0, \lfloor 0.5V \rfloor]$.

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

	WCA				ESA				FA				PVNS			
(V, M, p_{in}, p_{out})	Avg	Best	Std	t_{conv}	Avg	Best	Std	t_{conv}	Avg	Best	Std	t_{conv}	Avg	Best	Std	t_{conv}
$(35, 4, 0.6, 0.1)$	**0.526**	0.526	0.000	0.89	0.515	0.526	0.010	1.33	0.521	0.547	0.010	1.40	**0.526**	0.526	0.000	3.6
$(35, 4, 0.9, 0.4)$	**0.876**	0.876	0.000	1.34	0.860	0.876	0.010	1.23	0.745	0.768	0.010	1.40	**0.876**	0.876	0.000	4.43
$(35, 7, 0.6, 0.1)$	**1.000**	1.000	0.000	0.98	0.972	1.000	0.010	1.30	0.900	0.929	0.010	1.91	**1.000**	1.000	0.000	3.19
$(35, 7, 0.6, 0.4)$	0.807	0.807	0.000	1.85	**0.827**	0.863	0.010	1.32	0.800	0.828	0.010	1.59	0.806	0.807	0.010	3.95
$(35, 7, 0.8, 0.1)$	**1.000**	1.000	0.000	0.75	0.997	1.000	0.010	1.28	0.927	0.949	0.010	1.66	**1.000**	1.000	0.000	3.67
$(35, 7, 0.9, 0.4)$	**1.000**	1.000	0.000	0.80	0.997	1.000	0.010	1.29	0.914	0.935	0.010	1.95	**1.000**	1.000	0.000	4.34
$(35, 18, 0.6, 0.1)$	**0.960**	0.969	0.010	2.57	0.931	0.962	0.010	1.30	0.952	0.973	0.010	1.47	0.954	0.969	0.010	3.29
$(35, 18, 0.9, 0.4)$	**0.998**	1.000	0.010	1.21	0.971	0.974	0.010	1.34	0.974	0.990	0.010	1.97	**0.998**	1.000	0.010	3.52
$(50, 5, 0.6, 0.1)$	**1.000**	1.000	0.000	1.59	0.998	1.000	0.010	2.36	0.821	0.851	0.010	3.73	**1.000**	1.000	0.000	9.13
$(50, 5, 0.6, 0.4)$	**0.694**	0.699	0.010	4.80	0.680	0.699	0.010	2.71	0.640	0.658	0.010	4.66	0.689	0.699	0.010	9.12
$(50, 5, 0.9, 0.1)$	**1.000**	1.000	0.000	1.45	0.996	1.000	0.010	2.65	0.825	0.905	0.030	3.02	**1.000**	1.000	0.000	7.51
$(50, 10, 0.7, 0.4)$	**0.972**	0.972	0.000	1.80	0.971	1.000	0.010	2.63	0.893	0.908	0.010	4.20	**0.972**	0.972	0.010	8.72
$(50, 10, 0.9, 0.4)$	**1.000**	1.000	0.000	1.35	0.989	1.000	0.010	2.67	0.941	0.962	0.010	3.74	**1.000**	1.000	0.000	8.04
$(50, 25, 0.6, 0.1)$	**0.979**	0.989	0.010	6.31	0.952	0.965	0.010	2.74	0.955	0.969	0.010	3.14	0.967	0.977	0.010	9.27
$(50, 25, 0.6, 0.4)$	**0.955**	0.968	0.010	4.94	0.942	0.961	0.010	2.77	0.944	0.961	0.010	3.27	0.947	0.968	0.010	8.55
$(50, 25, 0.9, 0.4)$	**0.990**	0.991	0.010	3.73	0.971	0.987	0.010	2.79	0.970	0.980	0.010	3.19	0.982	0.991	0.010	8.99
$(75, 8, 0.6, 0.1)$	**0.987**	1.000	0.010	6.65	0.959	1.000	0.010	4.71	0.828	0.844	0.010	8.03	0.971	1.000	0.010	22.57
$(75, 8, 0.8, 0.3)$	**1.000**	1.000	0.000	2.69	**1.000**	1.000	0.000	4.55	0.865	0.888	0.010	7.85	**1.000**	1.000	0.000	21.83
$(75, 8, 0.9, 0.4)$	**1.000**	1.000	0.000	2.25	**1.000**	1.000	0.000	4.27	0.896	0.919	0.010	9.99	**1.000**	1.000	0.000	22.72
$(75, 15, 0.6, 0.2)$	**0.986**	0.987	0.010	5.98	0.982	0.989	0.010	5.48	0.892	0.917	0.010	8.27	0.984	0.989	0.010	22.98
$(75, 30, 0.6, 0.1)$	**0.971**	0.976	0.010	12.60	0.949	0.973	0.010	5.48	0.943	0.956	0.010	8.53	0.956	0.966	0.010	18.50
$(75, 30, 0.8, 0.4)$	**0.966**	0.970	0.010	16.53	0.951	0.971	0.010	5.77	0.939	0.955	0.010	8.64	0.958	0.979	0.010	23.30
$(75, 38, 0.9, 0.1)$	**0.984**	0.993	0.010	13.27	0.972	0.981	0.010	6.28	0.972	0.979	0.010	8.42	0.973	0.981	0.010	20.78
$(75, 38, 0.9, 0.4)$	**0.985**	0.993	0.010	18.35	0.968	0.981	0.010	6.06	0.970	0.982	0.010	9.56	0.973	0.994	0.010	22.57
Friedman's non-parametric test (mean ranking)																
Rank	1.3333				3.1042				3.7292				1.8333			

In Table 1, the results (average/best/standard deviation) obtained by the four methods are displayed in terms of the Normalized Mutual Information (NMI) with respect to the *ground of truth* partition of every network. The NMI score quantifies the level of agreement between both community partitions ignoring label permutations: if $\mathrm{NMI}(\widetilde{\mathcal{V}}, \widetilde{\mathcal{V}}') = 1$ both community distributions $\widetilde{\mathcal{V}}$ and $\widetilde{\mathcal{V}}'$ are equal to each other, whereas lower values of this score denote that there are differences between them. A first inspection reveals that WCA outperforms the other methods in almost all the instances, with consistently superior average NMI scores for most networks. As expected, results degrade when the values of (p_{in}, p_{out}) imprint topological noise on the *ground of truth* partition of every network, as exemplified by instances $(V, M, p_{in}, p_{out}) = (50, 5, 0.6, 0.4)$ (for which the best partition found attains NMI $= 0.699$) and $(50, 5, 0.9, 0.1)$ (which is perfectly resolved by WCA and PVNS in all runs). The average iteration index at which every solver converges as per the aforementioned convergence criterion is also included in the table (column t_{conv}). Although ESA and FA converge faster on average (specially for large networks), their lower NMI scores when compared to PVNS and WCA belittle this computational advantage.

A Friedman's non-parametric test for multiple comparison has been carried out to resolve the statistical significance of the results (last row of the Table). The mean ranking returned by this test is displayed for each of the compared algorithms. Furthermore, the Friedman statistic obtained is 53.0125. The confidence interval has been set in 99%, being 11.34 the critical point in a χ^2 distribution with 3 degrees of freedom. Since $53.0125 > 11.34$, it can be concluded that there are significant differences among the results. Furthermore, to prove the significance between the best two techniques – namely, WCA and PVNS, a Wilcoxon Signed-Rank test has been applied. The confidence interval has been established at 99% also for this test. Regarding the difference in the obtained results, the obtained Z-value is -3.0594, with a p-value equal to 0.00222. These results support the significance of the difference at 99% confidence level. Besides that, the obtained Wilcoxon test statistic is 0. The critical value of this statistic for the datasets in which results are not at $p \leq 0.010$ is 7. Therefore, the result is significant at this confidence level, thereby concluding that the WCA approach is the best performing alternative in the designed benchmark.

5 Conclusions and Future Research Lines

In this work community detection in weighted directed graphs has been approached by using nature-inspired heuristics. To this end, the discovery of optimal partitions is formulated as an optimization problem driven by a measure of modularity adapted to accommodate the directional and weighted nature of the edges of the network. To efficiently undertake this optimization problem, different heuristic techniques have been designed and adapted to deal with the particularities of the solution space, such as the potential representational ambiguity of label encoding and the definition of distance between solutions to the problem. In addition, each designed heuristic is also modified to account for a better evolution of the individuals (partitions) found during the search process. Their performance has been compared over 24 network instances composed by

35, 50 and 75 nodes, using NMI with respect to their *ground of truth* partition as the comparison criterion. The obtained results reveal that WCA dominates the benchmark with statistically significance, specially for large networks.

In light of the promising results obtained in this research work, we plan to conduct further efforts in different directions. Additional nature-inspired, evolutionary and swarm intelligent methods will be included in the benchmark, which will also consider network instances of higher scales than the ones used in this work. Moreover, we will explore how to hybridize the aforementioned heuristic solvers with local search techniques mimicking the operation of other heuristics found in the literature, such as recently contributed message passing procedures [26] and other techniques renowned for their good scalability [19].

Acknowledgements. E. Osaba and J. Del Ser would like to thank the Basque Government for its funding support through the EMAITEK program. I. Fister Jr. and I. Fister acknowledge the financial support from the Slovenian Research Agency (Research Core Fundings No. P2-0041 and P2-0057). A. Iglesias and A. Galvez acknowledge the financial support from the projects TIN2017-89275-R (AEI/FEDER, UE), PDE-GIR (H2020, MSCA program, ref. 778035), and JU12 (SODERCAN/FEDER UE).

References

1. Aldecoa, R., Marín, I.: Deciphering network community structure by surprise. PloS ONE **6**(9), e24195 (2011)
2. Bello-Orgaz, G., Jung, J.J., Camacho, D.: Social big data: recent achievements and new challenges. Inf. Fusion **28**, 45–59 (2016)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exp. **2008**(10), P10008 (2008)
4. Chakraborty, T., Dalmia, A., Mukherjee, A., Ganguly, N.: Metrics for community analysis: a survey. ACM Comput. Surv. (CSUR) **50**(4), 54 (2017)
5. 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, pp. 1396–1405. ACM (2014)
6. Cockbain, E., Brayley, H., Laycock, G.: Exploring internal child sex trafficking networks using social network analysis. Policing: J. Policy Pract. **5**(2), 144–157 (2011)
7. 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), pp. 2233–2239. IEEE (2016)
8. Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M.: Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems. Appl. Soft Comput. **110**(111), 151–166 (2012)
9. Falkenauer, E.: Genetic Algorithms and Grouping Problems. Wiley, New York (1998)
10. Guerrero, M., Montoya, F.G., Baños, R., Alcayde, A., Gil, C.: Adaptive community detection in complex networks using genetic algorithms. Neurocomputing **266**, 101–113 (2017)

11. Hafez, A.I., Zawbaa, H.M., Hassanien, A.E., Fahmy, A.A.: Networks community detection using artificial bee colony swarm optimization. In: Kömer, P., Abraham, A., Snášel, V. (eds.) Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014. AISC, vol. 303, pp. 229–239. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08156-4_23
12. Harris, J.M., Hirst, J.L., Mossinghoff, M.J.: Combinatorics and Graph Theory, vol. 2. Springer, New York (2008). https://doi.org/10.1007/978-0-387-79711-3
13. Hassan, E.A., Hafez, A.I., Hassanien, A.E., Fahmy, A.A.: A discrete bat algorithm for the community detection problem. In: Onieva, E., Santos, I., Osaba, E., Quintián, H., Corchado, E. (eds.) HAIS 2015. LNCS (LNAI), vol. 9121, pp. 188–199. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19644-2_16
14. Honghao, C., Zuren, F., Zhigang, R.: Community detection using ant colony optimization. In: IEEE Congress on Evolutionary Computation (CEC), pp. 3072–3078. IEEE (2013)
15. Hruschka, E.R., Campello, R.J., Freitas, A.A.: A survey of evolutionary algorithms for clustering. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **39**(2), 133–155 (2009)
16. Jia, G., et al.: Community detection in social and biological networks using differential evolution. In: Hamadi, Y., Schoenauer, M. (eds.) LION 2012. LNCS, pp. 71–85. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34413-8_6
17. Lara-Cabrera, R., Pardo, A.G., Benouaret, K., Faci, N., Benslimane, D., Camacho, D.: Measuring the radicalisation risk in social networks. IEEE Access **5**, 10892–10900 (2017)
18. Leicht, E.A., Newman, M.E.: Community structure in directed networks. Phys. Rev. Lett. **100**(11), 118703 (2008)
19. Lu, H., Halappanavar, M., Kalyanaraman, A.: Parallel heuristics for scalable community detection. Parallel Comput. **47**, 19–37 (2015)
20. Newman, M.E.: Analysis of weighted networks. Phys. Rev. E **70**(5), 056131 (2004)
21. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
22. 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. Appl. Soft Comput. **71**, 277–290 (2018)
23. Pizzuti, C.: GA-Net: a genetic algorithm for community detection in social networks. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87700-4_107
24. Pizzuti, C.: Evolutionary computation for community detection in networks: a review. IEEE Trans. Evol. Comput. **22**(3), 464–483 (2018)
25. Rahimi, S., Abdollahpouri, A., Moradi, P.: A multi-objective particle swarm optimization algorithm for community detection in complex networks. Swarm Evol. Comput. **39**, 297–309 (2018)
26. Shi, C., Liu, Y., Zhang, P.: Weighted community detection and data clustering using message passing. J. Stat. Mech.: Theory Exp. **2018**(3), 033405 (2018)
27. Tasgin, M., Herdagdelen, A., Bingol, H.: Community detection in complex networks using genetic algorithms. arXiv preprint arXiv:0711.0491 (2007)
28. 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. Concurr. Comput.: Pract. Exp. **28**(4), 1385–1395 (2016)

29. Wang, X., Tang, L.: A population-based variable neighborhood search for the single machine total weighted tardiness problem. Comput. Oper. Res. **36**(6), 2105–2110 (2009)
30. Westlake, B.G., Bouchard, M.: Liking and hyperlinking: community detection in online child sexual exploitation networks. Soc. Sci. Res. **59**, 23–36 (2016)
31. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimisation. Int. J. Bio-Inspir. Comput. **2**(2), 78–84 (2010)
32. Yip, P.P., Pao, Y.H.: Combinatorial optimization with use of guided evolutionary simulated annealing. IEEE Trans. Neural Netw. **6**(2), 290–295 (1995)