



Random search immune algorithm for community detection

Antonio G. Spampinato¹ · Rocco A. Scollo¹ · Vincenzo Cutello¹ · Mario Pavone¹

Accepted: 22 February 2023 / Published online: 3 April 2023
© The Author(s) 2023

Abstract

Community detection is a prominent research topic in Complex Network Analysis, and it constitutes an important research field on all those areas where complex networks represent a powerful interpretation tool for describing and understanding systems involved in neuroscience, biology, social science, economy, and many others. A challenging approach to uncover the community structure in complex network, and then revealing the internal organization of nodes, is *Modularity optimization*. In this research paper, we present an immune optimization algorithm (OPT- IA) developed to detect community structures, with the main aim to maximize the modularity produced by the discovered communities. In order to assess the performance of OPT- IA, we compared it with an overall of 20 heuristics and metaheuristics, among which one Hyper-Heuristic method, using social and biological complex networks as data set. Unlike these algorithms, OPT- IA is entirely based on a fully random search process, which in turn is combined with purely stochastic operators. According to the obtained outcomes, OPT- IA shows strictly better performances than almost all heuristics and metaheuristics to which it was compared; whilst it turns out to be comparable with the Hyper-Heuristic method. Overall, it can be claimed that OPT- IA, even if driven by a purely random process, proves to be reliable and with efficient performance. Furthermore, to prove the latter claim, a sensitivity analysis of the functionality was conducted, using the classic metrics *NMI*, *ARI* and *NVI*.

Keywords Metaheuristics · Community detection · Immune-inspired algorithms · Random search algorithms · Combinatorial optimization · Complex networks · Modularity optimization

1 Introduction

In the modern interdisciplinary sciences, the complex networks are a powerful interpretation tool useful for the analysis and representation of a wide number of real-world systems and are widely involved in many areas, such as for instance neuroscience, biology, social sciences, economics, and physics. With this graph-based model it is possible to represent connections and interactions of the underlying entities, where vertices are the elementary parts of the real systems, whilst edges represent their mutual interactions (Newman 2003; Boccaletti et al. 2006). Complex networks may contain specific groups of highly interconnected vertices organized in compartments or structure, where each of them has a role and/or a function that satisfy a specific property of cohesion. In terms of graph theory, compartments are represented by partitions of the set of nodes with high internal links density,

called *communities* or *modules*, which are loosely associated with other groups (Girvan and Newman 2002; Fortunato 2010).

Finding compartments in a graph-theoretic context is a fundamental issue in the study of network systems, in which often they exhibit significantly different functions and, therefore, a global analysis of the network would be inappropriate and impractical. A detailed analysis of individual communities, instead, may shed some light on the organization of systems and leads to more significant insights into the roles of individuals. This approach can also allow the visualization and analysis of large and complex networks focused on a new higher-level structure, in which each identified community can be compressed in a node belonging to the latter. It is important to emphasize that classical algorithms for graph clustering are not suitable for revealing the properties of community's structures, as they are mainly based on optimal subdivisions of graph in order to guarantee a minimum flow cut. On the other hand, finding the properties of a community's structure requires a complex analysis on the linking models and relationships. Community Detection (CD) has

✉ Mario Pavone
mpavone@dmi.unict.it

¹ Department of Mathematics and Computer Science,
University of Catania, V.le A. Doria 6, 95125 Catania, Italy

been shown meaningful structure in networks which identifies groups of nodes within which connections are denser than between them, in order to provide very important information for detecting structural and functional relations between objects, such as functional modules in protein-protein interaction networks, groups in online and contact-based social networks, etc. Fortunato (2010) In the last decade, a growing number of new methods and computational approaches have been developed, and several different metrics for community structure evaluation have been introduced, which are used for community detection in graphs (Porter et al. 2009; Newman 2012; Coscia et al. 2011). The most used and popular method for community detection is to maximize the *Modularity* of a complex network (Newman and Girvan 2004; Newman 2006, 2004), which measures the strength of the structure of the communities detected: high modularity denotes dense connections between the vertices inside a community, but sparse among the nodes belonging to different modules. Basically, it measures the difference between the actual fraction of edges within the community and the expected fraction in a random graph with the same number of nodes and the same degree sequence.

In this research work, an immune algorithm, called OPT-IA (Cutello et al. 2020; Stefano et al. 2016; Pavone et al. 2012), is presented, which has been designed for detecting communities in complex networks. The proposed OPT-IA algorithm is a population-based metaheuristic that takes inspiration from the dynamics and principles of the immune system, and successfully applied in several optimization problems (Cutello et al. 2019; Vitale et al. 2018; Stracquandano et al. 2015). Three immunological operators are the driving-force of OPT-IA - (i) *cloning*, (ii) *hypermutation*, and (iii) *stochastic aging*—which help the algorithm in performing a careful exploration of the search space, avoid getting trapped in local optima, and properly exploiting all information learned during the evolution. It is important to stress that all operators work in a purely random way without any deterministic guide to refine and improve the solutions neither taking advantage from features of the network. However, from the obtained outcomes, the combination of these random operators allows OPT-IA to be efficient and reliable. Many computational experiments have been performed in order to evaluate the efficiency and reliability of the proposed algorithm in community detection, and different types of complex networks have been taken into account, from social to biological ones, included many synthetic networks (around 80) in order to evaluate the algorithm in different search scenarios. Overall, then, OPT-IA was tested on a data set of 103 instances, from small networks ($|V| = 28$) to larger ones ($|V| = 3000$), and a comparative analysis with 20 different optimization algorithms was also performed in order to evaluate OPT-IA's robustness. Furthermore, an analysis on the computational time of OPT-IA has been performed using

the *Time-To-Target plots* (Aiex et al. 2002; Feo et al. 1994), which are a standard graphical methodology for characterizing the running time of stochastic algorithms, comparing the empirical and theoretical distributions. Although on the one hand OPT-IA needs more iterations with respect to the compared algorithms, but always within acceptable times, on the other hand it shows optimal performances relying basically only on random search, and without using any deterministic approach. In addition to the *TTT - plots* computational time analysis, a study on the asymptotic computational complexity of OPT-IA has been conducted, as well, from which is possible to claim that its upper bound running time is $\mathcal{O}(n^3)$.

Finally, from the obtained outcomes it is possible to assert that OPT-IA finds almost always the best modularity value, strictly outperforming most of the compared algorithms. Indeed, analysing the comparisons through a ranking, ordered from the best result to the worst one, it is possible to assert that OPT-IA is always in one of the first two positions and often in the first one. Also, on the biological networks, OPT-IA finds considerably better modularity than the compared algorithms, especially with respect to *HDSA* algorithm (Civicioglu 2012), which is a Hyper-Heuristic, and therefore based by definition on the use of different heuristics. On the other hand, it is also important to note that due to the randomness underlying the immunological operators that guide the search process, and therefore to the lack of any deterministic improvement/refinement approach, OPT-IA needs a greater number of iterations to find acceptable solutions on the larger biological networks: the more the network size grows, the more generations the algorithm needs. To confirm the robustness and reliability of OPT-IA, also a functional sensitivity analysis was conducted on several synthetic networks generated by the *LFR* algorithm (Lancichinetti et al. 2008; Lancichinetti and Fortunato 2009), and using the well-known community structure similarity metrics, such as: *NMI*—*Normalized Mutual Information* (Danon et al. 2005) (mostly used in community detection); *ARI*—*Adjusted Rand Index* (Hubert and Arabic 1985); and *NVI*—*Normalized Variation of Information* (Meilă 2007).

The rest of the paper is structured as follows: the community detection problem and modularity measure are presented and formalized in Sect. 2. The description concerning the proposed algorithm and modularity maximization approach is given in Sect. 3. The description of the experiments conducted on social and biological networks is given in Sect. 4. In this section, the experimental protocol, parameters tuning, convergence and learning analysis, as well as the computational complexity (*TTT - plots* and asymptotic analysis), are also inspected and described. The effectiveness of the Precompetition operator is also tested and inspected in this section. Comparative analysis on the results of OPT-IA against those obtained by the past published algorithms is reported in Sect. 5. Further, in this section a functional sensi-

tivity analysis is also presented using NMI , ARI and NVI as community structure similarity metrics. Finally, the conclusions are presented in Sect. 6.

2 Community detection and modularity maximization

Community, in a complex network level, represents a group of nodes sharing common and similar properties in which the obtained subgraph has a certain number of edges linking a number of vertices of the community that are closer each other, where edges inside community must be in greater numbers within the community than those that connect community vertices with the rest of the network (Porter et al. 2009; Fortunato 2010; Newman 2012). The aim of CD in graphs is to identify the modules and their hierarchical organization, by using only the information encoded in the graph topology. In particular, it refers to the division of the nodes of a network into groups such that connections are dense within groups but sparser between them. In other words, a cluster corresponds to a set of nodes with more edges inside the set than to the rest of the graph. Although not all networks support such divisions, the existence of good divisions is often taken as evidence of underlying structure or possible interactive behaviours, making CD a useful tool to understand how complex networks are structured and work.

CD problem gained the attention of scientist communities to bring valuable explanations to complex networks analysis. For example, in biology, applying graph clustering method on relations among genes or proteins, modelled by networks (Protein-Protein Interaction Network) is possible thanks to group proteins having the same specific patterns and mechanisms operating within the cell (Chen and Yuan 2006), or through analysis of the network produced by neuron interactions, understanding the functional architecture of the brain (Deco and Corbetta 2011). In the same way it is possible to identify, in information networks, clusters of web pages that share some common topics and similarity in a given social network to find individuals with common interests or friendship. A plethora of diverse algorithms and techniques have been proposed for detection of the communities in real-world networks. They differ from one to the other in criteria implementations for solving CD problem. They differ also in defining criteria of identification of communities. These approaches have been applied successfully in different domains of applications and in the many real-world areas (such as biological, chemical, ecological, economic, political, social, etc.).

The OPT-IA proposed was developed for the resolution of CD problem through modularity maximization, the most popular and widely accepted method for the community

detection. We begin by describing Modularity as a measure the quality of a partitioning of a graph into communities.

2.1 Modularity

Modularity proposed by Newman and Girvan (2004) is a benefit function that measures the quality of a particular partitioning of a graph into communities. Originally defined for undirected graphs has been subsequently extended to directed and weighted graphs (Newman 2004; Mucha et al. 2009; Bickel and Chen 2009). The modularity of a partition is a scalar value (where the maximum value can be 1) used to evaluate the density of links within communities with respect to links between communities (Girvan and Newman 2002; Newman 2006). A larger positive value of modularity indicates better community structure.

Modularity maximization is one of the most popular and most widely used methods for community partition. It detects communities by searching over possible partitions of a graph, over which modularity is maximized. In a given subgraph, the modularity function is defined as the difference between the actual density of edges inside the subgraph and the expected density of such edges if the graph was randomly conditioned on its degree distribution (Newman and Girvan 2004). This expected edge density depends on the chosen null model, a random copy of the original graph that maintains the structural properties but not those on the structure of the communities. The idea behind modularity is that a network with inherent community structure usually deviates from random graphs, or rather random graph does not have a community structure. Therefore, the edge density of a subgraph should be greater than the expected density of a subgraph whose nodes are randomly connected.

Given a graph $G = (V, E)$ with $|E| = m$, and given a partition of G with N_C clusters, the benefit function of modularity can be written as:

$$Q = \sum_{c=1}^{N_C} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right] \quad (1)$$

where, for each cluster, i.e. subgraphs c ,

- l_c is the total number of edges and
- d_c is the sum of the degrees of its vertices,
- $\left(\frac{l_c}{m}\right)$ represents the fraction of edges inside a certain cluster and
- $\left(\frac{d_c}{2m}\right)^2$ the fraction of the expected edges if the graph was random (null model).

Although an important resolution limit of the measure of modularity has been underlined by Fortunato and Barthelemy

(2007), modularity seems to be a useful measure of the community structures. In fact, algorithms that search for graph partitions that offer optimal modularity are already proposed and generally assert to be able to successfully find communities in very large and complex networks (Newman 2006, 2004).

Modularity defined in a heuristic way considers a good division the one that places most of the edges of a network within groups and only some of them between groups, in which the relationships of the members among themselves in the communities must be maximized and the relationships of these members with members of other communities must be minimized.

High values of modularity indicate better community structure. We desire a quality function Q which, given a network and a candidate division of that network into groups, assigns a score to each partition of a graph, in order to classify the partitions and evaluate when one partition is better than another (in a graph the partition corresponding to its maximum value should be the best, or at least a very good one). The maximization of modularity is therefore sought after at all costs.

Obviously, a brute force search to optimize Q is impossible above all large and complex graph structures, due to the enormous number of ways in which it is possible to partition a graph. Moreover, it has been proven that the problem of determining communities by using modularity optimization is an NP-complete problem (Brandes et al. 2007), so it is highly unlikely to perform the optimization task and find an optimal solution in polynomial time with respect to the dimension of the graph. Several algorithms for community detection in complex networks have been developed and have yielded satisfactory results in some cases, but not in all situations (the performance of many methods available in large complex networks is far below expectations). Therefore, we need to move to approximate optimization methods, which can find fairly good approximations of maximum modularity in a reasonable time especially when an exhaustive brute force search for the optimal solution is unfeasible, in a very large solution space.

As described in detail in the section below, in our study and in the development of the proposed algorithm, the function provided in Eq. 1 was used as a fitness function on the methods of modularity maximization for the resolution of the CD problem for detecting communities within real-world networks.

3 OPT- IA: an immune algorithm for community detection

Immune-inspired computation nowadays represents a large and established family of successful algorithms that take

inspiration from the mechanisms and dynamics of the immune system with which it protects the living organisms. What makes the immune system source of inspiration from an algorithmic perspective is its ability in detect, recognize, and distinguish entities own to the organism from foreign ones, together with its ability to learn new information and remember those foreign entities already recognized. Three principal theories are at the basis of the immune-inspired algorithms: (1) clonal selection (Pavone et al. 2012; Scollo et al. 2021); (2) negative selection (Fouladvand et al. 2017; Poggiolini and Engelbrecht 2013); and (3) immune networks (Smith and Timmis 2008). Among these, what has proven to be quite efficient is the one based on the clonal selection principle (called *Clonal Selection Algorithms*—CSA) (Cutello et al. 2007, 2010) mostly in search and optimization applications.

The proposed immune algorithm, OPT- IA, belongs then to this last class of algorithms, and is based on three main immune operators: (i) *static cloning*, whose aim is to generate a new population based on the highest fitness values; (ii) *hypermutation*, which explores the neighbourhood of each point of the search space; and (iii) *stochastic aging*, which removes solutions from the current population via a stochastic law, helping then OPT- IA in escaping from local optima. In addition to these, some diversification strategies have been also designed, whose aim is to keep high and proper diversity into the population, and to perform an appropriate exploration of the search space. The OPT- IA algorithm is based on two main concepts, following the biological metaphor: the antigen (Ag), which represents the problem to be solved, and the antibody (Ab), or B cell that is instead a solution for the problem to be solved.

At each timestep t , OPT- IA maintains a population of size d of B cells ($P^{(t)}$), and each B cell Ab represents a subdivision of the vertices of the graph $G = (V, E)$ in communities. In details, if n is the cardinality of the set of vertices V , a B cell $\vec{x} = \{x_1, \dots, x_n\}$ will be a sequence of n integers, between 1 and n , where $x_i = j$ indicates that the vertex i belongs to the community j . A description of OPT- IA is summarized in the pseudocode shown in Algorithm 1. The proposed algorithm takes as input: the network from which to detect the communities (G); population size (d); number of copies to be generated for each B cell (dup); the mutation rate (M); probability that an element will be removed from the population by the aging operator (P_{die}), and maximum number of generations allowed (T_{max}). It returns as output the communities detected and the relative community number, as well as *Best*, *Mean*, *Worst* and standard deviation (*StD*) values used for the comparisons.

As a first step, i.e. at the timestep $t = 0$, OPT- IA randomly generates d solutions using the uniform distribution, creating then the initial population $P^{(t=0)}$ (line 2 of Algorithm 1): any vertex is assigned to a community, randomly chosen in the

range $[1, n]$, with $n = |V|$. In this way, many communities with few assigned vertices will be generated; it will be the task to of the developed hypermutation operator (see Sect. 3.2) to compact the communities.

Algorithm 1 OPT-IA $(G, d, dup, M, P_{die}, T_{max})$

- 1: $t \leftarrow 0$
 - 2: $P^{(t)} \leftarrow \text{Initialize_Population}(d)$;
 - 3: $\text{Compute_Fitness}(P^{(t)})$
 - 4: **repeat**
 - 5: $P^{(clo)} \leftarrow \text{Cloning}(P^{(t)}, dup)$;
 - 6: $P^{(hyp)} \leftarrow \text{Hypermutation}(P^{(clo)}, M)$;
 - 7: $\text{Compute_Fitness}(P^{(hyp)})$
 - 8: $P^{(a)} \leftarrow \text{Stochastic_Aging}(P^{(t)}, P_{die})$;
 - 9: $P^{(pre)} \leftarrow \text{Precompetition}(P^{(a)})$;
 - 10: $P^{(t+1)} \leftarrow (\mu + \lambda)\text{-Selection}(P^{(pre)}, P^{(hyp)})$
 - 11: $t \leftarrow t + 1$
 - 12: **until** (termination criterion is satisfied)
-

Once the population is initialized, the next step is to evaluate the fitness function for each B cell $\vec{x} \in P^{(t)}$ by the function $\text{Compute_Fitness}(P^{(t)})$ (line 3 of Algorithm 1). In this research work, for each B cell \vec{x} , such function simply computes the value given by Eq. 1.

After the initialization of the population, and the computation of the fitness of each generated solution, the artificial evolution process begins, where the key operators take place. As in any evolutionary algorithms, OPT- IA will end its evolution process once a termination criterion is reached, which has been fixed in our experiments to a maximum number of generations allowed (T_{max}).

3.1 The cloning operator

The first immune operator to be performed is the cloning operator (line 5 of Algorithm 1), which has the main goal of producing a new population with higher affinities (i.e. fitness values), and together with the hypermutation perform careful local search. Just as it happens in nature that all those cells able to better recognize foreign entities will generate more copies of them, the cloning operator duplicates all those solutions that seems to be promising: simply it copies/clones dup times each element of the population, creating a new intermediate population $P^{(clo)}$ of dimensions $d \times dup$. It was developed a static version of the cloning operator, unlike what really happens in biology,¹ because this last shows the disadvantage to guide easily and quickly the algorithm towards local optima. Furthermore, to avoid a premature convergence, we also made dup independent from

¹ In nature, the number of clones produced for a B cell is proportional to its ability in detect and recognize the Ag.

the fitness function value of the B cell. In a nutshell, if we had chosen to increase the number of clones for high fitness elements, we would have achieved quickly a very homogeneous population, causing in turn a poor exploration of the search space.

3.2 The hypermutation operator

The hypermutation operator (line 6 of Algorithm 1) acts on each element of the population $P^{(clo)}$ performing M mutations with the main aim to explore the search space, and the neighbourhood of all solutions found so far. Similarly to the parameter dup , also the mutation rate M is a user-defined parameter and is not related to the fitness function of the solution, to avoid possible premature convergence. Importantly, unlike classical evolutionary algorithms, no mutation probability was considered. Furthermore, the introduction of blind mutations produces individuals with higher affinity (i.e. higher fitness function values), which will be then selected to form improved mature progenies. In this research work, different types of mutation operators have been developed, which can act on a single vertex in the solution, like a local operator, or on a group of nodes, like a global operator: (i) *Equiprobability*; (ii) *Destroy*; and (iii) *Fuse* operators.

Equiprobability operator

This mutation operator is locally applied and tries to find a better neighbour not still explored. Simply, it randomly selects a vertex i , and a community c_j among those existing at that moment, and, of course, different from the one to which i belongs (i.e. $c_i \neq c_j$), and then, the vertex i is moved into the community c_j .

Destroy operator

This operator works through a more global perspective than the first operator, as it acts directly on the communities rather than the single nodes. It is carried out as follows: two different communities are randomly selected, c_{out} and c_{in} , which are, respectively, the community from where the vertices will be moved; and the one that will receive such vertices. In particular, c_{out} is selected among the currently existing communities, whilst c_{in} is randomly assigned a value in the range $[1, n]$, with $n = |V|$. Note that this means that c_{in} could have a value that does not correspond to any existing community. After that, a probability is randomly chosen between 1% and 50%, and based on this probability every vertex in c_{out} will move into c_{in} . As already said, if c_{in} is among the existing ones, then the new vertices will increase the community itself; otherwise, a new community is created, and added to the others.

Fuse operator

This last operator has the main aim to try to reduce and compact the communities. Thus, it chooses randomly two

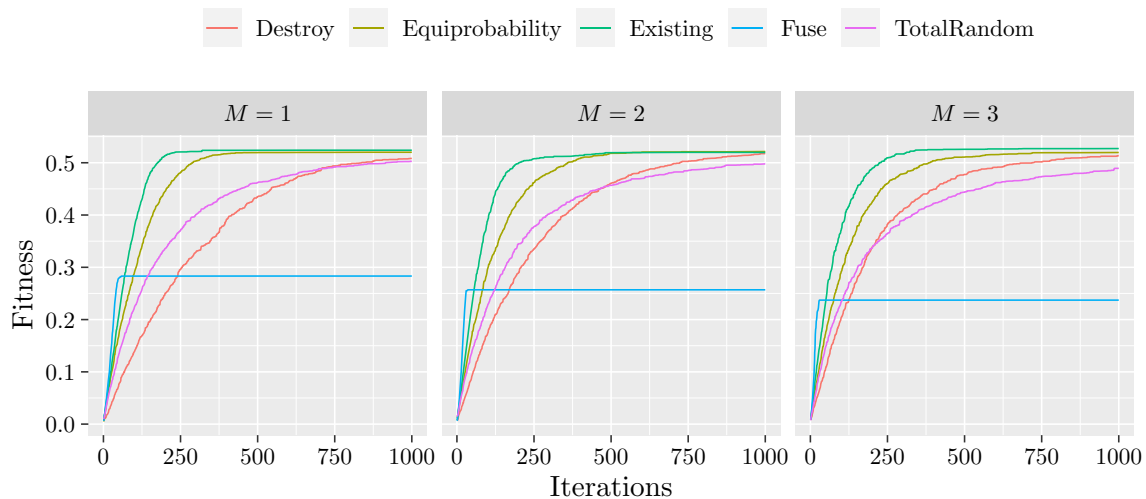


Fig. 1 A comparative analysis on the performances of some mutation operators (Spampinato et al. 2019)

different communities c_i and c_j and moves all vertices belonging to c_i into the community c_j , decreasing consequently by one the number of communities.

The effectiveness and efficiency of these three operators have already been proven in Spampinato et al. (2019) through a comparative analysis (Fig. 1). In particular, the first two have proven useful in improving the modularity function value, and in escaping to local optima; the fuse operator, instead, albeit doesn't offer good results individually (see Fig. 1), it also has been taken into account since its goal is to aggregate communities. Trying then to take advantage on the working of each operator, the Equiprobability and Destroy operators are applied with the same probability, whilst the last with low probability for the above reasons (around 49.5%, 49.5%, and 1%).

Finally, once the hypermutation was performed, and then modify all the cloned solutions, for each mutated B cell is computed its fitness value through the function $Compute_Fitness(P^{(hyp)})$.

3.3 Aging, precompetition, and selection operators

The aging operator has the main goal of helping the algorithm in jumping out of local optima and keep a high diversity inside the population. In this research work, the stochastic aging operator (line 8 of Algorithm 1) has been developed that guides OPT-IA to reduce premature convergences as much as possible. In details, the elements of the population are removed at each iteration with a P_{die} probability, which is a user-defined parameter. Diversification of solutions in a population is a crucial feature to avoid getting trapped into local optima. However, this can also become a limitation in carrying out a careful, and accurate exploration of their neighbourhoods, which also plays a key role in the success of

the algorithm. Thus, in order to have a right balance between these two key features, the stochastic aging is applied only on the $P^{(t)}$ population. In this way, diversity will be introduced in the population of the best current solutions, which will then compete with their offspring in generating the new population of the next generation, whilst the B cells in $P^{(hyp)}$ will have the task of properly exploring the neighbourhoods.

After the aging operator, to strengthen heterogeneity in the population $P^{(t)}$, a precompetition step (line 9 of Algorithm 1) has been developed, which simply randomly selects two different B cells from $P^{(t)}$: if the two solutions, although different, have the same number of community, then the lower fitness one will be removed from $P^{(t)}$ with a 50% probability. This strategy allows, thus, to maintain a more heterogeneous population during the evolutionary cycle, maintaining solutions with different number of community, in order to better explore the search space.

The last operator before the end of the iteration is the creation of the new population for the next timestep $t + 1$. In OPT-IA, the $(\mu + \lambda)$ -selection operator (line 10 of Algorithm 1) has been developed, which selects the best d B cells from the two populations $P^{(pre)}$ and $P^{(hyp)}$, without fitness repetitions. This selection operator, with $\mu \leq d$ and $\lambda = (d \times dup)$, identifies the best $\mu = d$ elements from the offspring set ($P^{(hyp)}$), and old parent B cells ($P^{(pre)}$), therefore ensuring monotonicity in the evolution dynamics. If two selected elements have the same fitness, then only one of them will be chosen randomly.

4 Benchmarks and behaviour analysis

In this section, we study the features of OPT-IA and analyse its behaviour in order to prove its efficiency and reliability. We start by describing the data sets, i.e. the complex net-

Table 1 Social and Biological Network instances

Name	Type	V	E	Δ (%)
Grevy's Zebras	<i>Social-ecological</i>	28	111	29.37
Zachary's Karate Club	<i>Social</i>	34	78	13.90
Bottlenose Dolphins	<i>Social-ecological</i>	62	159	8.41
Books about US Politics	<i>Social</i>	105	441	8.08
American College Football	<i>Social</i>	115	613	9.35
Jazz Musicians	<i>Social</i>	198	2742	14.06
Cattle PPI	<i>Biological</i>	268	303	0.85
<i>E. coli</i> TRN	<i>Biological</i>	418	519	0.60
<i>C. elegans</i> MRN	<i>Biological</i>	453	2025	1.98
<i>Helicobacter pylori</i> PPI	<i>Biological</i>	724	1403	0.54
<i>E. coli</i> MRN	<i>Biological</i>	1039	4741	0.88
<i>S. cerevisiae</i> PPI (1)	<i>Biological</i>	2018	2705	0.13
<i>S. cerevisiae</i> PPI (2)	<i>Biological</i>	2284	6646	0.25

works used for our studies and comparisons, along with the experimental protocol adopted for all the performed tests. Likewise, we show the experimental tuning on the population size, mutation rate and aging probability. Then, once the best setting of OPT-IA's parameters is determined, we show its dynamics and learning abilities. Finally, we present the analysis of the time complexity of OPT-IA via the well-known Time-To-Target plots, which are a classical methodology for the running time analysis for any stochastic algorithm. Experimental results and comparisons are shown in Sect. 5.

4.1 Data sets and experimental protocol

The OPT-IA algorithm, presented in Sect. 3, has been tested and studied on eight different real-world networks, which include three biological, and five social networks. These networks are related to two different areas and are obtained from real-world systems. Features and size of each network are detailed in Table 1.

More in details, we considered three social networks: a small size network, called *Zachary's Karate Club*, and two larger ones, respectively, called *Books about US politics* and *American college football* (Krebs 2008). The Zachary's Karate Club network represents the friendships between members of a university's karate club in the US over a period of 2 years (Zachary 1977). It has come to be a standard test network for clustering algorithms. Each node represents a member of the club, and each edge represents the relationship between the two corresponding members of the club. The network called American college football was presented in Girvan and Newman (2002) and represents the football match schedule for the 2000 season. Vertices in the graph represent the teams, whilst edges represent regular season games between the two corresponding teams. In the American Political Books network, the vertices represent the books on the

American politics purchased from amazon.com, whilst edges connect pairs of books which are frequently co-purchased. The books in this network, compiled by V. Krebs, were classified by Newman (2012) into liberal or conservative categories, with the exception of a small number of books without a clear ideological bias.

We also took into consideration two different types of networks, which identify social-ecological networks, called respectively *Bottlenose dolphins* (Lusseau et al. 2003) and *Grevy's Zebras* (Sundaressan et al. 2007). Bottlenose dolphins are aquatic mammals in the genus *Tursiops*. The network is built from a community of bottlenose dolphin community living in New Zealand and observed between 1994 and 2001. Edges denotes frequent association. In the Grevy's zebra network, edges represent the interactions between two *Equus grevyi* (node in the network) if it existed between them during the study.

Finally, six biological networks were also considered, namely: *E. coli transcriptional regulatory network* (Shen-Orr et al. 2002), *C. elegans metabolic reaction network* (Duch and Arenas 2005), *Cattle protein-protein interaction network* (Cattle 2015), *Helicobacter pylori protein-protein interaction network* (Xenarios et al. 2000; Rain et al. 2001), *E. coli metabolic reaction network* (Schellenberger et al. 2010), and *S. cerevisiae protein-protein interaction network* (1) and (2) (Yu et al. 2008; Bu et al. 2003). In particular, in the gene expression *E. coli* regulation network, which is a commonly used benchmark, the vertices represent operons, i.e. functioning units of DNA containing a cluster of genes, and edges are directed from a gene that encodes a transcription factor to a gene that it directly regulates it Shen-Orr et al. (2002).

Overall, all these networks, social and biological data in real work systems, are well-known and commonly used dataset for evaluating the efficacy and efficiency of designed algorithms for the community detection problem.

All the performed experiments were carried out on 30 independent runs, whilst the considered stopping criterion was fixed at a maximum number of generations allowed ($T_{max} = 4 \times 10^3$). Since OPT-IA is basically a blind search algorithm, it follows that without knowledge about the domain and without inclusion of any deterministic refinement approach, obviously OPT-IA must evolve for more generations than any hybrid/memetic approach or a hyper-heuristic. Nevertheless, the computational time of OPT-IA for reaching the overall best solution is, however, acceptable, as it is shown in Sect. 4.4.

Finally, for each experiment, and then for each network, the following outcomes were computed and shown: *best* modularity found on all runs; *mean* value of the best solutions found per each run; *worst* modularity value found on the overall; standard deviation (*StD*); and number of communities discovered (N_C).

4.2 Parameters tuning

As described in Sect. 3 (Algorithm 1), the crucial parameters that affect the performances of OPT-IA are, respectively: (i) the population size (d); (ii) the duplication factor (dup); (iii) the mutation rate (M); and (iv) the probability to remove a B cell at each iteration (P_{die}). Therefore, we need to find the best values for these parameters.

In a preliminary work (Spampinato et al. 2019), a small and sparse network (*almost_lattice* network with 64 nodes) was used to find the best setting for the parameter values. The network was chosen since it shows a particular complex landscape. From this preliminary study, the best parameter combinations obtained were, respectively, $d = 8$, and $dup = 4$, whilst mutation rate M varied between 1 to 3. The efficacy of such a setting was also validated by the comparison of OPT-IA with the well-known greedy optimization method LOUVAIN (Blondel et al. 2008), one of the most popular algorithms for the community detection. The comparison, which for convenience is reported in Table 3 (Sect. 5), shows that the proposed algorithm outperforms LOUVAIN in all networks tested.

Following these very good results, we ran OPT-IA on all the networks which are shown in Table 1 using the same parameter configuration. Although the algorithm was able to find the optimal solutions for some of the social networks, on the larger ones, such as for instance, on the biological networks, we obtained, instead, very poor results. In the light of this, we performed a new study on the parameter tuning, but this time we took into account the *Cattle PPI* biological network as testbed, which is a large and sparse graph, and consequently a hard enough testbed. It is important to highlight, also, that all experiments we conducted for the parameters tuning were performed on 30 independent runs, so to have more robust and reliable outcomes.

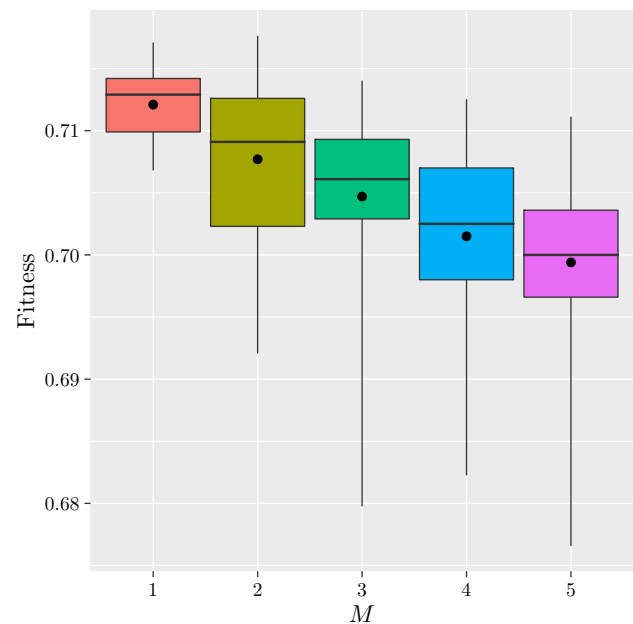


Fig. 2 Tuning the mutation rate M on the *Cattle PPI* network

Based on the knowledge acquired on OPT-IA in previous research works (Pavone et al. 2012; Cutello et al. 2020, 2019), the population size was set to $d = 100$, since it is mainly related to the dimension and complexity of the problem tackled. For all the other parameters, instead, the tuning was determined by evaluating the fitness trend at their different values. The first step of this study was conducted on the mutation rate parameter with five different values ($M = \{1, 2, 3, 4, 5\}$), using the following parameter configuration settings: population size $d = 100$, duplication parameter $dup = 4$, probability of random aging operator $P_{die} = 0.02$ and $T_{max} = 4000$.

The outcomes of these experiments are reported in Fig. 2 where we can see the distribution of fitness values over 30 independent runs at the varying of M on the *Cattle PPI* network. By observing the graph in the figure, it is possible to assert that by increasing the number of mutations the performance of OPT-IA decreases considerably with respect to both the best modularity found and the mean. The best performances are obtained using small mutation rate values, that is $M = \{1, 2\}$. It is important to note that although for $M = 1$ OPT-IA reaches a better median, the performances for $M = 1$ and $M = 2$ are, however, equivalent when compared to the best value found. For this reason, both these values for M have been taken into account for the next experiments. The good results obtained when performing a lower number of mutations are primarily due to the effect of the aging and pre-competition operators, which produce good heterogeneity, and consequently require the perturbation operator to carry out the search in the neighbourhood of the current solutions.

Once the mutation rate $M = \{1, 2\}$ was set, we performed a second step of parameter tuning to determine the best combination of (dup, P_{die}) . In particular, the duplication parameter was varied in the set $\{2, \dots, 10\}$, whilst P_{die} in $\{0, 0.01, 0.02, 0.05, 0.15, 0.3, 0.5\}$.

The results of these experiments are reported in Figs. 3 and 4.

From both figures, it is clear that better performances are obtained when increasing the value of dup . In particular, higher modularity values are obtained for $dup = \{8, 9, 10\}$ for both M values. In other words, having more copies of each solution helps OPT- IA in carrying out a careful and more accurate search in its neighbourhood. Focusing further the analysis only on these last three values, it is possible to assert that for $dup = 9$ and $dup = 10$, OPT- IA finds the best modularity more often than for $dup = 8$.

Inspecting now the plots from the perspective of parameter P_{die} and considering these last two values for dup , we can see that the better performances of OPT- IA were obtained for $P_{die} = 0.02$. With this value, indeed, the algorithm was able to find a better mean, and a lower standard deviation. Each candidate solution will have a 0.02 probability to be removed from the population and such a low probability value is enough to produce a good heterogeneity in the population (when, of course, combined with the precompetition operator). From these last experiments, it also emerges that for $M = 1$ the performances of OPT- IA are considerably better than for $M = 2$, since the algorithm reaches the best solution more often, with a better mean and standard deviation, proving in turn greater robustness and soundness.

In conclusion, from the overall experimental analysis, the best obtained parameter combination is the following: $d = 100, dup = \{9, 10\}; M = 1$ and $P_{die} = 0.02$. Note that, although $dup = 10$ showed slightly better performances on the considered network, we took into consideration both values, because their effect is also related to the network density (see results in Sect. 5).

4.3 Convergence behaviour

A right convergence behaviour together with a good learning ability is the key factor for any successful stochastic search algorithm. Thus, we conducted a deep analysis on the dynamic behaviour of OPT- IA as reported in this section. We used the networks *American College Football*, *Cattle PPI* and *C. elegans MRN* because, being different in types, sizes, density (Δ in Table 1), and mainly complexity, they allow a more robust analysis. As described above (Sect. 4.1), all these experiments were averaged over 30 independent runs.

Figure 5 shows the convergence curves of OPT- IA, and, in particular, the best fitness, average fitness of the population, and average fitness of the hypermutated population. In particular, one can see how in all three plots OPT- IA shows a very

good convergence towards the optimal solution. Indeed, the three curves grow slowly and improve step by step until they reach the best solution. It is important to note that initially the three curves are very close, and then begin to differentiate as they approach the optimal solution (see inset plots). This is due to the diversification of the solutions whose crucial impact happens mainly when the improvements are limited, and consequently when OPT- IA needs to get out of local optima.

Simply put, all three curves keep a right distance from each other, confirming the existence of a good degree of diversity among the solutions, which is useful for avoiding and/or escaping from local optima. Furthermore, it is also important to highlight that the curve of the best fitness does not increase monotonically: for some generations, the curve decreases slightly, and this corresponds to the discovery of better fitness values, right in the next generations.

Since OPT- IA is driven by random process, together to the analysis of the convergence behaviour becomes important also understand its learning ability, that is the information amount it is able to discover during the search process. The information learned during the evolutionary process clearly affects the overall performance of the algorithm. The *information gain*, also known as *Kullback–Leibler divergence*, is a well-known metric for analysing the amount of information gained by an algorithm whilst searching for the solution, i.e. during the learning phase (Kullback 1959; Cutello et al. 2007). Basically, it measures the reduction of entropy with respect to an initial distribution function (timestep $t = 0$); that is, generalizing, the entropy reduction from a prior state to a given next state. A good randomness measure is also given by the Shannon’s entropy (Shannon 1948), which is among the most used in Information Theory because it is able to measure the uncertainty of a random process. It defines the entropy of a random value in terms of its distribution probability. It follows that the Kullback–Leibler divergence (Kullback 1959) measures the “distance” between two probability distributions and how different they are.

In order to compute the information gain in OPT- IA, we indicate with B_m^t the number of all those solutions (B cells) having fitness value m at the time t , and we define the distribution function of solutions ($f_m^{(t)}$) as the ratio between B_m^t and d , that is the total number of the solutions:

$$f_m^{(t)} = \frac{B_m^t}{\sum_{m=0}^h B_m^t} = \frac{B_m^t}{d}. \tag{2}$$

It follows then that the definitions of the information gain $K(t, t_0)$, and the entropy, $E(t)$ are given, respectively, by:

$$K(t, t_0) = \sum_m f_m^{(t)} \log(f_m^{(t)} / f_m^{(t_0)}), \tag{3}$$

$$E(t) = - \sum_m f_m^{(t)} \log f_m^{(t)}. \tag{4}$$

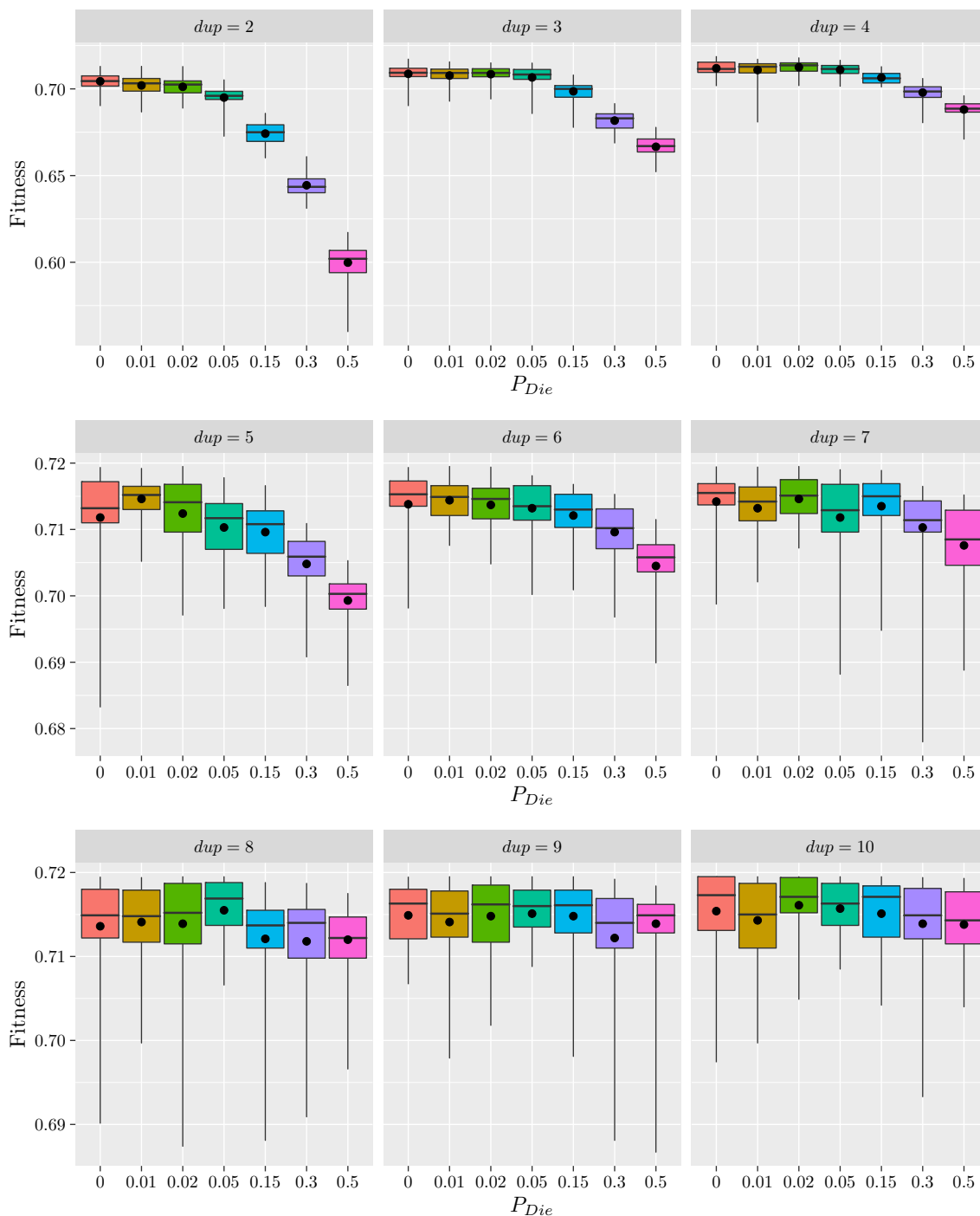


Fig. 3 Experimental tuning of P_{die} parameter with $M = 1$ on the *Cattle PPI* network

Equation 3 indicates therefore the amount of the information discovered by OPT-IA during the convergence process with respect the initial population $P^{(t=0)}$. The *maximum information-gain principle* (Jaynes 2003) tells us that once the search process begins the information gain curve monotonically increases until to reach a peak, which corresponds to

the maximum quantity of information discovered. Just after this peak, the curve transitions to a (roughly) steady-state, or in case it is fluctuating, however it will not reach values close to the maximum peak point:

$$\frac{dK}{dt} \geq 0. \tag{5}$$

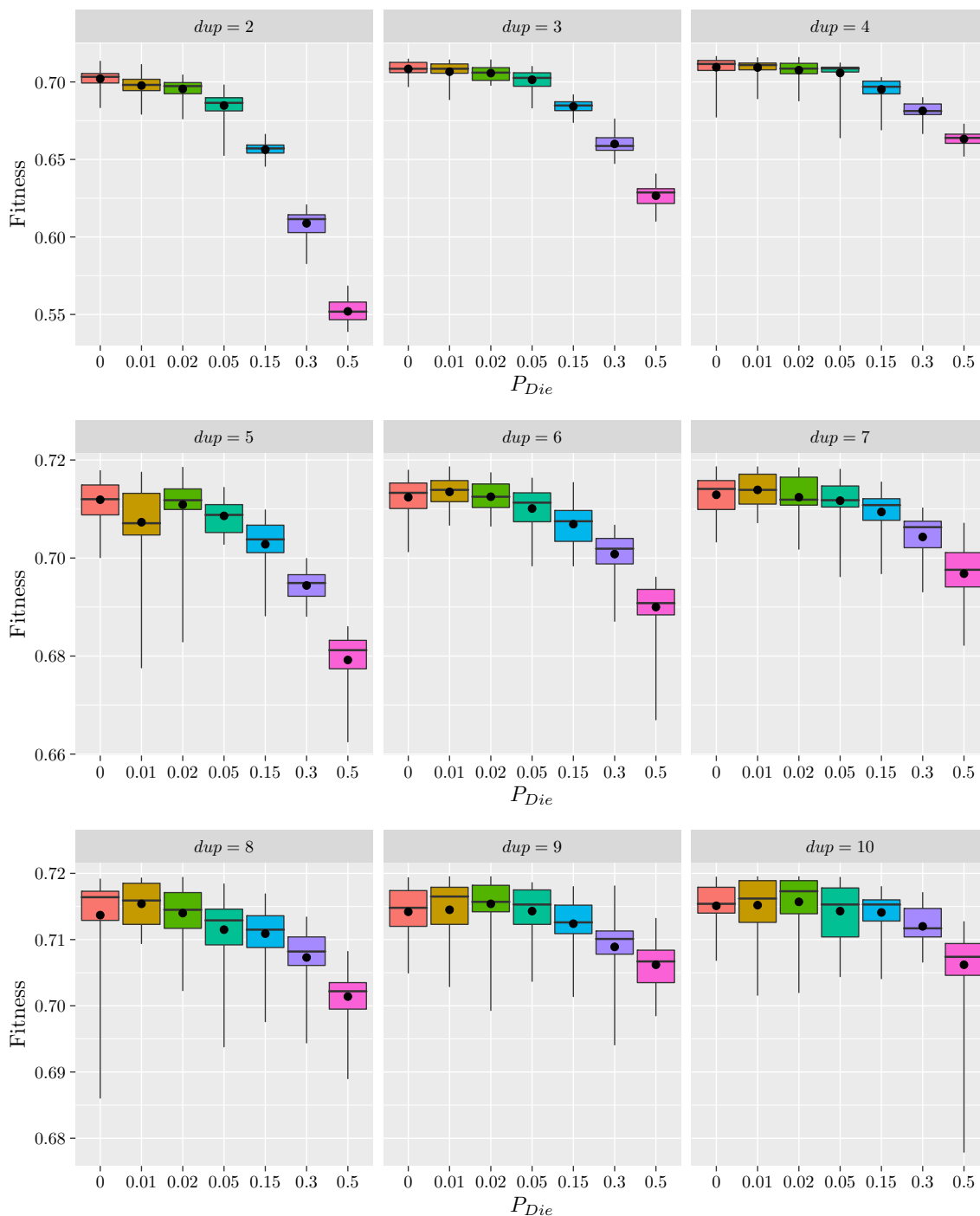


Fig. 4 Experimental tuning of P_{die} parameter with $M = 2$ on the *Cattle PPI* network

Overall, the maximum information-gain principle makes it a suitable tool to perform an appropriate parameter tuning and, at the same time, understand the convergence behaviour of the algorithm both *on-line* and at *run-time*.

As for the convergence analysis, the same three networks were also used to evaluate the learning ability of OPT- IA. Figure 6 displays the information gain curves, respectively, on

the *American College Football* in plot (a), *Cattle PPI* in plot (b) and *C. elegans MRN* in plot (c). Inspecting all three plots, we can clearly see how OPT- IA quickly gains enough information during the first iterations, reaching regions of search space with a good average, which proves the efficiency of the mutation operators designed to explore the search space. After that, due to the fully random search process and with-

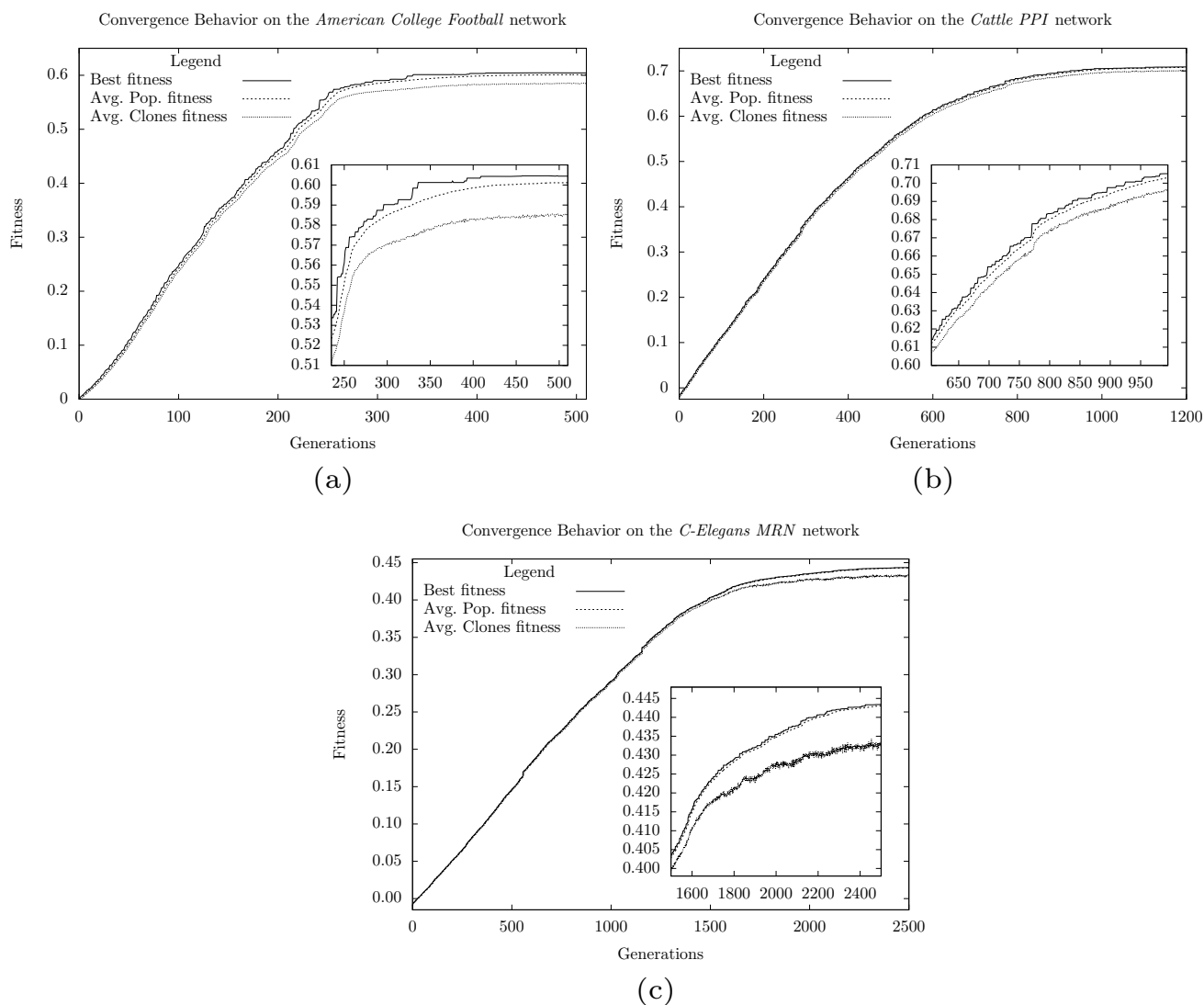


Fig. 5 Convergence behaviour of OPT- IA on **a** *American College Football*, **b** *Cattle PPI* and **c** *C. elegans MRN* networks. The figures show the average fitness function values and the best B cell versus generations

out any guided search specific to the problem, the learning process alternates in gaining or losing information, until it reaches the highest peak that corresponds to having reached the best overall solution. The inset plot shows the relative standard deviations. In plot (a) of Fig. 6 it is possible to see the learning behaviour of OPT- IA on the *American College Football* network. In this plot, the algorithm shows a different learning behaviour compared to the other two considered networks, because this network is a little simpler in size and network density. Indeed, once the highest peak is reached (in the generations range [20, 30]), OPT- IA begins to lose information until around the 200th iteration when the curve begins to increase again, and therefore it starts to gain again information. In the inset plot, we show the relative deviation standard (σ) of OPT- IA, which measures the amount of dispersion (uncertainty) inside the population. It is interesting

to note, indeed, that the iteration point where the algorithm reaches the maximum information gain corresponds exactly to the standard deviation lowest point. Correctly, then, to the maximum information gain corresponds minimum uncertainty. Similarly, at the point of the lowest information gain, reached before 200th generations, there is the highest standard deviation value. The information gain curves displayed in plots (b) and (c) show, instead, a steadier state behaviour once the higher information value is reached. Interestingly, we can see in both plots that after 1000 generations OPT- IA begins to discover new information, and particularly in the plot (b) it reaches even the highest information gain value. This is consistent with the standard deviation curves, which reach lowest values just after the 1000 generations.

In conclusion, both analyses (convergence and learning) prove the efficiency and robustness of OPT- IA in community

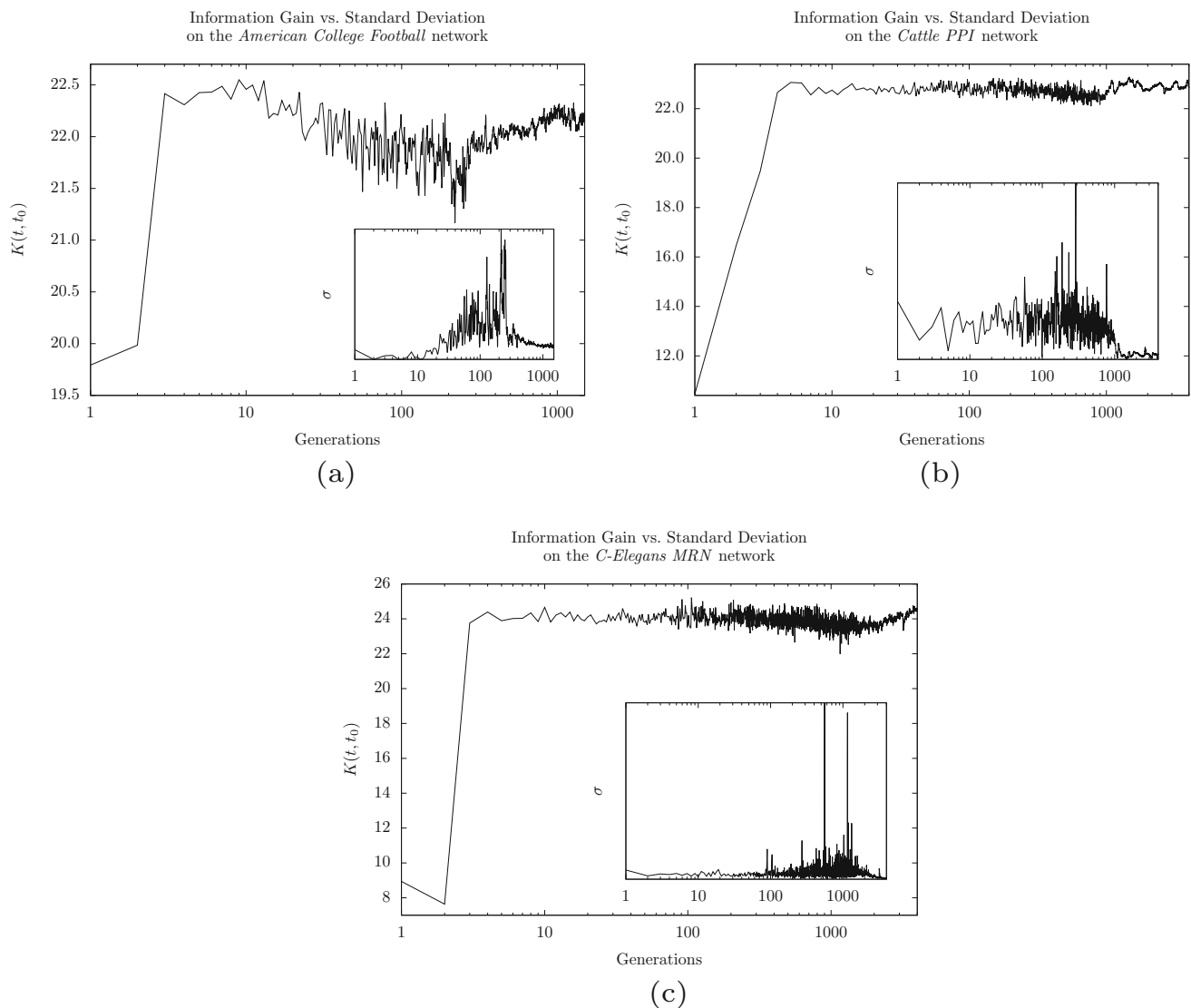


Fig. 6 Information gain curves of OPT-IA on **a** *American College Football*, **b** *Cattle PPI* and **c** *C. elegans MRN* networks. The inset plots show the relative standard deviations

detection task. More importantly, they highlight and prove how OPT-IA needs more iterations to discover high-quality solutions due to the strong randomness present in the developed operators.

4.4 Computational time complexity

The running time of OPT-IA for reaching the best solution is another crucial measure to take into account for proving the efficiency of the proposed immune algorithm. We used the *Time-To-Target plots* (Aiex et al. 2002; Feo et al. 1994) (*TTT-plots*) which are a standard graphical methodology for data analysis and for characterizing the running time of stochastic algorithms in order to solve a specific optimization problem. They measure the CPU times to find the target

of the problem instance tackled. The basic idea behind of *TTT-plots* is to compare the empirical and theoretical distributions, i.e. it displays the probability that an algorithm will find a solution as good as a target within a given running time. A Perl program has been proposed by Resende et al. in Aiex et al. (2007) for automatically generating the *TTT-plots*, which produces two different plots: (i) theoretical Quantile-Quantile plot (*QQ-plot*) with superimposed variability information, and (ii) superimposed empirical and theoretical distributions.²

In order to perform such an analysis, the OPT-IA algorithm is run n times on a given instance using the achieving of a target value (i.e. achieve global optimum) as a stopping

² The perl program can be downloaded at <http://mauricio.resende.info/ttplots/ttplots.zip>.

criterion. Obviously, for each single run, a different seed is considered for the random number generator so to have independent runs. Note that the larger the number n considered, the closer the empirical distribution will be to the theoretical one. Therefore, following the suggestions given in Aiex et al. (2007), we set $n = 200$ because it has been proven that this value gives very good approximations of the theoretical distributions. This analysis has been conducted on five different networks in size and complexity: *Grevy's Zebras*, *Zachary's Karate Club*, *Bottlenose Dolphins*, *Books about US Politics* and *C. elegans MRN*. For a proper analysis, it is important to consider not easy instances, since the exponential distribution would degenerate to a step function, due to the very small CPU times in almost all runs, as asserted in Aiex et al. (2007). Furthermore, these networks have been considered also because the new stopping criterion requires that in the tackled networks/instances the success rate is 100%.

In Figs. 7, 8, and 9, the *TTT* – plots produced on the cited networks are shown. In each figure, the left plot shows the empirical versus theoretical distribution, whilst in right plots show the *QQ* – plots with variability information.

It is important to point out that the *TTT* – plots experiments on the *Books about US Politics* network were performed considering the best solution found ($t = 0.5272$) as target value for the stopping criterion, and OPT- IA was able to find it in all 200 runs, although in Table 6 (Sect. 5) the *best* and *mean* values are not the same. This confirms that with a larger number of iterations the developed search process is able to discover even better solutions until it reaches the optimal ones, of course, with higher computational complexity time. However, from the relative *TTT* – plots (bottom plots in Fig. 8), the empirical curve follows the same behaviour of the theoretical one, proving consequently the efficacy of OPT- IA on this network. For the *C. elegans MRN* network—one of the larger networks in the dataset—two different target values were, instead, considered as stopping criteria, since OPT- IA found better modularity than the compared algorithms (see Table 6), either as *best*, *mean* and *worst* values. The first experiment was then conducted considering 0.4185 as target value (upper plots, Fig. 9), which corresponds to the best modularity found among all compared algorithms, and specifically by *HDSA*. Moreover, because the worst modularity computed by OPT- IA is still better than the one found by *HDSA*, a second *TTT* – plots experiment was performed setting the stopping target to 0.4221 (bottom plots in Fig. 9), i.e. the worst solution of OPT- IA on such a network (see Table 6).

Overall, by inspecting all plots in the three figures (Figs. 7, 8, and 9), it emerges how the empirical curve perfectly fits the theoretical one in all the four social networks, whilst OPT- IA improves the theoretical trend in the biological one. In the *QQ* – plots, instead, the OPT- IA algorithm shows how its results are in most of the cases equal or better

than the theoretical ones, and the empirical curve is much faster than the theoretical one. Focusing the analysis only on the plots of Fig. 9, that is the two different targets considered for the biological network, it appears clear how OPT- IA easily achieves the same maximum modularity of *HDSA*, whilst (obviously) needs more time to reach larger values of modularity. However, the empirical curve fits perfectly the theoretical one. Even on the *QQ* – plot (Fig. 9b) the empirical curve almost always fits the estimated one, also showing a better behaviour than in that shown in Fig. 9d.

4.5 The asymptotic computational analysis

To compute the upper bound of the computational cost of OPT- IA, it is important to recall that being OPT- IA a population-based algorithm, any computational analysis must be done with respect to the size of the input problem and implementation features, but also with respect to the choice of the key parameters, such as population size (d), maximum iteration numbers T_{max} , hypermutation operators, etc. Below, all these issues are properly discussed.

Inspecting the pseudocode of OPT- IA (Algorithm 1) it is possible to assert that:

- any solution is represented by an array of length n , where n is the number of vertices of the input graph;
- the procedure *InitializePopulation*(d) has the aim to randomly create a population of d solutions. It follows then that the total cost is $\mathcal{O}(d \times n)$. Note that the parameter d is a constant experimentally set (it is user defined), and therefore it is independent from the size of the input. Besides, in all presented experiments it was set to the value 100, therefore, is possible to assert that the cost of the procedure is actually $\mathcal{O}(n)$;
- the procedure *ComputeFitness*() simply evaluates the quality, i.e. the fitness, of all 100 solutions of the population using the Eq. 1. It follows then that the cost of the procedure is $\mathcal{O}(n^2)$;
- the aims of the procedure *Cloning*() is to create *dup* copies of each element of the population. Looking to the experimental setting, also *dup* is a fixed parameter and then independent from the input size as well. This allows us to say that the cost of *Cloning*() is $\mathcal{O}(n)$;
- the *Hypermutation*() procedure mutates each element of the population with constant probability M , and it is based on three different implementations. As each of these is based on the random selection of either a vertex or a cluster, it can happen having to reallocate several vertices in the two most consuming implementations, that is *destroy* and *fuse*. It follows then that the procedure has an upper bound $\mathcal{O}(n)$;
- in the *Precompetition*() procedure, two vertices are randomly selected, and their community number is

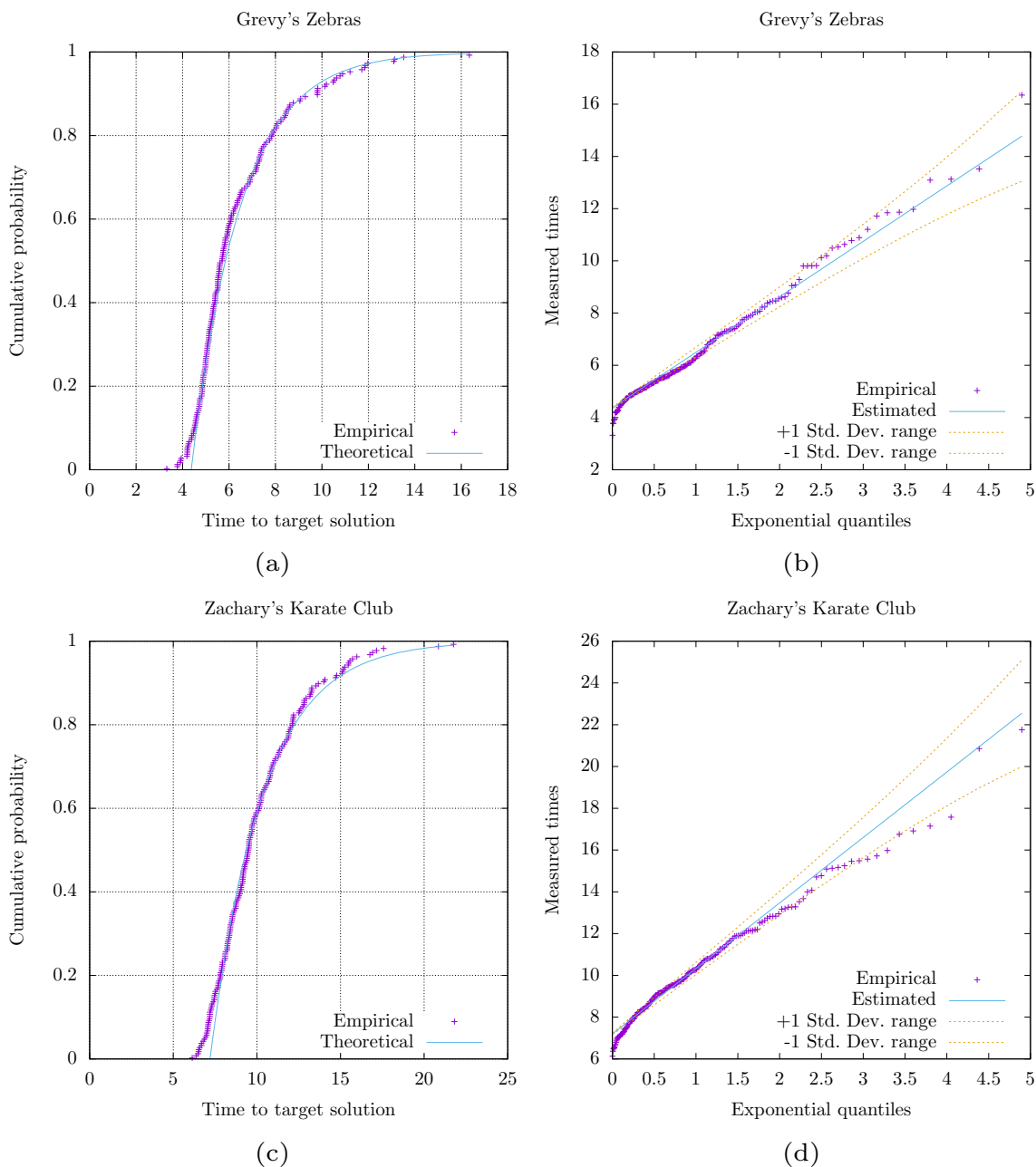


Fig. 7 Empirical versus theoretical distributions **a, b** and *QQ* – plots with variability information **c, d**, obtained on the *Grevy's Zebras* and *Zachary's Karate Club* networks

checked. If the number is the same (i.e. they belong to the same community), then the vertex with lower fitness is deleted with probability 1/2. Considering that the fitness was already computed, the overall cost is then $\mathcal{O}(1)$;

- the procedure *StochasticAging()* inspects all elements of the population and each one is removed with a P_{die} probability. Since P_{die} is a user-defined parameter and then constant, we can assert that the overall cost of the procedure is $\mathcal{O}(n)$;

- finally, *Selection()* is the last procedure that is performed, which chooses the best d solutions among $P_a^{(pre)}$ and $P^{(hyp)}$. Recalling that the number of elements (d) in both populations is constant and independent of the size of the input, it follows that the cost of the procedure is therefore $\mathcal{O}(1)$.

Then, summing up, from the computational analysis done for each procedure, it is possible to assert that the cost of

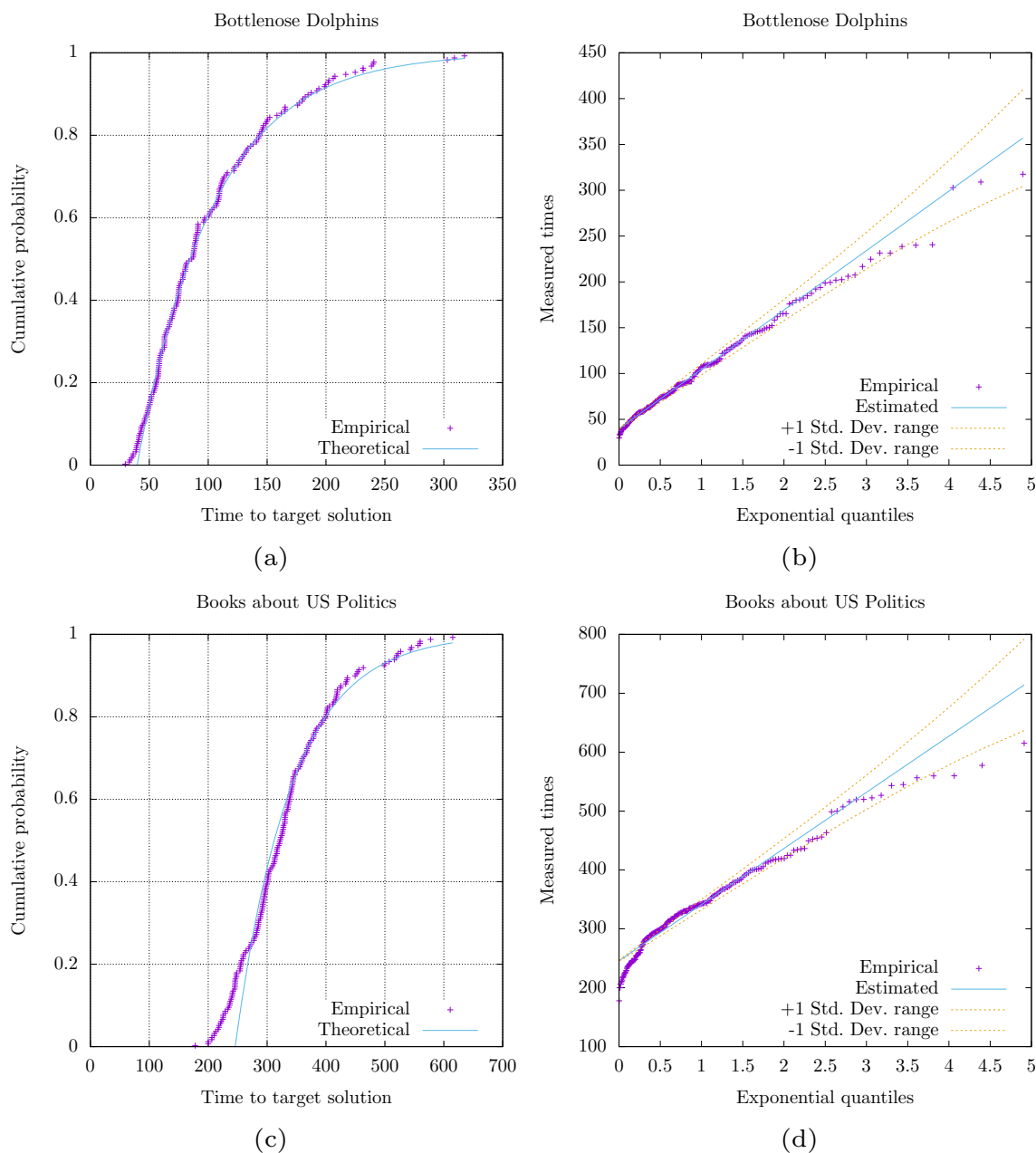


Fig. 8 Empirical versus theoretical distributions **a–c** and *QQ* – plots with variability information **b–d**, obtained on the *Bottlenose Dolphins* and *Books about US Politics* networks

one iteration of OPT- IA has a computational upper bound $\mathcal{O}(n^2 + n) = \mathcal{O}(n^2)$.

Let now us consider the number of iterations. It is worth to highlighting that, unlike to the other parameters, the number of generations is instead related with the size of the input: the bigger the input, the bigger is the number of generations that we expect to need. But how does it grow with respect to n ? Taking into account that the number of iterations is, however, an input parameter, then a constant number also on large networks, and looking to the results of the presented experiments, it is possible to assert that in the worst case its

growth is linear with respect to n . It follows that we can assert that OPT- IA needs at most $c \times n$ generations to obtain good results, wanting to be very cautious. Indeed, inspecting the convergence behaviours in Fig. 5, for instance plot (b), it is possible to see that after around 1200 iterations ($\sim 4.48 \times |V|$) the algorithm reaches the best solution. The same can be said for the other plots in Figure. In the light of this, adding such bound to the overall computational analysis, it is possible to claim that the upper bound of the OPT- IA running time is $\mathcal{O}(n^3)$.

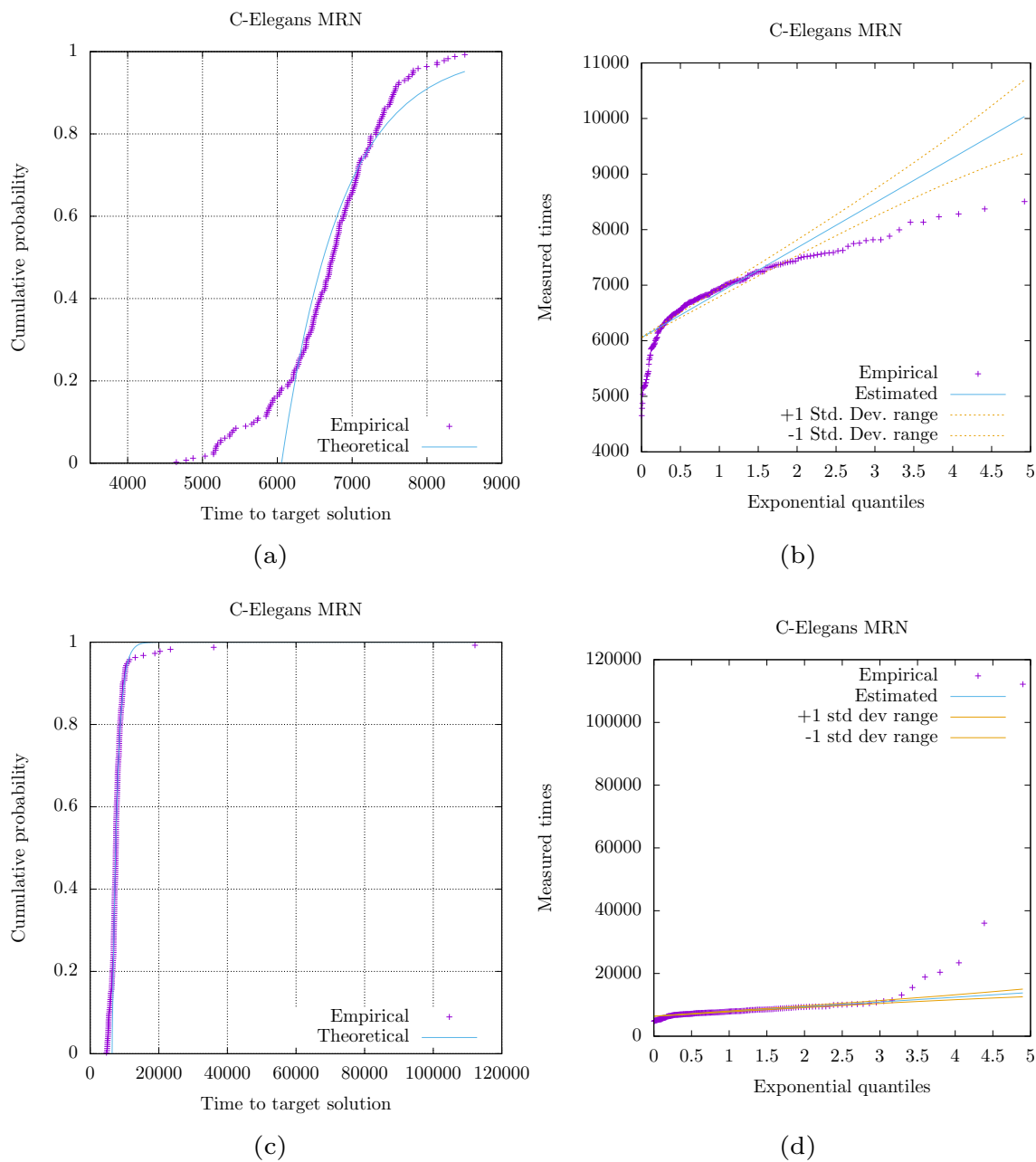


Fig. 9 Empirical versus theoretical distributions **a, c** and *QQ* – plots with variability information **b–d**, obtained on the *Books about US Politics* and *C. elegans MRN* networks

4.6 Precompetition operator effectiveness

The precompetition operator, in addition to the aging operator, plays a key role on the performances of OPT- IA since it allows the algorithm to jump away from local optima by introducing heterogeneity in the population. This, of course, is a crucial characteristic especially when addressing hard and complex problem. Although the usefulness and efficiency of the aging operator is well known Jansen and Zarges (2011a, b), little instead is possible to assert on the efficacy

of the precompetition operator, and how it affects the performance of OPT- IA.

In the light of this, in this section, an analysis on the overall effectiveness of the precompetition operator is presented, and it is shown in Table 2. For this analysis, four biological networks were considered (*Cattle PPI*, *E. coli TRN*, *C. elegans MRN*, and *Helicobacter pylori PPI*), and used for inspecting the convergence behaviour of OPT- IA, by enabling or disabling such an operator.

Table 2 The experimental results of OPT- IA with and without the *Pre-competition* operator on biological networks

Instance		Precompetition operator	
		Enabled	Disabled
Cattle PPI	<i>Best</i>	0.7195	0.7195
	<i>Mean</i>	0.7156	0.7151
	<i>StD</i>	0.0036	0.0044
<i>E. coli</i> TRN	<i>Best</i>	0.7796	0.7776
	<i>Mean</i>	0.7711	0.7701
	<i>StD</i>	0.0051	0.0055
<i>C. elegans</i> MRN	<i>Best</i>	0.4490	0.4470
	<i>Mean</i>	0.4369	0.4340
	<i>StD</i>	0.0053	0.0081
Helicobacter pylori PPI	<i>Best</i>	0.5416	0.5329
	<i>Mean</i>	0.5249	0.5235
	<i>StD</i>	0.0063	0.0055

Looking at the outcomes reported in table, the usefulness and efficacy of the precompetition operator is clearly evident: it allows OPT- IA to reach better modularity values not only with respect the maximum value found, but also with respect to the mean of the best found values in all independent runs, with the consequence of allowing the algorithm to obtain lower standard deviation values. It is important to highlight that, except for the *Cattle PPI* network where the best modularity is the same for both versions, the precompetition operator allows OPT- IA to produce considerably higher modularity values, proving the successful effect of this operator.

The precompetition operator, in combination with the stochastic aging compensates the indirect elitism provided by the selection operator; therefore, it helps to maintain a right balance of diversity in the population.

5 Results

We will discuss now the overall experimental results and compared them with the results obtained by state-of-the-art algorithms. It is important to stress first that, in a preliminary work (Spampinato et al. 2019), OPT- IA was compared to LOUVAIN algorithm on a set of different, simple, and small networks, which for simplicity are reported in Table 3. By inspecting this table, it becomes clear how OPT- IA, based on a pure random search, outperforms one of the best deterministic approach on community detection, which is LOUVAIN algorithm. However, given the low of complexity of these tested networks, a deeply and detailed analysis must be conducted in order to evaluate the real performance of OPT- IA. Therefore, for these new experiments, all networks in the data

Table 3 OPT- IA versus LOUVAIN

Instance	V	LOUVAIN		OPT- IA	
		Q	N_C	Q	N_C
<i>Dolphins</i>	62	0.5188	5	0.5285	5
<i>Karate</i>	34	0.4156	4	0.4198	4
<i>Ukfaculty</i>	81	0.4488	4	0.4488	4
<i>Miserables</i>	77	0.5583	6	0.5600	6
<i>Huckleberry</i>	69	0.5346	4	0.5346	4
<i>GN_benchmark2</i>	128	0.4336	2	0.4336	2
<i>GN_benchmark4</i>	128	0.5393	4	0.5393	4
<i>LFR_benchmark</i>	128	0.1560	6	0.1980	5
<i>Almost_lattice</i>	64	0.5279	8	0.5576	8
<i>3mixed</i>	128	0.3682	5	0.4297	3

set in Table 1 were considered, and the experimental protocol described in Sect. 4.1 was used. The main goal of these experiments, as well as all comparisons made, is to prove the competitiveness and reliability of OPT- IA in terms of solution quality found, i.e. maximizing the modularity function (Eq. 1).

To this end, the proposed OPT- IA algorithm was compared to several different heuristics and metaheuristics (20 in the overall), each of them designed and developed as a modularity optimization approach (Atay et al. 2017; Li et al. 2020; Doush et al. 2020). Specifically, the algorithms considered for the comparisons, in addition to the LOUVAIN's one, are:

BA—*Bat Algorithm*, a metaheuristic method based on the echolocation behaviour of bats (Yang 2010);

GSA—*Gravitational Search Algorithm*, a metaheuristic optimization algorithm based on the law of gravity and mass interactions (Rashedi et al. 2009);

BB—*BC*—*Big Bang-Big Crunch* algorithm, an algorithm inspired by the theories of the universe evolution in which, during the main phase, energy dissipation produces disorder and randomness, whilst in a second stage the randomly distributed particles are drawn into an order, i.e. the values in the vectors of the function to be optimized are determined (Erol and Eksin 2006);

BADE—*improved Bat algorithm based on Differential Evolutionary*, an improved version based on the combination (hybridization) of Bat Algorithm, and Differential Evolution (*DE*) algorithm (Storn 1995; Storn and Price 1997), where this latter is used in population regeneration process. Both algorithms are used together for the selection of adjacent nodes;

SSGA—*Scatter Search algorithm based on Genetic Algorithm*, a Scatter Search (SS) approach (Glover 1977; Martí et al. 2006) of the best chromosomes provided by Genetic Algorithm (GA) (Holland 1975; Goldberg 1989) and subject-

ing the population to the crossover and the mutation processes around the best solutions;

HDSA—Hyper-heuristic Differential Search Algorithm, a hyper-heuristics based on the migration of artificial super-organisms (Civicioglu 2012), where each of them in the population migrates between the maximum or minimum solution of the problem using the Differential Search Algorithm (DSA) in the process of regeneration of individuals;

MA—Net (Naeni et al. 2015), a memetic algorithm based on the combination of a genetic algorithm with a local search;

GACD (Shi et al. 2009), a genetic algorithm that takes advantage of the efficiency of the locus-based adjacency encoding scheme to represent a community partition;

CC—GA—Clustering Coefficient-based Genetic Algorithm (Said et al. 2018), a genetic algorithm that uses the clustering coefficient (CC), which is a social networks analysis measure, to generate a better initial population;

MSIG—Multi-Start Iterated Greedy algorithm (Sánchez-Oro and Duarte 2018), which uses a new greedy procedure for generating the initial solutions and reconstruct the solutions, but has the disadvantage of being computationally expensive;

IDPSO—RO—Improved Discrete Particle Swarm Optimization with Redefined Operator (Cao et al. 2015), based on particle swarm optimization, in which the update formulas of velocity and position are redefined according to the locus-based adjacency representation;

IG—Iterated greedy algorithm (Li et al. 2020) based on an iterative process that combines a destruction phase and a reconstruction phase: a complete candidate solution is partially destructed, and afterwards a new complete candidate solution is reconstructed via a greedy constructive heuristic;

MOBA—Multi-Objective Bat Algorithm (Doush et al. 2020), which is a multi-objective bat algorithm adapted to model and solve the community detection problem;

and, finally, the following heuristics and metaheuristics *EFF* (*Enhancement FireFly Algorithm*), *RB* (*Rosvall and Bergstrom Algorithm*), *Blondel Algorithm*, *RN* (*Ronhovde and Nussinov Algorithm*) (Ronhovde and Nussinov 2009), *CNM* (Newman and Girvan 2004) and *MOGA—Net* algorithm (Pizzuti 2008), which each was taken from (Doush et al. 2020).

In accordance with what previously described, the parameters setting of OPT-IA, in all the performed experiments, are: $d = 100$ as population size; $P_{die} = 0.02$ the probability of random aging operator; mutation rate M set to 1; and $T_{max} = 4000$ as the maximum number of generations used as stopping criterion. The duplication parameter dup in according with the parameters tuning reported in Sect. 4.2 has been set to 4 ($dup = 4$) for all those instances with $|V| < 100$ (small social networks), whereas for the larger ones ($|V| \geq 100$) the experiments were performed with $dup = 9$ and $dup = 10$. Every experiment has been per-

Table 4 The experimental results of OPT-IA on social networks with $dup = 4$

Network	Best	Mean	Worst	StD	N_C
Grevy's Zebras	0.2768	0.2768	0.2768	0.0000	4
Zachary's Karate Club	0.4198	0.4198	0.4198	0.0000	4
Bottlenose Dolphins	0.5285	0.5285	0.5285	0.0000	5

formed on 30 independent runs. The obtained outcomes are summarized in Tables 4 and 5.

Table 4 reports the results of the proposed algorithm on small social networks, whilst in Table 5 we show the results on larger social networks, and on biological networks. In this last table, the best results obtained for $dup = 9$ or $dup = 10$ are also highlighted in boldface.

This different setting of the duplication parameter is obviously due to the simplicity of the first networks ($dup = 4$) compared to the last ones ($dup = 9$ or $dup = 10$), which consequently require a more targeted search, and a less wide exploration of the solution space. Larger networks with a density $\Delta \geq 1\%$ (see Table 1) do not require a great variability in the population, and for this reason $dup = 9$ seems to be the most appropriate value. Indeed, although in the social networks there is little difference in the results between the two dup values ($dup = 9$ vs $dup = 10$), in *C. elegans MRN*, where $\Delta = 1.98\%$, a significant improvement is instead obtained in terms of best modularity found (*Best*), and average of values (*Mean*). On the other hand, for all networks with a low density ($\Delta < 1\%$) the parameter $dup = 10$ ensures good average values (*Mean*) in *Cattle PPI* instance and best modularity (*Best*) for *E. coli TRN* network. In these cases, a small increase in the dup parameter, i.e. having a larger number of duplicates, allows to produce higher variability, and consequently enables to work well on very sparse networks.

Tables 6, 7, 8, 9 report the comparisons of OPT-IA with other heuristics and metaheuristics. The results shown are averaged on 30 independent runs for all algorithms. Note that, unlike other algorithms that use a maximum number of generations fixed, *MA—Net* stops running just only when 30 generations are performed without any improvement. For each table, the best modularity values (*Best*), average values (*Mean*), worst modularity (*Worst*), standard deviation (*StD*), and number of communities discovered (N_C) are showed, respectively.

By analysing Table 6, it is clear how the proposed OPT-IA considerably outperforms all compared algorithms, excepts for *HDSA* hyper-heuristics. Regarding this latter, however, it is possible to note how both algorithms (OPT-IA and *HDSA*) show identical performances on the first two networks in the table reaching the same values of *Best* and *Mean*, whilst on the last two, *HDSA* outperforms OPT-IA only with respect

Table 5 The experimental results of OPT- IA on social and biological networks with $dup = \{9, 10\}$

Network	dup	Best	Mean	Worst	StD	N_C
Books about US Politics	9	0.5272	0.5270	0.5208	0.0012	5
	10	0.5272	0.5268	0.5208	0.0016	5
American College Football	9	0.6046	0.6011	0.5848	0.0052	10
	10	0.6046	0.5999	0.5891	0.0050	10
Cattle PPI	9	0.7195	0.7148	0.7018	0.0044	40
	10	0.7195	0.7161	0.7049	0.0039	40
E-Coli TRN	9	0.7734	0.7660	0.7486	0.0050	27
	10	0.7795	0.7670	0.7589	0.0049	32
C-Elegans MRN	9	0.4487	0.4366	0.4221	0.0070	8
	10	0.4464	0.4377	0.4231	0.0061	10

Table 6 Comparison of OPT- IA on social networks

Networks		OPT- IA	LOUVAIN	HDSA	BADE	SSGA	BB- BC	BA	GSA	MA- NET
Grevy’s Zebras	<i>Best</i>	0.2768	0.2768	0.2768	0.2768	0.2768	0.2768	0.2768	0.2768	–
	<i>Mean</i>	0.2768	–	0.2768	0.2768	0.2768	0.2766	0.2768	0.2768	–
	<i>Worst</i>	0.2768	–	0.2768	0.2768	0.2768	0.2761	0.2768	0.2768	–
	<i>StD</i>	0.0000	–	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	–
	N_C	4	4	4	4	4	4	4	4	–
Zachary’s Karate Club	<i>Best</i>	0.4198	0.4189	0.4198	0.4198	0.4198	0.4198	0.4198	0.4198	0.420
	<i>Mean</i>	0.4198	–	0.4198	0.4188	0.4198	0.4196	0.4133	0.4170	0.419
	<i>Worst</i>	0.4198	–	0.4198	0.4156	0.4198	0.4188	0.3946	0.4107	–
	<i>StD</i>	0.0000	–	0.0000	0.0018	0.0000	0.0004	0.0105	0.0037	0.002
	N_C	4	4	4	4	4	4	4	4	4
Bottlenose Dolphins	<i>Best</i>	0.5285	0.5285	0.5285	0.5268	0.5257	0.5220	0.5157	0.4891	0.529
	<i>Mean</i>	0.5285	–	0.5282	0.5129	0.5200	0.5141	0.4919	0.4677	0.523
	<i>Worst</i>	0.5285	–	0.5276	0.4940	0.5156	0.5049	0.4427	0.4517	–
	<i>StD</i>	0.0000	–	0.0005	0.0120	0.0040	0.0068	0.0289	0.0155	0.004
	N_C	5	5	5	4	5	5	4	6	5
Books about US Politics	<i>Best</i>	0.5272	0.5205	0.5272	0.5239	0.5221	0.4992	0.5211	0.4775	0.527
	<i>Mean</i>	0.5270	–	0.5272	0.5178	0.5203	0.4914	0.5020	0.4661	0.526
	<i>Worst</i>	0.5208	–	0.5272	0.5137	0.5167	0.4799	0.4815	0.4558	–
	<i>StD</i>	0.0012	–	0.0000	0.0042	0.0024	0.0084	0.0149	0.0079	0.002
	N_C	5	4	5	4	5	9	3	5	5
American College Football	<i>Best</i>	0.6046	0.6046	0.6046	0.5646	0.5330	0.5171	0.5523	0.4175	0.605
	<i>Mean</i>	0.6011	–	0.6033	0.5513	0.5277	0.5061	0.5272	0.4032	0.601
	<i>Worst</i>	0.5848	–	0.6019	0.5430	0.5189	0.4986	0.4742	0.3905	–
	<i>StD</i>	0.0052	–	0.0009	0.0085	0.0057	0.0069	0.0325	0.0109	0.003
	N_C	10	10	10	11	6	10	7	5	10
Jazz Musicians	<i>Best</i>	0.4451	0.4451	–	–	–	–	–	–	0.445
	<i>Mean</i>	0.4449	–	–	–	–	–	–	–	0.445
	<i>Worst</i>	0.4449	–	–	–	–	–	–	–	–
	<i>StD</i>	0.0001	–	–	–	–	–	–	–	0.000
	N_C	4	4	–	–	–	–	–	–	4

the average values (both reach the same *Best* values). On the network *Bottlenose Dolphins*, instead OPT- IA strictly outperforms *HDSA* reaching a better mean value, and a standard deviation value equal to zero. It is important to emphasize that *HDSA* is a hyper-heuristic and then by conception exploits different type of appropriate heuristics: in each generation it considers the one that returns the best result. It follows obviously that this method is potentially more robust from a *Mean* value perspective. However, the difference between the values obtained by both heuristics, as *Best* and *Mean*, is almost irrelevant, demonstrating that the two algorithms OPT- IA and *HDSA* can be considered comparable in the overall.

In Table 7, OPT- IA is compared with a second group of more recent metaheuristics methods. Also on this comparison, the proposed algorithm outperforms the compared algorithms in all networks. Indeed, if the comparison is inspected from a ranking perspective with respect to the *Best* values, OPT- IA is always at the top, whilst if it is analysed with respect to the *Mean* values, it is easy, instead, to assert that it is always among the first two positions and very often in the first one. It is worth emphasizing once again that, whilst these compared algorithms include deterministic and sophisticated strategies, OPT- IA is fully random both in the generation of the initial population and in the solutions search process into the search space. Therefore, having shown better performances, it confirms the robustness and efficiency of all designed random operators.

In Table 8, OPT- IA is compared with the last heuristics and metaheuristics group, which also includes a multi-objective approach. Here as well, OPT- IA outperforms all compared algorithms excepts for EFF, with which instead it alternates in the first position. Indeed, OPT- IA outperforms EFF and it is in the first rank on the *Bottlenose Dolphins* and *American College Football* networks, whilst it is outperformed by EFF on the *Zachary's Karate Club* and *Books about US Politics networks*, resulting in second position. Also on this comparison, therefore, it is possible to assert that the proposed algorithm is ranked always among the first two positions confirming its reliability and efficiency.

In Table 9, OPT- IA is compared with the first group of algorithms on biological networks. Unfortunately, no results were found by the other considered algorithms on these networks. Thus, inspecting this table, it is possible to see how OPT- IA strictly outperforms all algorithms, included *HDSA*, on the *C. elegans MRN* and *Helicobacter pylori PPI* networks compared to all evaluation metrics (*Best*, *Mean*, *worst* and *StD*), and detecting a smaller community value. However, on the other two networks, OPT- IA and *HDSA* are comparable in *Cattle PPI* with respect to the best value reached, but OPT- IA is outperformed by *HDSA* with respect to the mean values. Also, on the *E. coli TRN* instance *HDSA* outperforms OPT- IA in all assessment values. It is important

to highlight that OPT- IA performs better than *HDSA* on the larger networks. Focusing, finally, the inspection only on the comparison between OPT- IA and LOUVAIN it is easy to assert that the first considerably outperforms the latter, excepts for the *Helicobacter pylori PPI* network. Overall, then, analysing all outcomes and comparisons performed, it is possible to assert that the proposed algorithm OPT- IA outperforms all the compared metaheuristics, and shows comparable performances with respect to hyper-heuristic *HDSA*. It is important to highlight once again that whilst *HDSA*, being a hyper-heuristic, takes advantage of several efficient heuristics, and each time chooses the best solution among all the ones found by them, OPT- IA, instead, is entirely blind to the features of the problem, and it is based only on random search without any deterministic guide. Therefore, taking into account these main differences and features, and, primarily, having found results comparable with those of *HDSA*, it is possible to confirm the efficiency and reliability of the proposed random search algorithm OPT- IA.

In order to study OPT- IA on large networks, i.e. with more than 1000 nodes, a further set of networks was considered and tested, and the results are reported in Table 10. Of course, being OPT- IA fully based on random search, for these experiments a larger number of iterations was needed ($T_{max} = 10^3$). During these experiments, we saw that, increasing the network size, the combination of the developed operators guided the algorithm towards useless search, disregarding to properly and deeply exploring specific neighbourhoods. However, such behaviour did not happen on all previous tested networks. In the light of this, to indirectly guide the search to explore promising regions in the search space more intensively, a simple modification in OPT- IA was made: allow the selection operator to also choose elements having the same fitness. This modified version, reported in Table 10, is labelled as OPT- IA₂, whilst the previous one is called OPT- IA₁. Both versions are compared with the well-known LOUVAIN's algorithm. By comparing the two versions, it appears clear how such a simple change allows OPT- IA to improve the modularity values in the overall. At any rate, the results obtained by the best version of OPT- IA still remain a bit far than the results obtained by LOUVAIN. This is explainable with the features of OPT- IA to be fully based on random search and without any simple deterministic approach. It is very likely that by further increasing the number of generations the gap with LOUVAIN's results will be substantially narrowed. As expected, this is the main limitation of our proposed random search algorithm.

To produce an as large as possible comparison, three other different real networks and 13 other optimization methodologies were considered, in order to evaluate the efficiency of the two variants of OPT- IA from the perspective of the quality of the modularity found and are reported in Table 11.

Table 7 Comparison of OPT- IA on social networks

<i>Networks</i>		OPT- IA	GACD	CC- GA	MSIG	IDPSO- RO	IG
Zachary's Karate Club	<i>Best</i>	0.4198	0.4198	0.4198	0.4198	0.4198	0.4198
	<i>Mean</i>	0.4198	0.4198	0.4198	0.4198	0.4198	0.4198
	<i>Worst</i>	0.4198	0.4198	0.4198	0.4198	0.4198	0.4198
	<i>StD</i>	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	<i>N_C</i>	4	–	–	–	–	–
Bottlenose Dolphins	<i>Best</i>	0.5285	0.5285	0.5285	0.5201	0.5285	0.5285
	<i>Mean</i>	0.5285	0.5272	0.5275	0.5189	0.5271	0.5268
	<i>Worst</i>	0.5285	–	–	–	–	–
	<i>StD</i>	0.0000	0.0020	0.0019	0.0017	0.0010	0.0014
	<i>N_C</i>	5	–	–	–	–	–
Books about US Politics	<i>Best</i>	0.5272	0.5272	0.52729	0.5232	0.5272	0.5269
	<i>Mean</i>	0.5270	0.5257	0.5271	0.5149	0.5261	0.5269
	<i>Worst</i>	0.5208	–	–	–	–	–
	<i>StD</i>	0.0012	0.0002	0.0002	0.0070	0.0021	0.0000
	<i>N_C</i>	5	–	–	–	–	–
American College Football	<i>Best</i>	0.6046	0.5879	0.5787	0.6033	0.6044	0.6046
	<i>Mean</i>	0.6011	0.5777	0.5640	0.5954	0.5900	0.6017
	<i>Worst</i>	0.5848	–	–	–	–	–
	<i>StD</i>	0.0052	0.0069	0.0093	0.0084	0.0129	0.0033
	<i>N_C</i>	10	–	–	–	–	–

Table 8 Comparison of OPT- IA on social networks

<i>Networks</i>		OPT- IA	MOBA	EFF	RB	BLONDEL	RN	CNM	MOGA- NET
Zachary's Karate Club	<i>Best</i>	0.4198	0.4186	0.4201	0.4187	0.4186	0.4177	0.4176	0.4151
	<i>Mean</i>	0.4198	0.4176	0.4201	0.4187	0.4175	0.4157	0.4067	0.4149
	<i>Worst</i>	0.4198	0.4167	0.4201	0.4187	0.4164	0.4137	0.3948	0.4148
	<i>StD</i>	0.0000	0.0004	0.0000	0.0000	0.0011	0.002	0.011	0.001
	<i>N_C</i>	4	–	–	–	–	–	–	–
Bottlenose Dolphins	<i>Best</i>	0.5285	0.5238	0.5242	0.5149	0.5162	0.5186	0.4961	0.5048
	<i>Mean</i>	0.5285	0.5234	0.5233	0.5146	0.5152	0.5166	0.4853	0.5038
	<i>Worst</i>	0.5285	0.5230	0.5231	0.5143	0.5142	0.5146	0.4774	0.5029
	<i>StD</i>	0.0000	0.0001	0.0001	0.0003	0.0010	0.0020	0.0120	0.0090
	<i>N_C</i>	5	–	–	–	–	–	–	–
Books about US Politics	<i>Best</i>	0.5272	0.5198	0.5285	0.5199	0.5207	0.5221	0.5024	0.5176
	<i>Mean</i>	0.5270	0.5183	0.5285	0.5197	0.5206	0.5215	0.4818	0.5136
	<i>Worst</i>	0.5208	0.5168	0.5285	0.5195	0.5205	0.5209	0.5011	0.5075
	<i>StD</i>	0.0012	0.0015	0.0000	0.0002	0.0001	0.0006	0.0177	0.0039
	<i>N_C</i>	5	–	–	–	–	–	–	–
American College Football	<i>Best</i>	0.6046	0.6102	0.6045	0.5999	0.6017	0.6028	0.5766	0.5148
	<i>Mean</i>	0.6011	0.6083	0.6023	0.5938	0.5991	0.5988	0.5474	0.4978
	<i>Worst</i>	0.5848	0.6064	0.6011	0.5877	0.5965	0.5948	0.5283	0.4784
	<i>StD</i>	0.0052	0.0041	0.0018	0.0061	0.0026	0.0040	0.0294	0.0158
	<i>N_C</i>	10	–	–	–	–	–	–	–

Table 9 Comparison of OPT- IA on biological networks

Networks		OPT- IA	LOUVAIN	HDSA	BADE	SSGA	BB- BC	BA	GSA
Cattle PPI	<i>Best</i>	0.7195	0.7195	0.7195	0.7183	0.7118	0.7095	0.7143	0.7053
	<i>Mean</i>	0.7161	–	0.7195	0.7138	0.7079	0.7084	0.7100	0.6983
	<i>Worst</i>	0.7049	–	0.7194	0.7059	0.7052	0.7079	0.7063	0.6949
	<i>StD</i>	0.0039	–	0.0001	0.0051	0.0025	0.0007	0.0035	0.0041
	<i>N_C</i>	40	40	40	41	40	48	42	43
E-Coli TRN	<i>Best</i>	0.7795	0.7786	0.7822	0.7680	0.7507	0.7520	0.7629	0.7416
	<i>Mean</i>	0.7670	–	0.7815	0.7621	0.7457	0.7485	0.7599	0.7375
	<i>Worst</i>	0.7589	–	0.7808	0.7560	0.7412	0.7452	0.7542	0.7328
	<i>StD</i>	0.0049	–	0.0006	0.0043	0.0035	0.0026	0.0034	0.0034
	<i>N_C</i>	32	40	47	58	61	71	56	61
C-Elegans MRN	<i>Best</i>	0.4487	0.4263	0.4185	0.3473	0.3336	0.3374	0.3514	0.3063
	<i>Mean</i>	0.4366	–	0.4074	0.3385	0.3220	0.3266	0.3438	0.3039
	<i>Worst</i>	0.4221	–	0.3962	0.3335	0.3124	0.3194	0.3356	0.2974
	<i>StD</i>	0.0070	–	0.0010	0.0054	0.0077	0.0074	0.0073	0.0037
	<i>N_C</i>	8	9	13	25	22	21	22	24
Helicobacter pylori PPI	<i>Best</i>	0.5416	0.5450	0.5086	0.4926	0.4726	0.4681	0.4900	0.4600
	<i>Mean</i>	0.5249	–	0.5078	0.4854	0.4695	0.4660	0.4814	0.4567
	<i>Worst</i>	0.5116	–	0.5048	0.4809	0.4659	0.4642	0.4738	0.4549
	<i>StD</i>	0.0063	–	0.0017	0.0047	0.0021	0.0018	0.0073	0.0020
	<i>N_C</i>	19	23	52	69	70	75	62	77

Table 10 Experimental results of two variant of OPT- IA compared to the LOUVAIN algorithm on larger biological networks

Instance	Alg.	<i>Best</i>	<i>Mean</i>	<i>Worst</i>	<i>StD</i>	<i>N_C</i>
E. coli MRN	OPT- IA ₁	0.3287	0.3141	0.3026	0.0074	31
	LOUVAIN	0.3569	–	–	–	9
S. cerevisiae PPI (1)	OPT- IA ₂	0.3629	0.3437	0.3282	0.0064	9
	OPT- IA ₁	0.6387	0.6145	0.5955	0.0100	257
	LOUVAIN	0.7638	–	–	–	216
S. cerevisiae PPI (2)	OPT- IA ₂	0.6516	0.6344	0.6178	0.0089	386
	OPT- IA ₁	0.4753	0.4606	0.4405	0.0082	288
	LOUVAIN	0.5905	–	–	–	46
	OPT- IA ₂	0.4879	0.4746	0.4473	0.0085	317

Networks and algorithms considered for comparisons were taken from (Wang et al. 2020). In particular, the considered algorithms are: Physarum-Inspired Markov Clustering algorithm (PMCL) and its Evolutionary variant (EPMCL); Regularized Markov Clustering algorithm (RCML); two well-known efficient search algorithms, such as COMBO and MNDP (Modelling with Node Degree Preservation); stochastic model-based algorithm (KARRER); page rank-based algorithm (PPC); genetic algorithm based on a novel coding scheme (NGACD); multi-objective genetic algorithm (MOGA/N); two multi-objective evolutionary algorithms (MOEA/D and RMOEA); and two embedding-based algorithms (GEMSEC and DANMF). It is worth to underline that no details are provided in the cited paper about the exper-

imental protocol adopted, with particular reference to the fixed generations number.

By inspecting the results shown in the Table 11, it emerges that the two algorithms PMCL and EPMCL outperform all compared algorithms and with a considerable gap in terms of modularity found. These better performances are due to the combination between an evolutionary method and Markov chain-based dynamic process, both well known to be efficient search methodologies. In particular in EPMCL, along with an evolutionary approach to optimize the iterative updating, pruning and transition process of the Markov Clustering (MCL) algorithm, a genetic algorithm is also used to find the best parameter combination. At any rate, OPT- IA always stays on the top three ranking, on the *Les Miserables* and

Table 11 Comparison of the two variants of OPT- IA on three real networks. Networks and algorithms were taken from (Wang et al. 2020)

ALGORITHMS	LES MISERABLES	WORD ADJACENCIES	NETWORK SCIENCE
OPT- IA ₁	0.567	0.313	0.6281
OPT- IA ₂	0.567	0.310	0.6713
LOUVAIN	0.565	0.289	0.9597
EPMCL	0.634	0.440	0.9820
PMCL	0.615	0.376	0.971
RMCL	0.465	-0.112	0.962
COMBO	0.560	0.302	0.959
MNDP	0.543	0.271	0.833
KARRER	0.457	-0.104	0.640
PPC	0.454	0.255	0.779
NGACD	0.555	0.224	-
MOGA/N	0.546	0.069	0.881
MOEA/D	0.547	0.269	0.906
RMOEA	0.443	0.080	0.954
GEMSEC	0.476	0.190	0.758
DANMF	0.285	0.041	0.556

Word Adjacencies networks, finding higher modularity values than LOUVAIN and COMBO, for instances. As expected, however, the performances of OPT- IA decrease on the last network, which is larger than the first two, and this is due, as highlighted many times, to the randomness of the algorithm that would require larger number of generations. By increasing the number of generations, very likely, the gap with the other algorithms would narrow considerably. Furthermore, these experiments confirm to us how the variant of OPT- IA that allows to select elements with same fitness (OPT- IA₂) works better on large networks. It is worth to stress once again that although OPT- IA is fully guided by random rules and operators, without any deterministic and solution improvement approach, all presented results have shown and proved its efficiency, its robustness, and its competitiveness with respect to many optimization algorithms, which instead are much more sophisticated and refined in terms of search strategy.

In Fig. 10, finally, are displayed the communities detected by OPT- IA on the *American College Football* (plot a), *Books about US Politics* (plot b) and *C. elegans MRN* (plot c) networks, respectively.

5.1 Functional sensitivity analysis

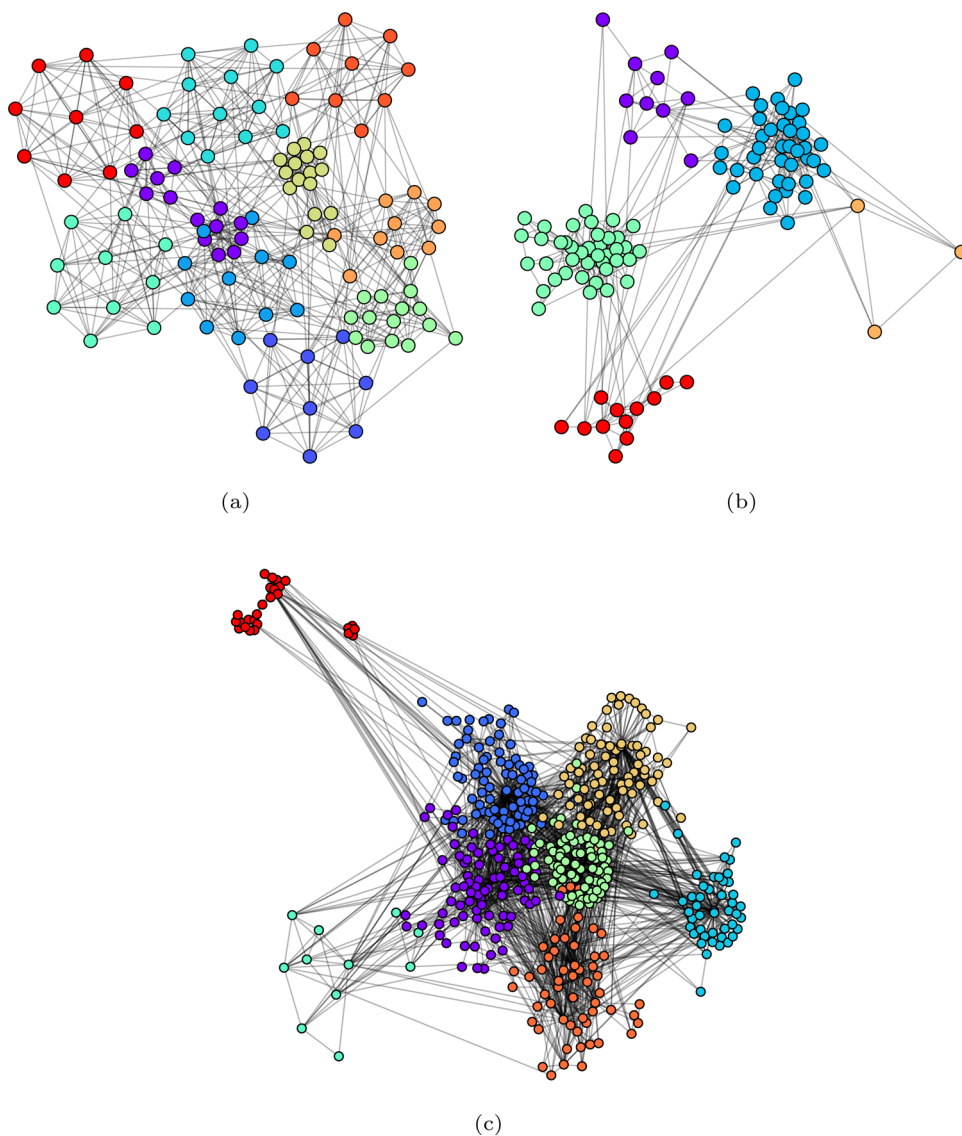
Although modularity is the commonly used evaluation metric, it tells very little about how similar the detected communities are when compared to the original/target ones. Furthermore, an important limitation in modularity optimization is that it can fail in identifying smaller communities, due to the degree of interconnectivity of the communities (Fortunato and Barthelemy 2007). To this end, we conducted

a second experimental step, using synthetic networks generated by *LFR* algorithm proposed in Lancichinetti et al. (2008); Lancichinetti and Fortunato (2009). The aims of this second experiment are to analyse the convergence behaviour of OPT- IA in different complexity scenarios, thanks to the diverse network features which can be generated, and, most importantly, by inspecting how good and similar are the communities uncovered by OPT- IA with respect to the target ones. Obviously, since all networks are artificially generated, their community structures are known. It is important to stress how this benchmark faithfully reproduces the key features of real graphs communities, affirming therefore its validation.

The networks generated for this experiment were, respectively, created with 300, 500, 1000, 2000 and 3000 nodes, each of them with average degree 15 and 20, and the maximum degree equal to 50. Furthermore, for each instance, we set: $\tau_1 = 2$ as exponent of the degree distribution; $\tau_2 = 1$ as the distribution of community sizes; $min_c = 10$ and $max_c = 50$, respectively, as minimum, and maximum of the communities' size. All experiments were conducted at the varying of the mixing parameter μ_t , which identifies the relationship between the node's external and internal degree with respect to its community: the greater the value of μ_t , the greater is number of edges that a node shares with nodes outside of its communities. In order to analyse the performances of OPT- IA on several scenarios, the mixing parameter was made to vary in the range $\{0.1, 0.2, \dots, 0.8\}$.

Once the synthetic networks were generated, each with different features, a functional sensitivity analysis was conducted using the well-known community structure similarity metrics, such as: (1) *Normalized Mutual Information (NMI)* (Danon et al. 2005) that measures the amount of information

Fig. 10 Communities obtained from *American College Football* (a), *Books about US Politics* (b) and *C. elegans MRN* (c)



correctly extracted, and allows for assessing how similar the detected communities are to real ones; (2) *Adjusted Rand Index (ARI)* (Hubert and Arabic 1985), which focuses on pairwise agreement, that is for each possible pair of elements it evaluates how similarly the two partitions treat them; and, finally, (3) *Normalized Variation of Information (NVI)* (Meilă 2007), expressed using the Shannon entropy, which measures the amount of information lost and gained in changing from one clustering to another one: sum of the information needed to describe C , given C' , and the information needed to describe C' given C . Note that NMI is the mostly used in community detection tasks. It is important also to point out that the closer to 1 the NMI and ARI values are (closer to 0 for the NVI value, instead), the more similar the uncovered communities are to the target ones.

In Fig. 11 we can see the graphics of NMI (top plots), ARI (middle plot) and NVI (bottom plot) indexes for the

LFR benchmarks with 300, 500 and 1000 vertices. By analysing each plot, it is possible to note how the NMI and ARI curves remain on high values (> 0.70) for $\mu_t \leq 0.6$ and $\mu_t \leq 0.5$, respectively, whilst the NVI curve remains on low values for $\mu_t \leq 0.5$. This proves that OPT-IA is able to uncover communities roughly closer to the original ones. The two NMI and ARI curves instead begin to decrease, and the NVI curve increases, as the graph begins to get more dense ($\mu_t > 0.6$); in this case OPT-IA detects community structures not well defined.

In Fig. 12, instead, it is displayed the functional sensitivity analysis conducted on the synthetic networks with 2000 (left column), 3000 (middle column) and 5000 (right column) vertices. By inspecting these plots, it is possible to assert that the NMI curves still continue to remain high for $\mu_t \leq 0.5$, whilst decrease at the increasing of the mixed parameter, corresponding then to more dense networks. It is important to

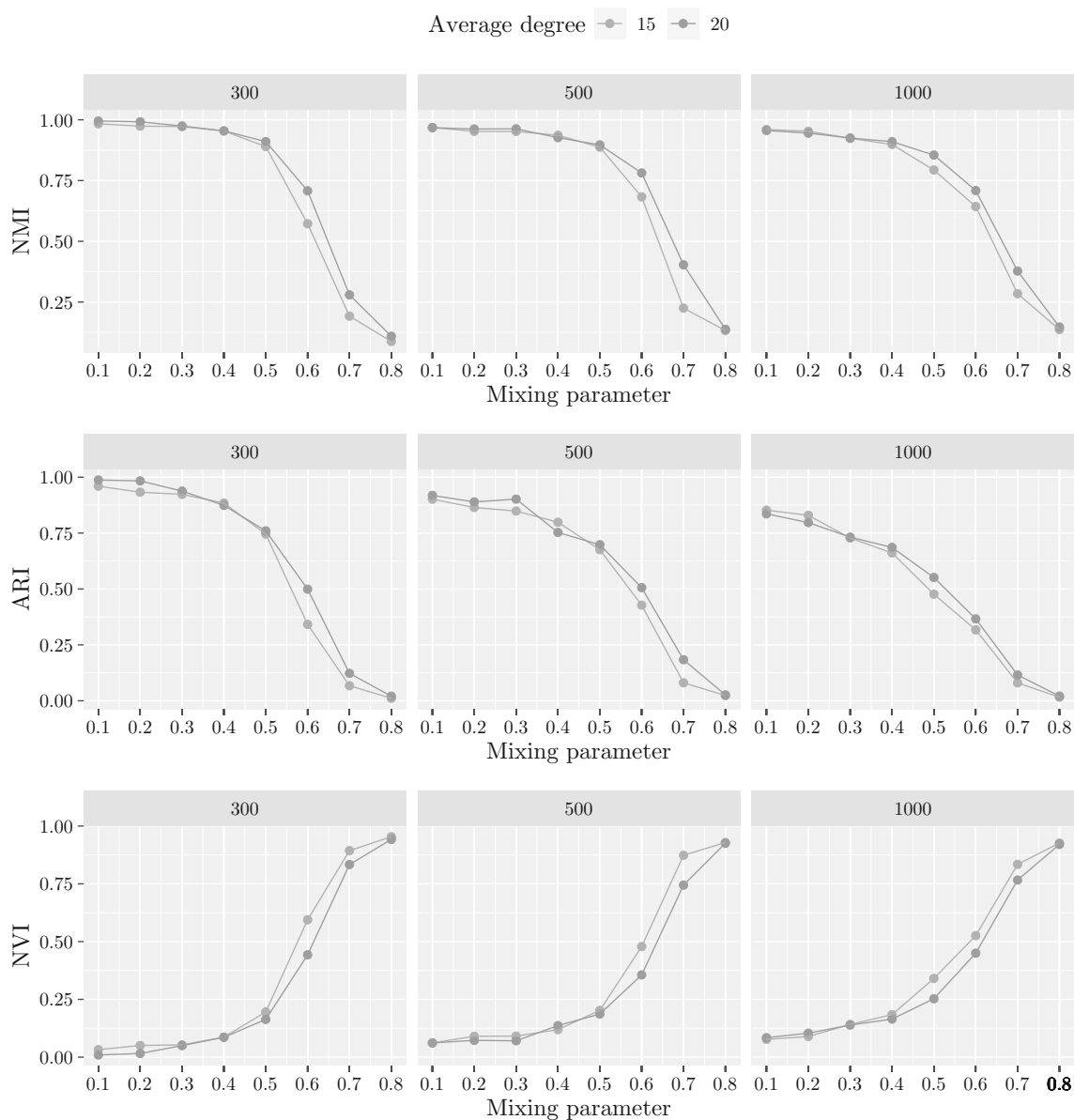


Fig. 11 The curves of the Normalized Mutual Information (top), Adjusted Rand Index (middle) and Normalized Variation of Information (bottom) indexes, performed on 300, 500 and 1000 nodes synthetic networks

note that the behaviour of the *NMI* curves on the middle and right plots, that is on 3000 and 5000 vertices, where the *NMI* curve values are on average high (≥ 0.58), highlight the limit of OPT-IA due to its randomness, and, consequently, pointing out the need to have longer iterations for solving larger networks. Same statements can also be made for the *ARI* and *NVI* plots for 3000 and 5000 vertices. On the other hand, however, these high *NMI* curve values obtained by OPT-IA prove the ability of the algorithm to detect communities as similar to the target ones as possible. The *ARI* curve values (middle plot, left column) remain acceptable for all $\mu_t \leq 0.4$ whilst decrease at higher values of μ_t . As we have repeatedly said, this is obviously caused by the fully random search at

the basis of the algorithm that requires longer time to convergence towards good solutions. Same analysis can be also done for the *NVI* curves (bottom plot, left column). This is confirmed by looking at the convergence behaviours shown in Sect. 4.3 (Fig. 5), where in each of them the relative convergence is represented by a monotonically increasing curve with respect to the number of generations.

6 Conclusions

A novel immune algorithm was designed and developed for community detection, which represents one of the most

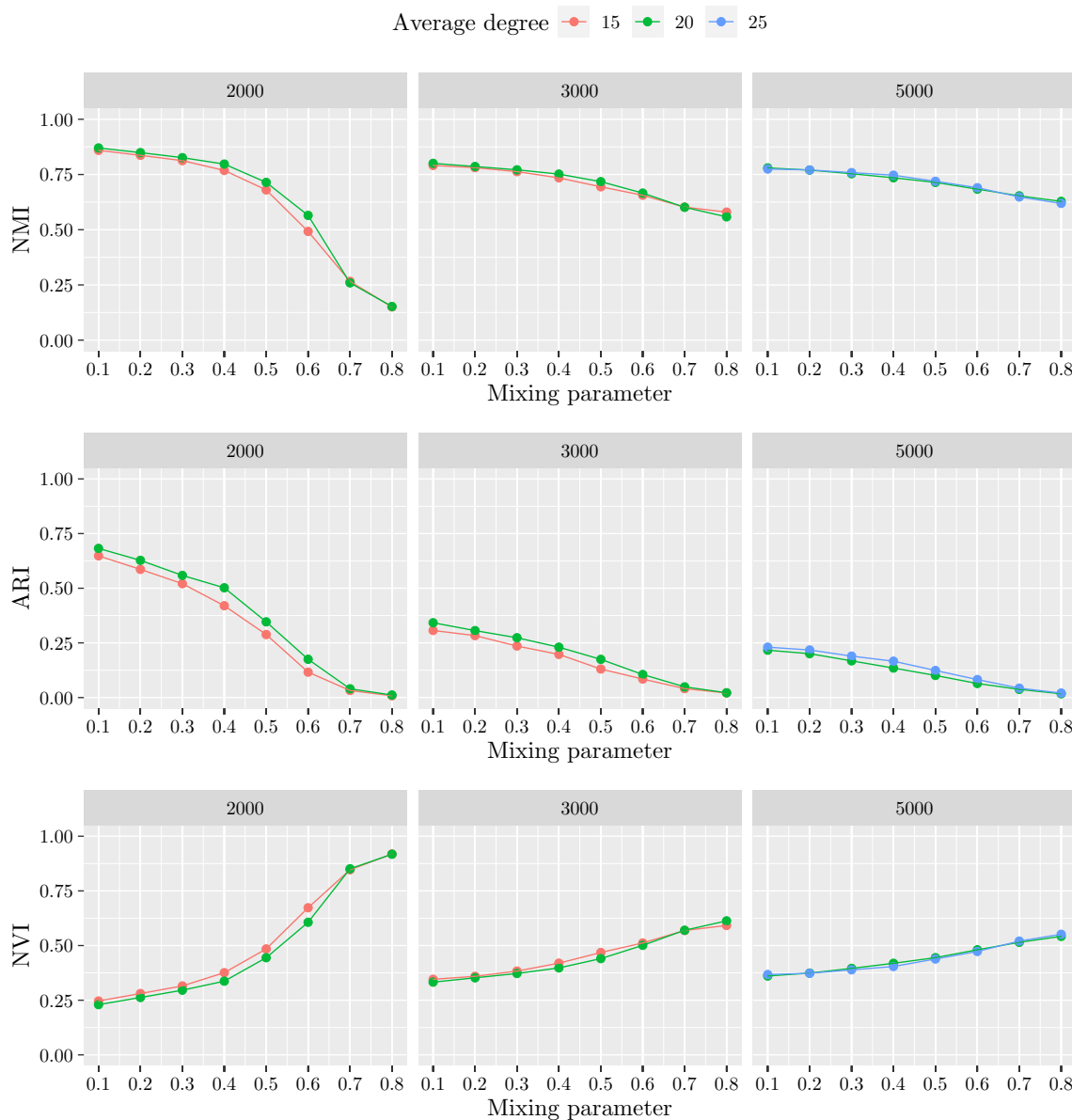


Fig. 12 The curves of the Normalized Mutual Information (top), Adjusted Rand Index (middle) and Normalized Variation of Information (bottom) indexes, performed on 2000, 3000 and 5000 nodes synthetic networks

influential problems in many research areas. The proposed algorithm, called OPT- IA, is inspired by the clonal selection principle, and consequently is based on three main immune operators, such as cloning, hypermutation and stochastic aging, whose combination allows the algorithm to perform in a proper way the exploration and exploitation of the search space. The presented algorithm is entirely blind to the features of the problem being mainly based on a pure random search of the solutions combined with stochastic operators. In this way, the algorithm can easily jump out from local optimal and perform an extensive exploration thanks to the high

diversity in the population produced by the several stochastic strategies developed.

The reliability and efficiency of OPT- IA in community detection has been tested on several social and biological networks, each of them showing different complexity and dimensions. By inspecting the results of all the performed experiments, it clearly emerges the efficiency and reliability of OPT- IA, as well as its robustness as proven in the analysis of the convergence quality and learning capability. Having included a random search strategy in OPT- IA along with several stochastic operators, it allows the algorithm to carry out a careful and at the same time vast exploration of the search

space. An analysis on the computational time complexity has been also conducted by making use of the *Time-To-Target plots* ($TTT - plots$), which confirm that OPT- IA albeit it needs more iterations compared to other algorithms (due to its pure randomness), it reaches, however, the best solutions in acceptable times. Indeed, an asymptotic complexity analysis has been also presented, from which it is possible to claim that the upper bound for its running time is $\mathcal{O}(n^3)$.

In order to assess OPT- IA with respect to the state of the art in community detection, the algorithm was compared against about twenty different heuristics and metaheuristics, included one Hyper-Heuristic methodology. From these comparisons, it appears very clear how the proposed algorithm strictly outperforms most of the compared algorithms, except for the Hyper-Heuristic where instead the performances can be considered comparable in the overall. In particular, the main difference of the performances between the Hyper-Heuristic and OPT- IA is given on the values of the average of the best solutions found on 30 independent runs. However, this is reasonably foreseeable since the main feature of Hyper-Heuristic methods is the combination of several heuristics, efficient on the problem to be tackled, in order to exploit the strength of one to overcome the weaknesses of the others, whilst OPT- IA is an algorithm entirely based on random search combined with pure stochastic operators.

In conclusion, all the outcomes and the analysis conducted prove the reliability of the proposed random search, making OPT- IA comparable with sophisticated algorithms, especially on networks that are not too much dense, such as biological networks for instance. Obviously, the limit of the random search, and therefore of OPT- IA, is the need to have a large number of generations to converge to acceptable solutions when tackling with wide networks (e.g. $|V| \geq 3000$). However, since the solution search process is entirely guided by randomness and stochastic operators, and therefore without any deterministic approach neither any information on the features of the network (OPT- IA is fully blind algorithm), it allows on the other hand to be easily adapted and applied in dynamic network scenarios and in situations of high uncertainty.

Acknowledgements The data shown in Table 3 was presented previously in Spampinato et al. (2019). The authors would like to thank the associate editor and anonymous reviewers for their helpful feedback that measurably improved the manuscript.

Author Contributions Experiments and data collection were performed by RAS and AGS, whilst the data analysis was performed by RAS and MP. The first draft of the manuscript was written by MP and AGS, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Catania within the CRUI-CARE Agreement. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of Interest The authors have no relevant financial or non-financial interests to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aiex RM, Resende MGC, Ribeiro CC (2002) Probability distribution of solution time in grasp: an experimental investigation. *J Heuristics* 6:343–373
- Aiex RM, Resende MGC, Ribeiro CC (2007) Ttt-plots: a perl program to create time-to-target plots. *Optim Lett* 1:355–366
- Atay Y, Koc I, Babaoglu I, Kodaz H (2017) Community detection from biological and social networks: a comparative analysis of meta-heuristic algorithms. *Appl Soft Comput* 50:194–211
- Bickel PJ, Chen A (2009) A nonparametric view of network models and newman-girvan and other modularities. *Proc Natl Acad Sci* 106(50):21068–21073
- Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech Theory Exp* 10:10008–10019
- Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang D-U (2006) Complex networks: Structure and dynamics. *Phys Rep* 424(4–5):175–308
- Brandes U, Delling D, Gaertler M, Gorke R, Hofer M, Nikoloski Z, Wagner D (2007) On modularity clustering. *IEEE Trans Knowl Data Eng* 20(2):172–188
- Bu D, Zhao Y, Cai L, Xue H, Zhu X, Lu H, Zhang J, Sun S, Ling L, Zhang N, Li G, Chen R (2003) Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucl Acids Res* 31(9):2443–2450
- Cao C, Ni Q, Zhai Y (2015) A novel community detection method based on discrete particle swarm optimization algorithms in complex networks. In: 2015 IEEE congress on evolutionary computation (CEC), pp 171–178. IEEE
- Cattle PPI. net. ((last accessed date: October 5, 2015)). [http://biit.cs.ut.ee/graphweb/exampleInput/Cattle_protein_interactions_\(IntAct\).txt](http://biit.cs.ut.ee/graphweb/exampleInput/Cattle_protein_interactions_(IntAct).txt)
- Chen J, Yuan B (2006) Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics* 22(18):2283–2290
- Civicoglu P (2012) Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Comput Geosci* 46:229–247

- Coscia M, Giannotti F, Pedreschi D (2011) A classification for community discovery methods in complex networks. *Stat Anal Data Min ASA Data Sci J* 4(5):512–546
- Cutello V, Nicosia G, Pavone M, Timmis J (2007) An immune algorithm for protein structure prediction on lattice models. *IEEE Trans Evol Comput* 11(1):101–117
- Cutello V, Nicosia G, Pavone M (2007) An immune algorithm with stochastic aging and kullback entropy for the chromatic number problem. *J Comb Optim* 14(1):9–33
- Cutello V, Nicosia G, Pavone M, Prizzi I (2010) Protein multiple sequence alignment by hybrid bio-inspired algorithms. *Nucl Acids Res* 39(6):1980–1992
- Cutello V, Oliva M, Pavone M, Scollo RA (2019) An immune meta-heuristics for large instances of the weighted feedback vertex set problem. In: *IEEE symposium series on computational intelligence*, pp 1–8. IEEE Press
- Cutello V, Oliva M, Pavone M, Scollo RA (2020) A hybrid immunological search for the weighted feedback vertex set problem. In: *International conference on learning and intelligent optimization, lecture notes in computer science*, vol 11968, pp 1–16. Springer
- Danon L, Díaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. *J Stat Mech Theory Exp* 2005(09):09008–09008
- Deco G, Corbetta M (2011) The dynamical balance of the brain at rest. *Neuroscientist* 17(1):107–123
- Di Stefano A, Vitale A, Cutello V, Pavone M (2016) How long should offspring lifespan be in order to obtain a proper exploration? In: *IEEE symposium series on computational intelligence*, pp 1–8. IEEE Press
- Doush IA, Alrashdan WB, Al-Betar MA, Awadallah MA (2020) Community detection in complex networks using multi-objective bat algorithm. *Int J Math Modell Numer Optim* 10(2):123–140
- Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. *Phys Rev E* 72(2):027104
- Erol OK, Eksin I (2006) A new optimization method: big bang-big crunch. *Adv Eng Softw* 37(2):106–111
- Feo TA, Resende MG, Smith SH (1994) A greedy randomized adaptive search procedure for maximum independent set. *Oper Res* 42(5):860–878
- Fortunato S (2010) Community detection in graphs. *Phys Rep* 486(3–5):75–174
- Fortunato S, Barthelemy M (2007) Resolution limit in community detection. *Proc Natl Acad Sci* 104(1):36–41
- Fouladvand S, Osareh A, Shadgar B, Pavone M, Sharafi S (2017) Densa: an effective negative selection algorithm with flexible boundaries for self-space and dynamic number of detectors. *Eng Appl Artif Intell* 62:359–372
- Girvan M, Newman ME (2002) Community structure in social and biological networks. *Proc Natl Acad Sci* 99(12):7821–7826
- Glover F (1977) Heuristics for integer programming using surrogate constraints. *Decis Sci* 8(1):156–166
- Goldberg DE (1989) *Genetic algorithms in search optimization and machine learning*. Addison-Wesley Publishing Company
- Holland J (1975) *Adaptation in natural and artificial systems: an introductory analysis with application to biology. Control and artificial intelligence*. MIT Press, Cambridge
- Hubert L, Arabic P (1985) Comparing partitions. *J Classif* 2:193–218
- Jansen T, Zarges C (2011) On benefits and drawbacks of aging strategies for randomized search heuristics. *Theoret Comput Sci* 412(6):543–559
- Jansen T, Zarges C (2011) On the role of age diversity for effective aging operators. *Evol Intel* 4(2):99–125
- Jaynes ET (2003) *Probability theory: the logic of science*. Cambridge University Press, Cambridge
- Krebs V (2008) A network of books about recent US politics sold by the online bookseller Amazon.com. <http://www.orgnet.com>
- Kullback S (1959) *Information theory and statistics*. Wiley publication in mathematical statistics, Wiley
- Lancichinetti A, Fortunato S (2009) Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys Rev E* 80:016118
- Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E* 78:046110
- Li W, Kang Q, Kong H, Liu C, Kang Y (2020) A novel iterated greedy algorithm for detecting communities in complex network. *Soc Netw Anal Min* 10:1–17
- Lusseau D, Schneider K, Boisseau O, Haase P, Slooten E, Dawson S (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behav Ecol Sociobiol* 54:396–405
- Martí R, Laguna M, Glover F (2006) Principles of scatter search. *Eur J Oper Res* 169(2):359–372
- Meilä M (2007) Comparing clusterings—an information based distance. *J Multivar Anal* 98:873–895
- Mucha PJ, Onnela J, Porter M (2009) Communities in networks. *Not Am Math Soc* 56:1082–1097
- Naeni LM, Berretta R, Moscato P (2015) Ma-net: A reliable memetic algorithm for community detection by modularity optimization. In: Handa H, Ishibuchi H, Ong Y-S, Tan KC (eds.) *Proceedings of the 18th Asia pacific symposium on intelligent and evolutionary systems*, Vol 1, Springer, pp 311–323
- Newman ME (2003) The structure and function of complex networks. *SIAM Rev* 45(2):167–256
- Newman ME (2004) Analysis of weighted networks. *Phys Rev E* 70(5):056131
- Newman ME (2004) Fast algorithm for detecting community structure in networks. *Phys Rev E* 69(6):066133
- Newman ME (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys Rev E* 74(3):036104
- Newman ME (2006) Modularity and community structure in networks. *Proc Natl Acad Sci* 103(23):8577–8582
- Newman ME (2012) Communities, modules and large-scale structure in networks. *Nat Phys* 8(1):25
- Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. *Phys Rev E* 69(2):026113
- Pavone M, Narzisi G, Nicosia G (2012) Clonal selection: an immunological algorithm for global optimization over continuous spaces. *J Global Optim* 53(4):769–808
- Pizzuti C (2008) Ga-net: a genetic algorithm for community detection in social networks. In: *International conference on parallel problem solving from nature*, Springer, pp 1081–1090
- Poggiolini M, Engelbrecht A (2013) Application of the feature-detection rule to the negative selection algorithm. *Expert Syst Appl* 40(8):3001–3014
- Porter MA, Onnela J-P, Mucha PJ (2009) Communities in networks. *Not AMS* 56(9):1082–1097
- Rain J-C, Selig L, De Reuse H, Battaglia V, Reverdy C, Simon S, Lenzen G, Petel F, Wojcik J, Schächter V, Chemama Y, Labigne A, Legrain P (2001) The protein-protein interaction map of *helicobacter pylori*. *Nature* 409(6817):211–215
- Rashedi E, Nezamabadi-Pour H, Saryzadi S (2009) Gsa: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
- Ronhovde P, Nussinov Z (2009) Multiresolution community detection for megascale networks by information-based replica correlations. *Phys Rev E* 80(1):016109
- Said A, Abbasi RA, Maqbool O, Daud A, Aljohani NR (2018) Cc-ga: a clustering coefficient based genetic algorithm for detecting communities in social networks. *Appl Soft Comput* 63:59–70
- Sánchez-Oro J, Duarte A (2018) Iterated greedy algorithm for performing community detection in social networks. *Futur Gener Comput Syst* 88:785–791

- Schellenberger J, Park JO, Conrad TM, Palsson BØ (2010) Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinform* 11(1):213
- Scollo RA, Cutello V, Pavone M (2021) Where the local search affects best in an immune algorithm. In: Baldoni M, Bandini S (eds) *AIXIA 2020 - Advances in artificial intelligence*, vol 12414. *Lecture notes in artificial intelligence science (LNAI)*. Springer, Cham, pp 99–114
- Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27:379–423/623656
- Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of *escherichia coli*. *Nat Genet* 31(1):64
- Shi C, Wang Y, Wu B, Zhong C (2009) A new genetic algorithm for community detection. In: *International conference on complex sciences*, Springer, pp 1298–1309
- Smith SL, Timmis J (2008) An immune network inspired evolutionary algorithm for the diagnosis of parkinson's disease. *Biosystems* 94(1–2):34–46
- Spampinato AG, Scollo RA, Cavallaro S, Pavone M, Cutello V (2019) An immunological algorithm for graph modularity optimization. In: *UK workshop on computational intelligence*, Springer, pp 235–247
- Storn R (1995) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical report*, International Computer Science Institute 11
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
- Stracquadanio G, Greco O, Conca P, Cutello V, Pavone M, Nicosia G (2015) Packing equal disks in a unit square: an immunological optimization approach. In: *IEEE congress on systems immunology, immunoinformatics and immune-computation*, pp 1–5. IEEE Press
- Sundaresan SR, Fischhoff IR, Dushoff J, Rubenstein DI (2007) Network metrics reveal differences in social organization between two fission-fusion species, grevy's zebra and onager. *Oecologia* 151:140–149
- Vitale A, Di Stefano A, Cutello V, Pavone M (2018) The influence of age assignments on the performance of immune algorithms. In: *UK workshop on computational intelligence, advances in computational intelligence systems*, vol 840, pp 16–28. Springer
- Wang Z, Wang C, Li X, Gao C, Li X, Zhu J (2020) Evolutionary Markov dynamics for network community detection. *IEEE Trans Knowl Data Eng* 34(3):1206–1220
- Xenarios I, Rice DW, Salwinski L, Baron MK, Marcotte EM, Eisenberg D (2000) Dip: the database of interacting proteins. *Nucl Acids Res* 28(1):289–291
- Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, pp 65–74
- Yu H, Braun P, Yildirim MA, Lemmens I, Venkatesan K, Sahalie J, Hirozane-Kishikawa T, Gebreab F, Li N, Simonis N, Hao T, Rual J-F, Dricot A, Vazquez A, Murray RR, Simon C, Tardivo L, Tam S, Svrikapa N, Fan C, de Smet A-S, Motyl A, Hudson ME, Park J, Xin X, Cusick ME, Moore T, Boone C, Snyder M, Roth FP, Barabási A-L, Tavernier J, Hill DE, Vidal M (2008) High-quality binary protein interaction map of the yeast interactome network. *Science* 322(5898):104–110
- Zachary WW (1977) An information flow model for conflict and fission in small groups. *J Anthropol Res* 33(4):452–473

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.