

Article

Optimization Algorithms for Detection of Social Interactions

Vincenzo Cutello , Georgia Fargetta , Mario Pavone *  and Rocco A. Scollo 

Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria 6, 95125 Catania, Italy; cutello@unict.it (V.C.); georgia.fargetta@phd.unict.it (G.F.); rocco.scollo@phd.unict.it (R.A.S.)

* Correspondence: mario.pavone@unict.it

Received: 8 April 2020; Accepted: 5 June 2020; Published: 11 June 2020



Abstract: Community detection is one of the most challenging and interesting problems in many research areas. Being able to detect highly linked communities in a network can lead to many benefits, such as understanding relationships between entities or interactions between biological genes, for instance. Two different immunological algorithms have been designed for this problem, called OPT-IA and HYBRID-IA, respectively. The main difference between the two algorithms is the search strategy and related immunological operators developed: the first carries out a random search together with purely stochastic operators; the last one is instead based on a deterministic Local Search that tries to refine and improve the current solutions discovered. The robustness of OPT-IA and HYBRID-IA has been assessed on several real social networks. These same networks have also been considered for comparing both algorithms with other seven different metaheuristics and the well-known greedy optimization LOUVAIN algorithm. The experimental analysis conducted proves that OPT-IA and HYBRID-IA are reliable optimization methods for community detection, outperforming all compared algorithms.

Keywords: community detection; optimization; modularity optimization; complex networks; metaheuristics; immunological-inspired computation

1. Introduction

In last few years, many approaches have been proposed to detect communities in social networks using diverse ways. Community detection is one of the most important research topics in network science and graph analysis. Informally, a community is defined as a set of network's elements that are highly linked within the group and weakly linked to the outside. Modeling and examining complex systems that contain biological, ecological, economic, social, technological, and other information is a very difficult process because the systems used for the real-world data representation contain highly important information, such as social relationships among people or information exchange interactions between molecular structures in a body. For this reason, the study of community structures inspires intense research activities to visualize and understand the dynamics of a network at different scales [1–3]. In order to evaluate the quality of node partitions of a network, the *modularity* is certainly the most used quality index [4].

Taking the modularity (Q) into account as an evaluation measure, community detection can easily be seen as a combinatorial optimization problem, since the problem aims to find a clustering that maximizes Q . It is also a hard optimization problem since Community Detection has been proven to be an \mathcal{NP} -complete problem [5]. Therefore, several search algorithms (both exact and approximate) for clustering problems have been proposed, and, generally, they have been proven to be robust in finding as cohesive as possible communities in large and complex networks [6,7]. It is well known in

the literature that metaheuristics work better than exact methods in large, complex, and uncertainty environments. Indeed, they are approximation methods successfully applied on many hard and complex problems able to find good solutions within reasonable computing times [8].

In this paper, we propose two Immunological Algorithms for the community detection problem, called OPT-IA and HYBRID-IA, respectively. The first is based on a random and blind search, and employs specifically designed stochastic operators to carefully explore the search space, while HYBRID-IA uses a Local Search (LS) technique that deterministically tries to refine the solutions found so far.

The main goal in this paper is to prove the efficiency, robustness, and reliability of the two immune-inspired algorithms in community detection problem, as they have been successfully applied to several other areas and optimization tasks on networks. The efficiency and robustness of both algorithms were tested on several social networks (with different sizes), and, further, a comparison with seven other metaheuristic methods has also been performed in order to assess the reliability of OPT-IA and HYBRID-IA with respect to the state-to-the-art on community detection. In view of such comparisons, we are able to assert that both immunological algorithms outperforms all other methods used for comparison, and find the best modularity in all tested networks. By reducing the comparison to only OPT-IA and HYBRID-IA, it is possible to assert overall that the algorithms are comparable, although the first one, as expected, needs a higher number of generations due to its random and blind search.

Finally, it is important to underline the fact that, differently than all other algorithms used for comparison, which have been tested and refined over the years, OPT-IA and HYBRID-IA are already very competitive and successful, even though they are at a first stage of development in community detection tasks, and still need a deep study of their key parameters and operators.

The rest of the paper is organized as follows. In Section 2, the problem of community detection is introduced and the definition of the modularity measure are formulated. In Section 3, the two immunological algorithms are introduced and we focus the description primarily on their common concepts, whilst the detailed descriptions on their features and operators developed are presented, respectively, in Section 3.1 (OPT-IA) and Section 3.2 (HYBRID-IA). In Section 4, all experiments performed and comparisons done are described in detail, including the used dataset and experimental protocol, and an analysis on the convergence behavior of the two proposed immunological algorithms (OPT-IA and HYBRID-IA). Finally, conclusions and some future research directions are presented in Section 5.

2. Mathematical Definition of Modularity in Networks

The main aims in community detection are to uncover the inherent community structure of a network, that is to say, those groups of nodes sharing common and similar properties. This means, then, to detect groups (*communities* or *modules*) internally strongly connected, and externally weakly connected. Being able to detect community structures is a key and relevant task in many areas (biology, computer science, engineering, economics, and politics) because they allow for uncovering and understanding important information about the function of the network itself and how its elements affect and interact with each other. For instance, in World Wide Web networks, the communities can identify those pages dealing with topics which are related; in biological networks, instead, they correspond to proteins having the same specific function; in social sciences, they can identify circles of friends or people who have the same hobby, or those people who live in the same neighborhood.

In order to evaluate the quality of the uncovered groups in a network, *modularity* is certainly the most used quality index [4]. It is based on the idea that a random graph is not expected to have a community structure; therefore, the possible existence of communities can be revealed by the difference of density between vertices of the graph and vertices of a random graph with the same size and degree distribution. Formally, it can be defined as follows: given an undirected graph $G = (V, E)$, with V

the set of vertices ($|V| = N$), and E the set of edges ($|E| = M$), the modularity of a community is defined as:

$$Q = \frac{1}{2M} \left[\sum_{i=1}^N \sum_{j=1}^N \left(A_{ij} - \frac{d_i d_j}{2M} \right) \delta(i, j) \right], \quad (1)$$

where A_{ij} is the adjacency matrix of G , d_i and d_j are the degrees of nodes i and j respectively, and $\delta(i, j) = 1$ if i, j belong to the same community, 0 otherwise.

As asserted by Brandes et al. in [5], the modularity value for unweighted and undirected graphs lies in the range $[-0.5, 1]$. Then, a low Q value, i.e., close to the lower bound, reflects a bad clustering, and implies the absence of real communities, whilst good groups are identified by a high modularity value that implies the presence of highly cohesive communities. For a trivial clustering, with a single cluster, the modularity value is 0. However, the modularity has the tendency to produce large communities and, therefore, fails to detect communities which are comparatively small with respect to the network [9].

3. Immunological Algorithms

Immunological Algorithms (IA) are among the most used population-based metaheuristics, successfully applied in search and optimization tasks. They take inspiration from the dynamics of the immune system in performing its job of protecting living organisms. One of the features of the immune system that makes it a very good source of inspiration is its ability to detect, distinguish, learn, and remember all foreign entities discovered [10]. Both proposed algorithms, OPT-IA and HYBRID-IA, belong to the special class *Clonal Selection Algorithms* (CSA) [11,12], whose efficiency is due to the three main immune operators: (i) cloning, (ii) hypermutation, and (iii) aging. Furthermore, both are based on two main concepts: antigen (Ag), which represents the problem to tackle, and B cell, or antibody (Ab) that represents a candidate solution, i.e., a point in the solution space. At each time step t , both algorithms maintain a population of d candidate solutions: each solution is a subdivision of the vertices of the graph $G = (V, E)$ in communities. Let $N = |V|$, then a B cell \vec{x} is a sequence of N integers belonging to the range $[1, N]$, where $x_i = j$ indicates that the vertex i has been added to the cluster j . The population, for both algorithms, is initialized at the time step $t = 0$ randomly assigning each vertex i to a group j , with $j \in [1, N]$. Then, just after the initialization step, the fitness function (Equation (1)) is evaluated for each randomly generated element ($\vec{x} \in P^{(t)}$) by using the function $\text{ComputeFitness}(P^{(t)})$. The two algorithms end their evolutionary cycle when the halting criterion is reached. For this work, it was fixed to a maximum number of generations ($MaxGen$).

Among all immunological operators, *cloning* is the only one that is roughly the same between OPT-IA and HYBRID-IA (6th line in Algorithms 1 and 2). This operator simply copies dup times each B cell producing an intermediate population $P^{(clo)}$ of size $d \times dup$. We used a static cloning for both algorithms in order to avoid premature convergences. Indeed, if a number of clones proportional to the fitness value is produced instead, we could have a population of B cells very similar to each other, and we would, consequently, be unable to perform a proper exploration of the search space getting easily trapped in local optima. HYBRID-IA, more in detail, assigns an age to each cloned B cell, which determines how long it can live in the population, from the assigned age until it reaches the maximum age allowed τ_B , a user-defined parameter). Specifically, a random age chosen in the range $[0 : \frac{2}{3}\tau_B]$ is assigned to each clone. In this way, each clone is guaranteed to stay in the population for at least a fixed number of generations ($\frac{1}{3}\tau_B$ in the worst case). The age assignment and the aging operator (see Section 3.2) play a crucial role on HYBRID-IA performances, and any evolutionary algorithm in general because they are able to keep a right amount of diversity among the solutions, thus avoiding premature convergences [13].

3.1. OPT-IA

OPT-IA [14] is a totally blind stochastic algorithm, a particular type of optimization method that generates a set of random solutions for the community detection problem. It is based on

- (i) the *hypermutation operator*, which acts on each clone in order to explore its neighborhood;
- (ii) the *precompetition operator*, which makes it possible to maintain a more heterogeneous population during the evolutionary cycle;
- (iii) the *stochastic aging operator* whose aim is keep high diversity into the population and consequently help the algorithm in escaping from local optima; and, finally,
- (iv) the *selection operator* which identifies the best d elements without repetition of fitness, ensuring heterogeneity into the population.

In Algorithm 1, the pseudocode of OPT-IA, and its key parameters, are described:

Algorithm 1 Pseudo-code of OPT-IA

```

1: procedure OPT-IA( $d, dup, m, P_{die}$ )
2:    $t \leftarrow 0$ 
3:    $P^{(t)} \leftarrow \text{InitializePopulation}(d)$ ;
4:    $\text{ComputeFitness}(P^{(t)})$ 
5:   while  $\neg \text{StopCriterion}$  do
6:      $P^{(clo)} \leftarrow \text{Cloning}(P^{(t)}, dup)$ 
7:      $P^{(hyp)} \leftarrow \text{Hypermutation}(P^{(clo)}, m)$ 
8:      $\text{ComputeFitness}(P^{(hyp)})$ 
9:      $P^{(pre)} \leftarrow \text{Precompetition}(P^{(t)})$ 
10:     $P_a^{(pre)} \leftarrow \text{StochasticAging}(P^{(pre)}, P_{die})$ 
11:     $P^{(t+1)} \leftarrow \text{Selection}(P_a^{(pre)}, P^{(hyp)})$ 
12:     $t \leftarrow t + 1$ ;
13:  end while
14: end procedure

```

The purpose of the hypermutation operator is to carefully explore the neighborhood of each solution in order to generate better solutions from iteration to iteration. It basically performs at most m mutations on each B cell, where m is a user-defined constant parameter. Unlike the other clonal selection algorithms, including HYBRID-IA, the m mutation rate of each element is not determined by a law inversely proportional to the fitness function, but it will be the same for anyone. In this way, possible premature convergences are avoided. In this algorithm, three different types of mutation have been designed, which can act on a single node (local operators) or a set of nodes (global operators):

1. *equiprobability*: randomly select a vertex from the solution and reassign it to a cluster among those existing at that time. Each cluster has the same probability of being selected;
2. *destroy*: randomly select a cluster c_i from the solution \vec{x} , a percentage P in the range $[1\%, 50\%]$, and a cluster c_j in the range $[1, N]$. All vertices in c_i are then moved to the cluster c_j with P probability. Note that, if the cluster c_j does not exist, then a new community is created;
3. *fuse*: randomly select a cluster and assign all its nodes to a randomly selected cluster among those existing.

After the hypermutation operator, the fitness values for all mutated B cell are computed. Three operators act on the population at this point.

Precompetition operator: the primary aim of this operator is to obtain a more heterogeneous population by trying to maintain solutions which have different community numbers in order to

better explore the search space. Basically, this operator randomly selects two different B cells from $P^{(t)}$, and, if they have same community number, the one with a lower fitness value will be deleted with a 50% probability.

StochasticAging operator: this operator helps the algorithm to escape from local optima by introducing diversity in the population. At each iteration, each B cell in $P^{(pre)}$ will be removed with probability P_{die} (a user-defined parameter). Using this type of aging operator, OPT-IA is able to, on one hand, to introduce diversity in the population, which is crucial for jumping out from local optima, and, on the other hand, to have an accurate exploration and exploitation of the neighborhoods.

Selection operator: finally, the selection operator has the task to generate the new population for the next generation made up of the best B cells discovered so far. Therefore, the new population $P^{(t+1)}$ is created by selecting the best d B cells among the survivors in $P_a^{(pre)}$ and hypermutated B cells in $P^{(hyp)}$. It is important to highlight that no redundancy is allowed during the selection: if a hypermutated B cell is candidate to be selected for the new population $P^{(t+1)}$, but it has the same fitness value with someone in $P_a^{(pre)}$, then it will be discarded. This ensures monotonicity in the evolution dynamic.

3.2. HYBRID-IA

A fundamental difference between OPT-IA and HYBRID-IA is that the former, as described above, is a stochastic IA that finds solutions by random methods. The latter, instead, uses a deterministic local search, based on rational choices that refine and improve the solutions found so far. The pseudocode of HYBRID-IA is described in Algorithm 2.

Algorithm 2 Pseudo-code of HYBRID-IA.

```

1: procedure HYBRID-IA( $d, dup, \rho, \tau_B$ )
2:    $t \leftarrow 0$ 
3:    $P^{(t)} \leftarrow \text{InitializePopulation}(d)$ 
4:    $\text{ComputeFitness}(P^{(t)})$ 
5:   while  $\neg \text{StopCriterion}$  do
6:      $P^{(clo)} \leftarrow \text{Cloning}(P^{(t)}, dup)$ 
7:      $P^{(hyp)} \leftarrow \text{Hypermutation}(P^{(clo)}, \rho)$ 
8:      $\text{ComputeFitness}(P^{(hyp)})$ 
9:      $(P_a^{(t)}, P_a^{(hyp)}) \leftarrow \text{Aging}(P^{(t)}, P^{(hyp)}, \tau_B)$ 
10:     $P^{(select)} \leftarrow (\mu + \lambda)\text{-Selection}(P_a^{(t)}, P_a^{(hyp)})$ 
11:     $P^{(t+1)} \leftarrow \text{LocalSearch}(P^{(select)})$ 
12:     $\text{ComputeFitness}(P^{(t+1)})$ 
13:     $t \leftarrow t + 1$ ;
14:   end while
15: end procedure

```

The hypermutation operator developed in HYBRID-IA has the main goal of exploring the neighborhoods of solutions by evaluating how good each clone is. Unlike OPT-IA, the mutation rate is determined through an *inversely proportional* law to the fitness function value of the B cell considered, that is, the better the fitness value of the element is, the smaller the mutation rate will be. In particular, let \vec{x} be a cloned B cell, the *mutation rate* $\alpha = e^{-\rho \hat{f}(\vec{x})}$ is defined as the probability to move a node from one community to another one, where ρ , a user-defined parameter that determines the shape of the mutation rate, and $\hat{f}(\vec{x})$ is the fitness function normalized in the range $[0, 1]$. Formally, the designed hypermutation works as follows: for each B cell, two communities c_i and c_j are randomly chosen ($c_i \neq c_j$): the first is chosen among all existing ones, and the second in the range $[1, N]$. Then, all vertices in c_i are moved to c_j with probability given by α . If a value that does not correspond to any currently existing community is assigned to c_j , a new community c_j is created and added to the existing ones.

The idea behind this approach is to better explore the search space and create and discover new communities by moving a variable percentage of nodes from existing communities. This search method balances the effects of local search (as described below), by allowing the algorithm to avoid premature convergences towards local optima.

The static aging operator in Hybrid-IA acts on each mutated B cells by removing older ones from the two populations $P^{(t)}$ and $P^{(hyp)}$. Basically, let τ_B be the maximum number of generations allowed for every B cell to stay in its population; then, once the age of a B cell exceeds τ_B (i.e., $\text{age}=\tau_B+1$), it will be removed independently from its fitness value. However, an exception may be done for the best current solution, which is kept into the population even if its age is older than τ_B . Such a variant of the aging operator is called *elitist aging operator*. In the overall, the main goal of this operator is to allow the algorithm to escape and jump out from local optima, assuring a proper turnover between the B cells in the population, and producing, consequently, high diversity among them.

After the aging operator, the best d survivors from both populations $P_a^{(t)}$ and $P_a^{(hyp)}$ are selected, in order to generate the temporary population $P^{(select)}$, on the local search will be performed. Such a selection is performed by the $(\mu + \lambda)$ -Selection operator, where $\mu = d$ and $\lambda = (d \times dup)$. The operator identifies the d best elements among the set of offsprings and the old parent B cells, ensuring consequently monotonicity in the evolution dynamics.

The Local Search designed and introduced in HYBRID-IA is the key operator to properly speed up the convergence of the algorithm, and, in a way, drive it towards more promising regions. Furthermore, it intensifies the search and explore the neighborhood of each solution using the well-known *Move Vertex* approach (MV) [15]. The basic idea of the proposed LS is to assess deterministically if it is possible to move a node from its community to another one within its neighbors. The MV approach takes into account the *move gain* that can be defined as the variation in modularity produced when a node is moved from a community to another. Before formally defining the move gain, it is important to point out that the modularity Q , defined in Equation (1), can be rewritten as:

$$Q(c) = \sum_{i=1}^k \left[\frac{l_i}{M} - \left(\frac{d_i}{2M} \right)^2 \right], \tag{2}$$

where k is the number of the found communities; $c = \{c_1, \dots, c_i, \dots, c_k\}$ is the set of communities that is the partitioning of the set of vertice V ; l_i and d_i are, respectively, the number of links inside the community i , and the sum of the degrees of vertices belonging to the i community. The *move gain* of a vertex $u \in c_i$ is, then, the modularity variation produced by moving u from c_i to c_j , that is:

$$\Delta Q_u(c_i, c_j) = \frac{l_{c_j}(u) - l_{c_i}(u)}{M} + d_V(u) \left[\frac{d_{c_i} - d_V(u) - d_{c_j}}{2M^2} \right], \tag{3}$$

where $l_{c_i}(u)$ and $l_{c_j}(u)$ are the number of links from u to nodes in c_i and c_j respectively, and $d_V(u)$ is the degree of u when considering all the vertices V . If $\Delta Q_u(c_i, c_j) > 0$, then moving node u from c_i to c_j produces an increment in modularity, and then a possible improvement. Consequently, the goal of MV is to find a node u to move to an adjacent community in order to maximize ΔQ_u :

$$\operatorname{argmax}_{v \in \text{Adj}(u)} \Delta Q_u(i, j), \tag{4}$$

where $u \in C_i, v \in C_j$ and $\text{Adj}(u)$ is the adjacency list of node u .

For each solution in $P^{(select)}$, the Local Search begins by sorting the communities in increasing order with respect to the ratio between the sum of inside links and the sum of the node degrees in the community. In this way, poorly formed communities are identified. After that, MV acts on each community of the solution, starting from nodes that lie on the border of the community, that is, those that have at least an outgoing link. In addition, for communities, the nodes are sorted with respect to the ratio between the links inside and node degree. The key idea behind LS is

to deterministically repair the solutions which were produced by the hypermutation operator, by discovering then new partitions with higher modularity value. Equation (3) can be calculated efficiently because M and $d_V(u)$ are constants, the terms l_{c_i} and d_{c_i} can be stored and updated using appropriate data structures, while the terms $l_{c_i}(u)$ can be calculated during the exploration of all adjacent nodes of u . Therefore, the complexity of the move vertex operator is linear on the dimension of the neighborhood of node u .

4. Results

In this section, all the experiments and comparisons performed on the two proposed algorithms, OPT-IA and HYBRID-IA, are presented, in order to assess their efficiency and robustness in detecting highly linked communities. Different social networks have been taken into account for the experimental analyses conducted on the two algorithms, and for their comparison with other efficient metaheuristics which are present in literature.

The instances are well-known networks used for the community detection problem and their characteristics are summarized and reported in Table 1. *Gravy's Zebra* [16] is a network created by Sundaresan et al. in which a link between two nodes indicates that a pair of zebras appeared together at least once during the study. In *Zachary's Karate Club* [17] network, collected by Zachary in 1977, a node represents a member of the club and an edge represents a tie between two members of the club. *Bottlenose Dolphins* [18] is an undirected social network of dolphins where an edge represents a frequent association. *Books about US Politics* [19] is a network of books sold, compiled by Krebs, where edges represent frequent co-purchasing of books by the same buyers. Another network considered is *American College Football* [1], a network of football games between colleges. *Jazz Musicians* [20] is the collaboration network between Jazz musicians. Each node is a Jazz musician and an edge denotes that two musicians have played together in a band. These specific networks have been chosen because they are the most used as test benches and, consequently, allowed us to compare our algorithms with several others and, in particular, with different metaheuristics. Although there exist bigger networks in literature, large and meaningful comparisons are more difficult to develop.

For all experiments carried out and presented in this section, both OPT-IA and HYBRID-IA maintain a population of B cells of size $d = 100$, whereas the number of generated clones depends upon the approach used. Since OPT-IA needs high variability, $dup = 10$ has been set for larger networks ($|V| \geq 100$), and $dup = 4$ for smaller ones ($|V| < 100$); $dup = 2$, has been, instead, set in HYBRID-IA for all tested network types.

Table 1. The social networks used in the experiments.

<i>Name</i>	$ V $	$ E $
Grevy's Zebras [16]	28	111
Zachary's Karate Club [17]	34	78
Bottlenose Dolphins [18]	62	159
Books about US Politics [19]	105	441
American College Football [1]	115	613
Jazz Musicians Collaborations [20]	198	2742

Moreover, the m and P_{die} parameters in OPT-IA have been set to 1 and 0.02, respectively; however, for HYBRID-IA, ρ and τ_B have been set to 1.0 and 5. Note that all parameters of OPT-IA have been determined through careful tuning experiments, whilst those of HYBRID-IA have been identified both from the knowledge learned by previous works [11,12], and from a preliminary and not in-depth experiments. As described above, the maximum number of generations has been taken into account as stopping criterion. For all experiments performed in this research work, it was fixed $MaxGen = 1000$ for OPT-IA, and $MaxGen = 100$ for HYBRID-IA. Note that a higher iterations number has been

considered for OPT-IA due to its blind search, which, for obvious reasons, requires larger time steps to reach acceptable solutions.

Initially, the experimental analysis has been focused on the inspection of the efficiency of both algorithms, OPT-IA and HYBRID-IA, in terms of convergence and solution quality found. In Figure 1, the convergence behavior of both OPT-IA and HYBRID-IA on the *Books about US Politics* network is showed. In this plot, the curves represent the evolution of the best and average fitness of the population; the standard deviation of the fitness values of the population is superimposed onto the average fitness and gives an idea about how heterogeneous the elements in the population are.

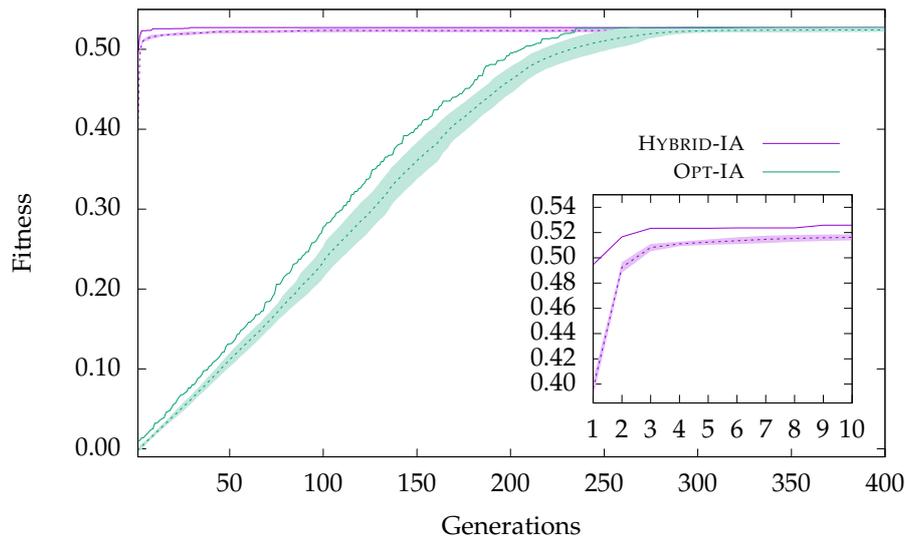


Figure 1. Convergence behavior of OPT-IA and HYBRID-IA on the *Books about US Politics* network.

From Figure 1, one can note how OPT-IA converges more slowly towards the best solution, as expected, always keeping a certain variability within the population. This allows the algorithm to better explore the search space. When the population is composed of very different elements, i.e., when the standard deviation is high, the algorithm discovers new solutions, significantly improving the current best solution. However, after about 250 generations, OPT-IA reaches the optimal solution and the curves (best and average fitness) tend to overlap. Moreover, the achievement of the optimal solution helps the creation of better clones, reducing the variability of the population. Unlike OPT-IA, HYBRID-IA converges easily thanks to the local search applied to the elements after the selection phase. As can be noted from the inset plot in Figure 1, HYBRID-IA reaches the optimal solution after a few generations. Even in this case, once the best solution is reached, the population follows the same trend of the curve of the best fitness, and both curves continue (almost) as a single line. If on one hand the local search helps to quickly discover good solutions, on the other, it reduces the diversity inside the population, reducing then the exploration of the search space. In particular, as demonstrated by the worst value found in the *Jazz Musicians* network, reported in Table 2, HYBRID-IA prematurely converges towards local optima, from which it will hardly be able to get out. At the end of the analysis of Figure 1, it is possible to conclude that the stochastic operators designed in OPT-IA guarantee an excellent and large exploration of the search space, but with the disadvantage of requiring a longer evolution time; however, the local search developed in HYBRID-IA, and relative sorting criteria, allow for quickly discovering good solutions to exploit during the evolutionary process.

In order to evaluate the performances and reliability of both immunological algorithms with respect the state-of-the-art, a wide comparison has been performed with several metaheuristics, and a well-known deterministic algorithm, on the all dataset reported in Table 1. In particular, OPT-IA and HYBRID-IA have been compared with: LOUVAIN [21], a greedy optimization method; HDSA, a Hyper-Heuristics Differential Search Algorithm based on the migration of artificial

superorganisms [22]; BADE, an improved Bat Algorithm based on Differential Evolution algorithm [23,24]; SSGA, a Scatter Search [25,26] based on Genetic Algorithm; BB-BC, a modified Big Bang–Big Crunch algorithm [27]; BA, an adapted Bat Algorithm for community detection [28]; GSA, a Gravitational Search Algorithm re-designed for solving the community detection problem [29]; and MA-NET, a Memetic Algorithm [30]. For all algorithms, we considered $MaxGen = 100$, except for OPT-IA, as described above, and MA-NET whose stopping criterion has been set to 30 generations without improvement (see [30] for details). All results of the above methods, as well as the experimental protocol, have been taken from Atay et al. [31], except for LOUVAIN and MA-NET.

Table 2. Comparison of OPT-IA and HYBRID-IA on social networks with reference algorithms. For HDSA, BADE, SSGA, BB-BC, BA, and GSA the results are calculated over 30 independent runs, while for MA-NET over 50 independent runs.

Networks	Algorithms										
	LOUVAIN	OPT-IA	HYBRID-IA	HDSA	BADE	SSGA	BB-BC	BA	GSA	MA-NET	
Zebra	Best	0.2768	-								
	Mean	-	0.2768	0.2768	0.2768	0.2768	0.2768	0.2766	0.2768	0.2768	-
	Worst	-	0.2768	0.2768	0.2768	0.2768	0.2768	0.2761	0.2768	0.2768	-
	σ	-	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	-
	k	4	4	4	4	4	4	4	4	4	-
Karate Club	Best	0.4188	0.4198	0.420							
	Mean	-	0.4198	0.4198	0.4198	0.4188	0.4198	0.4196	0.4133	0.4170	0.419
	Worst	-	0.4198	0.4198	0.4198	0.4156	0.4198	0.4188	0.3946	0.4107	-
	σ	-	0.0000	0.0000	0.0000	0.0018	0.0000	0.0004	0.0105	0.0037	0.002
	k	4	4	4	4	4	4	4	4	4	4
Dolphins	Best	0.5185	0.5285	0.5285	0.5285	0.5268	0.5257	0.5220	0.5157	0.4891	0.529
	Mean	-	0.5285	0.5273	0.5282	0.5129	0.5200	0.5141	0.4919	0.4677	0.523
	Worst	-	0.5268	0.5220	0.5276	0.4940	0.5156	0.5049	0.4427	0.4517	-
	σ	-	0.0003	0.0009	0.0005	0.0120	0.0040	0.0068	0.0289	0.0155	0.004
	k	5	5	5	5	4	5	5	4	6	5
Political Books	Best	0.5205	0.5272	0.5272	0.5272	0.5239	0.5221	0.4992	0.5211	0.4775	0.527
	Mean	-	0.5267	0.5270	0.5272	0.5178	0.5203	0.4914	0.5020	0.4661	0.526
	Worst	-	0.5063	0.5246	0.5272	0.5137	0.5167	0.4799	0.4815	0.4558	-
	σ	-	0.0028	0.0005	0.0000	0.0042	0.0024	0.0084	0.0149	0.0079	0.002
	k	4	5	5	5	4	5	9	3	5	5
Football	Best	0.6046	0.6046	0.6046	0.6046	0.5646	0.5330	0.5171	0.5523	0.4175	0.605
	Mean	-	0.5989	0.6039	0.6033	0.5513	0.5277	0.5061	0.5272	0.4032	0.601
	Worst	-	0.5736	0.6031	0.6019	0.5430	0.5189	0.4986	0.4742	0.3905	-
	σ	-	0.0078	0.0007	0.0009	0.0085	0.0057	0.0069	0.0325	0.0109	0.003
	k	10	10	10	10	11	6	10	7	5	10
Jazz Musicians	Best	0.4451	0.4451	0.4451	-	-	-	-	-	-	0.445
	Mean	-	0.4449	0.4450	-	-	-	-	-	-	0.445
	Worst	-	0.4449	0.4446	-	-	-	-	-	-	-
	σ	-	0.0001	0.0002	-	-	-	-	-	-	0.000
	k	4	4	4	-	-	-	-	-	-	4

The comparison performed on all dataset is reported in Table 2, where we show for each algorithm (where possible) the *best*, *mean*, and *worst* values of the Q modularity; standard deviation (σ), and, finally, the created communities number (k). Furthermore, whilst the experiments for OPT-IA and HYBRID-IA were performed on 100 independent runs, for all other compared algorithms, only 30 independent runs have been considered, excepts for MA-NET with 50 runs. It is important to note that, for the MA-NET outcomes, the values reported, and taken from [30], have been rounded to third decimal places unlike the others that are instead based on four decimals. Obviously, all 10 algorithms optimize the same fitness function reported in Equation (1) and rewritten in a simpler way in Equation (2).

From Table 2, it is possible to note that all algorithms reach the optimal solution in the first two networks *Zebra* and *Karate Club*, except LOUVAIN, which fails on the second; on all the other network instances, both OPT-IA and HYBRID-IA outperform all other compared algorithms, matching their best values only with HDSA and MA-NET ones. It is important to point out, which proves even more the efficiency of the two proposed immunological algorithms, how the *mean* values obtained by

OPT-IA and HYBRID-IA, on all tested networks, are better than the best modularity found by the other algorithms, such as: BADE, SSGA, BB-BC, BA, and GSA; even on the *Dolphins* and *Football* networks, the worst modularity value obtained by OPT-IA is equal to or greater than the best one obtained by the same algorithms. On the *Dolphins* network, OPT-IA reaches a better mean value than HYBRID-IA, HDSA, and MA-NET, since, because of its random/blind exploration of the search space, it jumps out from local optima more easily than the other three. The opposite behavior of OPT-IA occurs when the size and complexity of the networks increase. In such a case, it obviously needs more generations to converge towards the optimal solutions and this is highlighted by the mean value and the standard deviation obtained for *Political Books* and *Football*. Note that, with longer generations, OPT-IA finds roughly the same mean values as HYBRID-IA.

HYBRID-IA shows more stable results on all tested networks than OPT-IA, obtaining lower standard deviation values on all instances. On *Dolphins* network, HYBRID-IA has a mean value slightly lower than OPT-IA and HDSA, while in *Political Books* lower only than HDSA. As described above, this is due to the local search that leads the algorithm to a premature convergence towards local optima, obtaining the lowest worst value. Furthermore, HDSA is a hyper-heuristic which uses a genetic algorithm and scatter search to create the initial population for the differential search algorithm, speeding up the convergence of the algorithm, and reducing the spread of results. In *Jazz Musicians* network, both OPT-IA and HYBRID-IA algorithms obtain similar results, comparable to those obtained by MA-NET. Finally, if we focus on the comparison with only the deterministic LOUVAIN algorithm, both immunological algorithms outperform it in almost all networks (3 out of 6). In conclusion, from the experimental analysis, it is possible to assert that both OPT-IA and HYBRID-IA perform very well on networks (large and small) considered for the experiments and are competitive with the state-of-the-art in terms of efficiency and robustness. Furthermore, inspecting the found mean values, we can see that both algorithms show themselves to be reliable optimization methods in detecting highly linked communities.

4.1. Large Synthetic Networks

In order to assess the scalability of HYBRID-IA, as last step of this research work, we considered larger synthetic networks with 1000 and 5000 vertices. The clear advantages of using them are given by the knowledge of the communities and consequently by the possibility of evaluating the goodness of the detected communities, as well as the possibility of testing the algorithm on different scenarios and complexities. The validity of this benchmark is given by faithfully reproducing the keys features of real graphs communities. The new benchmark was produced using the *LFR* algorithm proposed by Lancichinetti and Fortunato in [32]. Two different network instances were considered with $|V| = 1000$ and $|V| = 5000$, respectively, considering $k = 20$ as average degree and $k_{max} = 50$ as maximum degree. For both values of $|V|$, we considered $\tau_1 = 2$ as exponent of the degrees distribution and $\tau_2 = 1$ for the one of the communities' sizes. Moreover, about the community dimension, we fixed $min_c = 10$ and $max_c = 50$, respectively, minimum and maximum sizes. Finally, the mixing parameter μ_t was varied from 0.1 to 0.7, which identifies the relationships percentage of a node with those belonging to different communities. Therefore, the larger the value of μ_t , the larger are the relationships that a vertex has with other nodes outside its community. More details on the *LFR* algorithm, the key parameters, and the generated benchmark can be found in [32,33].

Thanks to the advantages offered by the synthetic networks, we took into account a new evaluation metric in order to confirm the efficiency of the proposed HYBRID-IA in detecting strong communities. In particular, we considered the *Normalized Mutual Information (NMI)* [34], which is a widely used measure to compare community detection methods since it discloses the similarity between the true community and the detected community structures. Thus, while the modularity allows for assessing how cohesive the detected communities are, the *NMI* allows for evaluating how similar they are with respect to the real ones.

In Table 3, the outcomes of HYBRID-IA on these new synthetic datasets are then reported and compared to the ones obtained by LOUVAIN. Modularity (Q) and NMI values are presented and considered for the comparisons. In the first column of the table, the features of each instance tackled are also shown.

Table 3. Comparison between HYBRID-IA and LOUVAIN on Synthetic Networks with 1000 and 5000 vertices, with respect to modularity (Q) and NMI evaluation metrics.

(V , k, μ_t)	HYBRID-IA		LOUVAIN		DIFFERENCE	
	Q	NMI	Q	NMI	Q	NMI
(1000, 20, 0.1)	0.8606	0.9980	0.8607	0.9931	-0.0001	+0.0049
(1000, 20, 0.2)	0.7622	0.9970	0.7622	0.9909	0.0	+0.0061
(1000, 20, 0.3)	0.6655	0.9927	0.6642	0.9790	+0.0013	+0.0137
(1000, 20, 0.4)	0.5668	0.9905	0.5656	0.9588	+0.0012	+0.0317
(1000, 20, 0.5)	0.4685	0.9857	0.4685	0.9393	0.0	+0.0464
(1000, 20, 0.6)	0.3687	0.9767	0.3658	0.9084	+0.0029	+0.0683
(1000, 20, 0.7)	0.2707	0.9127	0.2635	0.6969	+0.0072	+0.2158
(5000, 20, 0.1)	0.8923	0.9991	0.8934	0.9589	-0.0011	+0.0402
(5000, 20, 0.2)	0.7927	0.9966	0.7948	0.9399	-0.0021	+0.0567
(5000, 20, 0.3)	0.6929	0.9967	0.6959	0.9282	-0.0030	+0.0685
(5000, 20, 0.4)	0.5931	0.9945	0.5975	0.9076	-0.0044	+0.0869
(5000, 20, 0.5)	0.4936	0.9953	0.5001	0.8789	-0.0065	+0.1164
(5000, 20, 0.6)	0.3939	0.9976	0.4027	0.8518	-0.0088	+0.1458
(5000, 20, 0.7)	0.2929	0.9942	0.3033	0.8064	-0.0104	+0.1878
(5000, 20, 0.7)	0.2929	0.9942	0.3033	0.8064	-0.0104	+0.1878

Analyzing the comparison, it is possible to see how HYBRID-IA outperforms LOUVAIN in all networks with 1000 vertices with respect to the Q modularity metric, whilst the opposite happens for those instances with 5000 vertices, where instead LOUVAIN outperforms HYBRID-IA. This gap is due to the combination between the random search and local search that, together with the diversity produced by the immune operators, requires a longer convergence time than the LOUVAIN one. It is important to point out that very likely, with a larger number of generations, HYBRID-IA would reach comparable results to LOUVAIN in terms of Q modularity also on these larger networks.

Different instead is the assessment of the comparison if it is analyzed with respect the NMI metric [34]: HYBRID-IA outperforms LOUVAIN in all networks, and this proves a better ability of the hybrid immune algorithm proposed in detecting communities closer to the true ones than the greedy optimization algorithm. Note that, although modularity assesses the cohesion of the communities detected, maximizing Q might not correspond to detect the true communities.

4.2. On the Computational Complexity of OPT-IA and HYBRID-IA

Both our algorithms are population based algorithms; therefore, any analysis of their computational complexity must deal not only with the size of the input problems and its implementation, which in turn implies the analysis of the computational cost of computing the fitness of any individual, but also with the choice of the key parameters, such as the number of elements in the population d , the total number of iterations $MaxGen$, etc. We will discuss these issues properly in what follows.

When we look at the code for OPT-IA, we have the following:

- Any element of the population, i.e., a tentative solution, is an array of length N , where N is the number of vertices of the input graph.
- The operator InitializePopulation(d) randomly creates a population of d tentative solutions. Thus, total cost is $\mathcal{O}(dN)$. However, we stress here the fact that d , which is set experimentally, is actually constant, i.e., it does not depend on the size of the input. In our case, we fixed to the value 100 for all of the experiments. This allows us to say that the cost of the procedure is actually $\mathcal{O}(N)$.

- The operator `ComputeFitness()` computes the fitness for all the $d = 100$ elements of the population. A bound on the cost of the procedure can easily be computed using Equation (1), and it clearly is $\mathcal{O}(N^2)$.
- The operator `Cloning()` creates dup copies of each of the elements of the population. As you can see from our settings, dup is a parameter which does not depend on the size of the input graph, but just on the nature of the algorithm. This allows us to say that the cost of `Cloning()` is $\mathcal{O}(N)$.
- The operator `Hypermutation()` mutates each element of the population with constant probability m . As we described earlier, we have three different implementations. In all three cases, we have either the random selection of a vertex or the random selection of a cluster. In the two most time-consuming implementations, namely *destroy* and *fuse*, we might have the reallocation of several vertices. Thus, all in all, `Hypermutation()` has an upper bound $\mathcal{O}(N)$.
- The operator `Precompetition()` randomly selects two vertices, checks their community numbers, and, if they are the same, it deletes with probability $1/2$ the one with lower fitness. Since the fitness value was already computed, the overall cost of the operator is clearly $\mathcal{O}(1)$.
- The operator `StochasticAging()` goes through all the elements of the population and removes an element with the user defined probability P_{die} . Thus, its overall cost is $\mathcal{O}(N)$.
- Finally, the operator `Selection()` chooses, without repetition of fitness values, the best d B cells among $P_a^{(pre)}$ and $P^{(hyp)}$. As we underlined before, the number of elements in these two populations is constant with respect to the size of the input, so we can simply say that the cost of the operator is $\mathcal{O}(1)$.

In summary, the cost of one iteration of OPT-IA has a computational upper bound $\mathcal{O}(N^2 + N) = \mathcal{O}(N^2)$. Finally, let us take into consideration the number of iterations. We mentioned that for all the experiments concerning OPT-IA the number of generations was fixed to 1000. For the bigger graphs, it is a number smaller than the number of edges and about five times the number of vertices. It is clear, though, that, contrary to the other parameters, the number of generations cannot be considered independent from the size of the input. The bigger the input, the bigger is the number of generations that we expect to need. However, how does it grow with respect to N ? The possibility of keeping it constant for large graphs, and the results of our experiments tell us that such a growth is at worst linear with respect to N . Thus, if we want to be very cautious in estimating the overall complexity of OPT-IA, we can say that we need at most $c \cdot N$ generations to obtain good results. If we check again Figure 1, we see that, after about $250 \sim 2.5|V|$ generations, OPT-IA reaches the optimal solution. In general, we estimate that the number of generations is at most $5 \cdot |N|$. If we add such a bound on the overall computational analysis of OPT-IA, we can certainly claim that the upper bound for its running time is $\mathcal{O}(N^3)$.

If we study the computational cost of HYBRID-IA, we find very few differences with what we saw for OPT-IA. Namely,

- The operator `Hypermutation()` in the worst case moves all the vertices from one community to another, but it is still obviously $\mathcal{O}(N)$.
- The operator `Aging()` goes through all the elements of the populations which we consider a constant number, again independent from the size of the input graph, so, all in all, it is $\mathcal{O}(1)$.
- The operator `LocalSearch()` acts on every vertex of the given graph and explores its vicinity. Therefore, we can estimate that its work has an upper bound $\mathcal{O}(N^2)$.

In summary, the cost of one iteration of HYBRID-IA has a computational upper bound $\mathcal{O}(N^2)$, just like OPT-IA, though clearly higher internal constant factors. Such a constant factor is balanced by the number of generations. We fixed it to 100 for all the experiments, including the very large synthetic graphs, where we noted we should have probably had a larger number of generations. Once again, the number of generations cannot be considered independent from the size of the input. However, certainly for HYBRID-IA, it is asymptotically not larger than what we estimated for OPT-IA. Moreover,

even assuming that its growth is linear with respect to $|V|$, the constant factor is definitively smaller. In any case, we can conclude that, also for HYBRID-IA, we have a computational upper bound $\mathcal{O}(N^3)$.

5. Conclusions

Two novel immunological algorithms have been developed for the community detection, one of the most challenging problems in network science, with an important impact on many research areas. The two algorithms, respectively OPT-IA and HYBRID-IA, are inspired by the clonal selection principle, and take advantage of the three main immune operators of cloning, hypermutation, and aging. The main difference between the two algorithms is the designed search strategy: OPT-IA performs a random and blind search in the search space, and it is coupled with pure stochastic operators, whilst HYBRID-IA is based on a refinement of the current best solutions through a deterministic local search. The efficiency and efficacy of both algorithms have been tested on several real social networks, different both in complexity and size. From the experimental analysis, it emerges that OPT-IA, thanks to its structure, carries out a careful exploration of the solutions space, but it needs a larger number of iterations, whilst HYBRID-IA quickly discovers good solutions, and exploits them during the evolutionary cycle. OPT-IA and HYBRID-IA have also been compared with seven efficient metaheuristics (included one Hyper-Heuristic), and one greedy optimization method. The obtained outcomes prove the reliability of both algorithms, showing competitiveness, and efficiency with respect to all other algorithms to which they are compared. In particular, they prove that, under the same conditions, both OPT-IA and HYBRID-IA reach better solutions (i.e., higher modularity) than the other algorithms, including LOUVAIN. Moreover, these results, along with previous successful applications in several optimization tasks on networks, prove once again that our proposed immunological approach and algorithm is one of the best metaheuristics methods in literature.

Several points need to be carefully analyzed as future work, such as an appropriate parameters tuning, an improvement of effectiveness of the hypermutation operator, and local search method in order to build more efficiently the clusters, better guide the move of the nodes between the clusters, and speed up the convergence. Finally, an obvious future research direction is to tackle larger networks, especially biological ones, which are harder and of high relevance.

Author Contributions: The authors contributed equally to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the anonymous reviewers for their useful comments and suggestions that have allowed to significantly improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. doi:10.1073/pnas.122653799. [[CrossRef](#)] [[PubMed](#)]
2. Gulbahce, N.; Lehmann, S. The art of community detection. *BioEssays* **2008**, *30*, 934–938. doi:10.1002/bies.20820. [[CrossRef](#)] [[PubMed](#)]
3. Didimo, W.; Montecchiani, F. Fast layout computation of clustered networks: Algorithmic advances and experimental analysis. *Inf. Sci.* **2014**, *260*, 185–199. doi:10.1016/j.ins.2013.09.048. [[CrossRef](#)]
4. Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. doi:10.1103/PhysRevE.69.026113. [[CrossRef](#)] [[PubMed](#)]
5. Brandes, U.; Delling, D.; Gaertler, M.; Görke, R.; Hoefer, M.; Nikoloski, Z.; Wagner, D. On Modularity Clustering. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 172–188. doi:10.1109/TKDE.2007.190689. [[CrossRef](#)]
6. Newman, M.E.J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133. doi:10.1103/PhysRevE.69.066133. [[CrossRef](#)] [[PubMed](#)]
7. Newman, M.E.J. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **2006**, *74*, 036104. doi:10.1103/PhysRevE.74.036104. [[CrossRef](#)] [[PubMed](#)]

8. Talbi, E.G. *Metaheuristics: From Design to Implementation*; Wiley: Hoboken, NJ, USA, 2009.
9. Fortunato, S.; Barthélemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 36–41. doi:10.1073/pnas.0605965104. [[CrossRef](#)]
10. Fouladvand, S.; Osareh, A.; Shadgar, B.; Pavone, M.; Sharafi, S. DENSA: An effective negative selection algorithm with flexible boundaries for self-space and dynamic number of detectors. *Eng. Appl. Artif. Intell.* **2017**, *62*, 359–372. doi:10.1016/j.engappai.2016.08.014. [[CrossRef](#)]
11. Pavone, M.; Narzisi, G.; Nicosia, G. Clonal selection: an immunological algorithm for global optimization over continuous spaces. *J. Glob. Optim.* **2012**, *53*, 769–808. doi:10.1007/s10898-011-9736-8. [[CrossRef](#)]
12. Cutello, V.; Oliva, M.; Pavone, M.; Scollo, R.A. An Immune Metaheuristics for Large Instances of the Weighted Feedback Vertex Set Problem. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 1928–1936. doi:10.1109/SSCI44817.2019.9002988. [[CrossRef](#)]
13. Di Stefano, A.; Vitale, A.; Cutello, V.; Pavone, M. How long should offspring lifespan be in order to obtain a proper exploration? In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–8. doi:10.1109/SSCI.2016.7850270. [[CrossRef](#)]
14. Spampinato, A.G.; Scollo, R.A.; Cavallaro, S.; Pavone, M.; Cutello, V. An Immunological Algorithm for Graph Modularity Optimization. In *Advances in Computational Intelligence Systems*; Ju, Z., Yang, L., Yang, C., Gegov, A., Zhou, D., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 1043, pp. 235–247. doi:10.1007/978-3-030-29933-0_20. [[CrossRef](#)]
15. Kernighan, B.W.; Lin, S. An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.* **1970**, *49*, 291–307. doi:10.1002/j.1538-7305.1970.tb01770.x. [[CrossRef](#)]
16. Sundaresan, S.R.; Fischhoff, I.R.; Dushoff, J.; Rubenstein, D.I. Network metrics reveal differences in social organization between two fission–fusion species, Grevy’s zebra and onager. *Oecologia* **2007**, *151*, 140–149. doi:10.1007/s00442-006-0553-6. [[CrossRef](#)] [[PubMed](#)]
17. Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropol. Res.* **1977**, *33*, 452–473. doi:10.1086/jar.33.4.3629752. [[CrossRef](#)]
18. Lusseau, D.; Schneider, K.; Boisseau, O.J.; Haase, P.; Slooten, E.; Dawson, S.M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **2003**, *54*, 396–405. doi:10.1007/s00265-003-0651-y. [[CrossRef](#)]
19. Krebs, V. A Network of Books about Recent US Politics Sold by the Online Bookseller Amazon.com. 2008. Available online: <http://www.orgnet.com> (accessed on 8 April 2020).
20. Gleiser, P.M.; Danon, L. Community structure in Jazz. *Adv. Complex Syst.* **2003**, *6*, 565–573. doi:10.1142/S0219525903001067. [[CrossRef](#)]
21. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *10*, 10008–10019. doi:10.1088/1742-5468/2008/10/P10008. [[CrossRef](#)]
22. Civicioglu, P. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Comput. Geosci.* **2012**, *46*, 229–247. doi:10.1016/j.cageo.2011.12.011. [[CrossRef](#)]
23. Storn, R.; Price, K. *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*; Technical Report TR-95-012; International Computer Science Institute: Berkeley, CA, USA, 1995.
24. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. doi:10.1023/A:1008202821328. [[CrossRef](#)]
25. Glover, F. Heuristics for integer programming using surrogate constraints. *Decis. Sci.* **1977**, *8*, 156–166. doi:10.1111/j.1540-5915.1977.tb01074.x. [[CrossRef](#)]
26. Martí, R.; Laguna, M.; Glover, F. Principles of scatter search. *Eur. J. Oper. Res.* **2006**, *169*, 359–372. doi:10.1016/j.ejor.2004.08.004. [[CrossRef](#)]
27. Erol, O.K.; Eksin, I. A new optimization method: Big Bang–Big Crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. doi:10.1016/j.advengsoft.2005.04.005. [[CrossRef](#)]
28. Yang, X.S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74. doi:10.1007/978-3-642-12538-6_6. [[CrossRef](#)]
29. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. doi:10.1016/j.ins.2009.03.004. [[CrossRef](#)]

30. Naeni, L.M.; Berretta, R.; Moscato, P. MA-Net: A Reliable Memetic Algorithm for Community Detection by Modularity Optimization. In Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Singapore, 10–12 November 2014; Handa, H., Ishibuchi, H., Ong, Y.S., Tan, K.C., Eds.; Springer International Publishing: Cham, Switzerland, 2015; Volume 1, pp. 311–323. doi:10.1007/978-3-319-13359-1_25. [[CrossRef](#)]
31. Atay, Y.; Koc, I.; Babaoglu, I.; Kodaz, H. Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms. *Appl. Soft Comput.* **2017**, *50*, 194–211. doi:10.1016/j.asoc.2016.11.025. [[CrossRef](#)]
32. Lancichinetti, A.; Fortunato, S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* **2009**, *80*, 016118. doi:10.1103/PhysRevE.80.016118. [[CrossRef](#)] [[PubMed](#)]
33. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark Graphs for Testing Community Detection Algorithms. *Phys. Rev. E* **2008**, *78*, 046110. doi:10.1103/PhysRevE.78.046110. [[CrossRef](#)] [[PubMed](#)]
34. Danon, L.; Díaz-Guilera, A.; Duch, J.; Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, *2005*, P09008–P09008. doi:10.1088/1742-5468/2005/09/p09008. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).