An enhanced evolutionary approach for solving the community detection problem

Cheikh Salmi a, Bouchema Sara b and Zaoui Sara b

aLaboratoire de modélisation d’optimisation et de système electroniques (LIMOSE), University of M’Hammed Bougara, Boumerdès, Algeria; bComputer Science Department, University of M’Hammed Bougara, Boumerdès, Algeria

ABSTRACT
Community detection concepts can be encountered in many disciplines such as sociology, biology, and computer science, etc. Nowadays, a huge amount of data is produced by digital social networks and needs to be processed. In fact, the analysis of this data makes it possible to extract new knowledge about groups of individuals, their communication modes, and orientations. This knowledge can be exploited in marketing, security, Web usage, and many other decisional purposes. Community detection problem (CDP) is NP-hard and many algorithms have been designed to solve it but not to a satisfactory level. In this paper, we propose a hybrid heuristic approach based on a combination of genetic algorithms and tabu search that does not need any prior knowledge about the number or the size of each community to tackle the CDP. The method is efficient because it uses an enhanced encoding, which excludes redundant chromosomes while performing genetic operations. This approach is evaluated on a wide range of real-world networks. The result of experiments shows that the proposed algorithm outperforms many other algorithms according to the modularity (Q) measure.

ARTICLE HISTORY
Received 2 September 2021
Accepted 4 September 2021

KEYWORDS
Community detection; genetic algorithm; tabu search; social networks

1. Introduction
The concept of network is omnipresent in many disciplines (sociology, chemistry, biology, etc.), particularly in several research fields in computer science. Networks are modelled via graphs, which make it easier to study, and understand their structure using graph theory. A graph is composed of nodes and edges with the possibility of orientation. In most real-world problems, arcs and edges are labelled by weights, which represent how these nodes interact in a particular context. For instance, in a collaborative network, two individuals are linked together if they cooperate to accomplish the same task.

A social network (e.g. Facebook, Twitter, Instagram, etc.) is a set of social actors (nodes), such as individuals or organizations, linked (edges) together by connections representing social interactions.
A network \( N \) can be modelled as a graph \((G)\) resulting from a set of objects, called nodes or vertices, and a set of links, called edges, that connect two nodes together. Detecting \( (K) \) communities in a network, where the number \( K \) is usually unknown, consists of finding \( K \) shards of strongly connected nodes inside the same partition and weakly connected to those of other partitions. In real-life social networks, it is often the case to find groups of nodes, which are strongly connected to each other but weakly connected to the other nodes of the network.

Therefore, community detection is an important question since it can be encountered in several fields of application and real-world situations. For example, in social networks it can reveal communities representing individuals with common interests. Accordingly, it could be possible to predict the behaviour of individuals by analyzing the behaviour of other ones belonging to the same community. Community detection in social networks is based on algorithms and methods from two relatively independent research fields’ namely automatic classification and graph theory.

Hence, these methods fall into three categories (1) hierarchical classification methods which make it possible to choose a community structure among several hierarchical levels representing different possible structures (2) graph theory algorithms that use notions of density or path search to extract community structures (3) optimization methods which identify communities by maximizing a given quality measure as an objective function. The approach proposed in the present work is based on optimization methods. It attempts to find optimal/near-optimal communities of the entire graph representing the social network by maximizing the modularity metric \( Q \) as an objective function.

The paper presents a collective approach based on the genetic algorithm (GA) and tabu search (TS) method referred to as HGT for community detection in social networks. The TS algorithm is used, as a local search technique, by this way, the exploration ability of the HGT will be improved since the HGT algorithm inherited the properties of the GA and the TS algorithm. These properties consist of the large space exploration of GA and the neighbourhood exploration and prohibition of TS.

Based on the definition of the problem of grouping given in Falkenauer (1998), one can think of CPD as a non-constrained grouping problem. This means that any grouping of objects can constitute a feasible solution, which considerably increases the search space. Therefore, the coding of the solutions must at least be unique and do not allow redundancy.

This study extends the work in Salmi et al. (2020) with an enhanced chromosome encoding to prune and optimize the search space and more experiments using diverse datasets.

The paper is organized as follows: In Section 2, an overview on state of the art solution methods will be given. Section 3 contains a description of the community detection problem (CDP). In Section 4, the HGT will be described in detail. In Section 5, the design of the numerical experiments will be outlined and the numerical results will be shown. A summary of the main findings and an outlook on further research opportunities will complete the paper.

2. Related work

As we explained in the introductory section, networks become more and more complicated and communities are often present in all types of networks. The detection of
these communities can be very beneficial to reduce the complexity of networks in order to better understand them. In this section, we present a literature review on CDP.

Many algorithms have been proposed to solve the problem of community detection. In what follows, we cite some notable works, which are known as the best algorithms and considered as good benchmarks for a performance comparison in terms of the accuracy in community detection algorithms development. The Louvain algorithm (Blondel et al., 2008) is a hierarchical clustering algorithm, that recursively merges communities into a single node and executes the modularity clustering on the condensed graphs. The algorithm attempts to make the intra-community density exceeds the inter-community density. Initially, every vertex belongs to a different partition. As iteration progress, vertices are grouped, in partitions of optimal modularity. Having reached a first optimum situation, the process continues at the higher level: each partition is treated as a vertex and so on. The operation continues until there is no further improvement in modularity.

This algorithm is currently the best algorithm in terms of complexity to calculate communities on very large graphs (it is capable of processing graphs with more than a billion vertices and edges in less than 3 h).

Greedy is one of the agglomerative approaches. Girvan and Newman have introduced it. At each step, the algorithm tries to merge communities in order to increase the value of modularity $Q$ (Souravlas et al., 2021). Initially, each vertex is considered as a community. Then, a merging of pairs of neighbouring communities is performed to maximize the modularity $Q$. However, the algorithm does not merge pairs of communities, which are not connected. This process is repeated until the modularity $Q$ cannot be improved. This algorithm has been widely disseminated because it is able to process networks of hundreds of millions of vertices in minutes and also it is able to find small communities, even in very large graphs.

The Edge betweenness centrality algorithm (Freeman, 1979) starts by calculating the centrality for each edge. Then, the edge that has a strong betweenness is removed. This process is repeated until all the edges have been removed which allows to put highlight the different communities that exist. The modularity $Q$ is used to choose the best level of partition from a dendrogram.

As for each partition obtained, the value of modularity $Q$ is recalculated. The drawbacks of this algorithm appear in calculating the measure of centrality. The calculation process is too slow because a course by all possible paths between all the pairs of vertices must be made for each link.

MENSGA (Li et al., 2013) is a GA encoding the individuals by adjacency matrix $M$ where rows represent vertices, and columns represent communities of the graph $G$. In the first step, it uses an algorithm for population initialization based on nodes similarity (PINS). Genetic operation are then performed to optimize the modularity function.

Authors in Gharaghooshi et al. (2020) introduce a new approach for community detection based on the formalization of the community concept and define a new objective function, called SIWO (Strong Inside, Weak Outside) which only favours the addition of strong links to the community while avoiding weak links that connect the community to others.

In Zhenqi et al. (2018) a novel community detection method based on cliques is developed. The proposed method relies on the mathematical properties of cliques. The method attempts to remedy the shortcomings of the k-clique percolation method used in (Imre et al., 2005).
3. Community detection problem (CDP)

The objective of community detection is to partition a graph representing a social network into disjoint or overlapping groups of vertices so that the nodes within the same group are densely connected. In the particular case of disjoint communities, this also means that the resulting groups are weakly connected. To do so, some algorithms (including ours) use the modularity $Q$ (Newman, 2006) optimization over the possible graph partitions. The modularity measure represents the difference between the adjacency value between two nodes of the same community and the probability that these nodes are connected. More formally, let $G = (V, E)$ be the original graph denoting the social network, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes is and $E = \{E_1, E_2, \ldots, E_m\}$ is the set of links. The objective is to find a partitioning which gives the best community structure $C = \{C_1, C_2, \ldots, C_k\}$, i.e., a maximum value for the $Q$ function regarding $C$. The modularity cost function is defined by the equation:

$$Q = \frac{1}{2m} \sum_{ij} (B_{ij} - \frac{d_i d_j}{2m}) \delta(k_i, k_j)$$ (1)

where $i = 1, \ldots, n$ and $j = 1, \ldots, n$, $m$ is the number of links in the graph, $n$ is the number of nodes, $B_{ij}$ is 1 if the nodes $i$ and $j$ are linked and 0 otherwise; the variable $d_i$ is the degree of node $i$, $d_j$ is the degree of node $j$ and $\delta$ is the Kronecker delta function which evaluates to one if nodes $i$ and $j$ belong to the same community and zero otherwise. The CDP has been shown to be NP-hard (Karp, 1972). For more details on modularity, see Newman (2006).

4. Hybrid genetic-tabu for community detection

As mentioned previously, we investigate the use of a hybrid GA and TS for the CDP in social networks. Therefore, a novel hybrid genetic approach called (HGT) is proposed and compared with the state-of-the-art approaches. The basic idea of HGT is to refine the solution found by GA (Sivanandam & Deepa, 2007) using TS (GendreauJean & Potvin, 2007) which is an evolutionary heuristic that updates a single solution. Hence, the rationale behind the HGT approach is to start from an already good solution given by GA and successively move it to one of its current neighbours using TS always with the aim of improving the modularity $Q$. Figure 1 shows a global view of the proposed approach. Its key modules are described in detail in the following sections.

4.1. GA for community detection

GAs are optimization algorithms based on techniques derived from genetics and natural evolution. GAs are population-based optimization. A population is a set of elements called chromosomes. A GA is used to determine the extrema(s) of a function defined on a search space relating to a given problem. It is based on the following components: (1) principle for coding population chromosomes (2) mechanism for generating the initial population (3) function to optimize (4) operators to diversify the population over generations (cross-over, mutation, selection, etc.) (5) design parameters (population size, total number of generations, probabilities of application of crossover and mutation operators).
4.1.1. Chromosome encoding

Chromosome encoding is a very important step in GA-based approaches. It allows describing how to associate a chromosome to a solution. In HGT, each value of a chromosome represents an association of a node to its corresponding community. For a graph \( G = (V, E) \) with \( n \) nodes, a chromosome \( i \) is represented by an integer array \( X_i = \{X_{i1}, X_{i2}, \ldots X_{in}\} \), where each \( X_{ik} \) is an integer value that represents the index of the community to which the node \( k \) belongs. These index values range from 1 to \( n \).

In contrast to the approaches proposed in Pizzuti (2008) and Handl and Knowles (2007), this encoding is straightforward (the node-community relationship is present in the encoding itself) and does not involve any decoding procedure to associate a node with its corresponding partition. In addition, this coding scheme does not allow having the same node in two different communities, which is very useful for the detection of non-overlapping communities. However, the mapping between the sets of chromosomes and communities is not a bijection. This means that several chromosomes can represent the same community.

Figure 1. Hybrid genetic and tabu search (HGT) approach global view.
Example 4.1: For a network containing nine nodes, \{5, 1, 2, 3, 1, 2, 4, 6\} is a possible encoding of a solution containing six communities. The resulting communities are \{[1]\; [2, 5, 6]; [3, 7]; [4]; [8]; [9]\}.

Enhanced Chromosome Encoding: as pointed out in Falkenauer (1998), the major drawback of GA-based approaches in solving grouping problems is that the representation can produce several chromosomes to encode the same solution, i.e. for the same solution, it is possible to have several encodings. This redundancy, if it is large, can pose problems for the convergence and the quality of the obtained solutions. To reduce coding redundancy and improve the performance of our approach, we use a restricted growth (RG) string proposed in Ruskey (1993).

RG String Principle: as in the basic encoding, a decomposition of a graph \( G \) into a set of \( k \) communities (\( k \geq 0 \)) is a set \( C(k) = \{C_1, C_2, \ldots, C_k\} \) where each \( C_i \) (\( i = 1 \ldots k \)) is a community and \( \forall i, j \in \{1 \ldots k\}, C_i \cap C_j = \emptyset \). The restricted growth functions (RG functions) of length \( n \), denoted \( R(n) \) are those strings \( a_1 \ldots a_n \), of non-negative integers satisfying \( a_1 = 0 \) and \( a_i \leq 1 + \max\{a_1, \ldots, a_{i-1}\} \). A chromosome is an RG string.

Example 4.2: \{0, 1, 0, 2, 3, 4, 1, 5, 2\} is an RG string but \{6, 8, 6, 13, 17, 50, 8, 10, 13\} is not. These two chromosomes denote the same solution in basic encoding scheme, which is \{[1, 3]; [2, 7]; [4, 9]; [5]; [6]; [8]\}.

4.1.2. Solution repair
To avoid chromosomes redundancy, it is important that each chromosome is an RG string. However, random initialization does not ensure each chromosome to be an RG string. In addition, genetic operations (crossover and mutation) may generate new chromosomes that are not RG strings. Example 4.3 illustrates such case.

Example 4.3: Let \( r_1 = \{1, 1, 2, 3, 2, 1, 3\} \) and \( r_2 = \{1, 2, 1, 3, 4, 2, 3, 1\} \) be two RG strings; and assume one-point crossover occurs after position 4. The resulting two new chromosomes are \( \{1, 1, 1, 2, 4, 2, 3, 1\} \) and \( \{1, 2, 1, 3, 3, 2, 1, 3\} \); but the first chromosome is not RG string. Similarly, one-point mutation on \( r_1 \) at position 2 may produce the chromosome \( \{1, 1, 1, 3, 3, 2, 1, 3\} \), which does respect RG string rules.

To guarantee that all chromosomes are RG string we use a repairing algorithm as illustrated by the algorithm presented in Figure 2.

4.2. TS for community detection
TS is a heuristic local search method used to solve complex and NP-hard problems. Its main idea is to continue the exploration of the search space even if a local optimum is encountered, allowing movements in the search space that do not improve the solution and using the memory principle to avoid going backwards (cyclical movements). The memory is represented by a tabu list used to improve the solution diversification. It contains the movements, which are temporarily prohibited. However, the role of the tabu list could evolve during the resolution towards intensification. Hence, it is possible to violate the tabu list restriction if a prohibited movement could improve the best solution
recorded so far. To implement these intensification and diversification strategies, three lists are maintained:

- **LTI**: a tabu intensification list, which is a medium-term memory in order to avoid cycles in a local space.
- **LTD**: a tabu list of diversification which is a long-term memory in order to store the best solutions provided for each iteration for a specific duration.
- **LC**: a candidate’s list is used to store neighbour solutions if they are not already in the LTI and LTD lists.

### 4.3. Neighbourhood structure

An important issue of any local search algorithm for combinatorial optimization problems is the definition of an effective neighbourhood around an initial solution. In this work, immediate neighbours of a given solution are determined using node permutation. Given a graph $G(V, E)$, let $C$ denote the set of feasible solutions represented by their partitioning schemes. A neighbourhood structure is a function $N: S \rightarrow 2^s$, which associates a set of solutions $N(s)$ with each solution $s \in S$ obtainable by a predefined partial modification of $s$ which consists to change the community of one node, usually called move. Three types of moves are considered in this work: (1) permutation of two arbitrary distinct elements, (2) permutation of two successive elements and (3) single element shifting. The search moves from one solution to a new one by choosing the best not forbidden element in the neighbourhood. To optimize the search process, a solution $s'$ is considered to be forbidden if the current solution $s$ can be transformed into $s'$ by applying one of the moves in the tabu list, i.e., only forbidden moves are stored in the tabu list.
4.4. HGT: the hybridization approach

The proposed hybridization consists of alternating the stages of the GA and TS global and local search processes to diversify and intensify the solutions. In fact, the HGT begins by creating an initial solution \( S \) and then generates a set of neighbours, which forms the initial population \( P \). Then a combination of global and local search is performed to evaluate the population \( P \) by calculating the fitness function \( Q \) for each solution and the best solutions \( s \) are determined. These solutions are selected and inserted in the LTI list if they do not exist in the tabu list LTD. Then, the local search is called again to generate a set of neighbours for each element of the LTI list. Solutions that do not already exist in this list are stored in the candidate’s list LC. Meanwhile, the two best solutions are select, one from LTI and the other from LC (representing the parents) to perform the genetic operation (crossover, mutation). This process results in two new solutions that represent children. Finally, if the stopping criterion is not yet met, a new population is produced and the process is relaunched again.

The detailed steps of the proposed HGT framework (see Figure 3) are discussed in the following steps:

- **Generation of an initial solution:** HGT starts from an initial solution \( s_0 \). This solution can be generated by different methods, either from metaheuristics, heuristics, and exact methods or by using a random solution.
- **Generation of a population:** local research aims to enrich the set of solutions by exploring the neighbours solutions of a current solution \( s \). Hence, from the solution \( s \), we generate a set of its neighbours \( N(s) \) using the moves described above (see section. Neighbourhood Structure). At each time a community is randomly changed (structurally we change the gene value). The resulting set forms a population \( P \) (the first generation) that may contains both optimal and not-optimal communities.

![Figure 3. Illustration of hybrid genetic and tabu search (HGT) to solve CDP.](image-url)
• Evaluation and Selection of the best solutions: population $P$ individuals are evaluated by calculating the quality measure $Q$ as the objective function. For each individual of $P$, the measure $Q$ indicates its quality. Therefore, the individual with the highest value of $Q$ is considered to be the best in $P$. Then, the best solutions are selected by elitism and placed in the LTI list if they do not exist in the LTD list. Recall here, that LTI (tabu list of intensification) is used to avoid being trapped in a local minima and LTD (tabu list of diversification) is the long-term memory used to store the best solutions provided at each iteration (for a specific period of time).

• Generation of neighbours for each element of LTI: after putting the solutions selected in the tabu list LTI, the next goal is to intensify the solution space. First, for each solution $s_0$ found in this list, a set of neighbours $N(s_0)$ is generated. Then, for each neighbour, if it does not exist in the LTI and LTD lists, then it will be stored in the LC list (candidate list), else, another neighbour is selected. This step allows having new solutions that do not already exist.

• Selection of parent solutions: in this step, we choose the two best solutions (individuals) as parents for next generations respectively from the LTI and LC tabu list. Recall here, that both parents should not be in the LTD tabu list.

• Evaluation and comparison of parents with the best solution $S^*$: if the two parents have different modularity, the best parent is taken and compared with $S^*$. If the modularity $Q$ of a parent is greater than that of $S^*$, then $S^*$ is updated and the best parent added in the taboo list LTD. Otherwise the process continues to the next step.

• Updating tabu list LTD: the tabu LTD list allows prohibiting the return to a solution already visited. Therefore, at each iteration, the best solutions are saved in LTD with a precise duration where this tabu list must be updated by releasing certain prohibited solutions. This update of LTD makes it possible to give opportunities for prohibited solutions to be used in the next iterations.

• Crossover between parents: in this step, parents are combined to produce new individuals to diversify the space of solutions. A one-point crossover is performed to produce two new solutions (child 1, child 2).

• Mutation, evaluation and comparison of children with the best solution $S^*$: after performing the crossover, the new solutions are compared with $S^*$ as it is done with their parents. The search space is enriched with new solutions, the LTD list is updated and the best solution is accepted if its modularity is better than that of the current $S^*$.

• Stopping criteria: the stop criteria is based on the number of iterations (generations). If the last iteration is not reached then the initial population is replaced by a new population, which contains the best solutions by combining the solutions of the initial population, of the LC list and the children solutions. Then, the process continues to the next iteration. Note that, the termination condition is a succession of 1000 generations without significant improvement in modularity (greater than $10^{-6}$).

4.5. Complexity analysis

The modularity fitness is a two-dimensional function. The size of a chromosome is equal to the number of nodes ($n$) in the graph, so the complexity of each solution evaluation is $O(n^2)$. 
Thus, the overall complexity of the fitness evaluation has order of $O(kgn^2)$ with $(k)$ individuals in the population and $(g)$ generations. Besides the evaluation of the modularity routine, $O(n)$ operations are needed to initialize a single chromosome. Thus, a population creation for all generations has the time complexity of $O(kgn)$. The selection of the best solutions from a population of size $(k)$ takes $O(k\log(k))$ time if a quicksort is used. We crossover and mutate two chromosomes; if $m_p$ and $c_p$ denote respectively the mutation and crossover probabilities, then the number of operation needed for mutations and crossovers are respectively $O(2m_pn)$ and $O(2c_pn)$ since we need to go along each chromosome. Previous operation (selection of the best solutions, mutation and crossover) are repeated for all generations. Therefore, the times complexity of HGT is: $O(kgn^2k\log(k) + 2gm_pn + 2gc_pn)$. Since $k$, $g$, $m_p$ and $c_p$ are constants, the overall time complexity of our approach is on the order of $O(n^2)$. There are many community detection algorithms in social networks with different time complexity. We quote Edge betweenness (Girvan & Newman, 2002) with a complexity of $O(nm^2)$ for a network with $n$ nodes and $m$ edges, Fastgreedy (Clauset et al., 2004) with $O(n\log^2 n)$ for sparse hierarchical networks, Label propagation (Raghavan et al., 2007) with computational complexity $O(m)$, Leading eigenvector (Newman, 2006) with a complexity of $O(n^2)$. Walktrap (Pons & Latapy, 2005) with $O(mn^2)$ and $O(n^2\log(n))$ for sparse networks and ELPA (Wei et al., 2016) with a complexity of $O(n + m)$. There are many community detection algorithms that have quadratic complexity but are efficient (in terms of execution time) in practice especially for medium-sized networks. The main advantages of our approach are (1) it can be used with all community quality measurement functions (2) it can be easily parallelized.

5. Implementation and tests

Our HGT algorithm is implemented in Python on a PC equipped with an Intel Core i7 870 2.93 GHz processor with 8 GB of RAM. Since our approach is a stochastic one, at each generation, the population changes and there is no absolute assurance that it will always result in the same solution, or the best possible solution. To be more realistic, we take the average value of the fitness function for 10 different runs.

5.1. GA configuration

A key to GA success is an optimal configuration. Good GA search performance is only possible if care is taken to configure the algorithm appropriately. GA theory provides little guidance for the proper selection of the configuration parameter settings. Although several papers describe optimization for GA configuration, the refined tuning of GA is a time-consuming trial-and-error one that is unavoidable if the algorithm is to be successful. The configuration used in the case studies is based on our experiments and the recommendations commonly found in the literature. The following presents the parameters used for the trail runs conducted in the experiments:

- Population size: 200
- Mutation rate: 0.01
- Selection mechanism: 2-competitor tournament
- Elite copies: 10.
It is worth noting that the selection of the population size and the mutation rate parameters are carried out beforehand during the experimentation phase by varying one parameter at a time while keeping the other constant.

5.2. Datasets

In order to evaluate and benchmark the HGT algorithm outlined in this paper, Real-World Datasets are used. The goal of this section is to formally introduce these datasets. Real-world datasets describe relationships observed in real phenomenons. In contrast with synthetic graphs, which are generated specially for benchmarking algorithms. In the scope of this work, only small and medium networks sizes networks are considered, as the aim of our algorithms is to achieve the higher possible modularity. The datasets studied are summarized in Table 1.

5.3. Normalized mutual information results

To assess solution quality, where reliable ground-truth data is available, we report the congruence between the results of our algorithms and the reference communities using normalized mutual information (NMI) (Leon et al., 2005), which is defined as:

\[
\text{NMI}(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log \left( \frac{C_{ij}N}{C_i C_j} \right)}{\sum_{i=1}^{C_A} C_i \log \left( \frac{C_i N}{N} \right) + \sum_{j=1}^{C_B} C_j \log \left( \frac{C_j N}{N} \right)}
\]

(2)

Therefore, the NMI is used to calculate levels of similarity between the ground truth partitions A and identified partitions B. \(C_A\) represent the number of actual communities, \(C_B\) the number of identified communities, C a confusion matrix, \(C_{ij}\) the number of nodes shared in common between communities \(C_i\) and \(C_j\) (\(C_{ij}\) is the number of nodes of community \(i\) of the partition A that are also in the community \(j\) of the partition B), \(C_i\) is the sum over row \(i\) of matrix C, and \(C_j\) is the sum over column \(j\) of matrix C. The NMI value range is between 0 and 1. If \(\text{NMI}(X, Y) = 1\), the two partitions are considered identical, otherwise they are considered independent. An NMI measure is used to check the quality of

| Networks          | \(|V|\) | \(|E|\) | Reference                  |
|-------------------|-------|-------|----------------------------|
| Karate Club       | 34    | 73    | Zachary (1977)             |
| Dolphins          | 62    | 159   | Lusseau et al. (2003)      |
| Political Books   | 105   | 441   | Orgnet (2020)              |
| College Football  | 115   | 613   | Girvan et al. (2002)       |
| Jazz              | 198   | 2742  | Girvan et al. (2002)       |
| E-Coli            | 418   | 519   | Shen-Orr et al. (2002)     |
| Email             | 1133  | 5451  | Guimerà et al. (2003)      |
| Power Grid        | 4941  | 6494  | Watts et al. (1998)        |
| Wiki vote         | 7066  |       | Leskovec et al. (2010)     |
| Erdos             | 1133  | 5451  | Grossman (2020)            |
| PGP               | 9377  | 24,107| Boguñá et al. (2004)       |
| CA-HepTh          | 9877  |       | Leskovec et al. (2007)2007 |
| CA-CondMat        | 23,133|       | Leskovec (2007)            |
| Email-Enron       | 36,692| 183,831| Snap (2020)               |
identified community structures compared to an actual partition for Karate, Polbooks, Football, Dolphins and Polblogs networks. Results are reported in Table 2. Here $Q$ is the modularity result, $NC$ is the number of communities and NMI the normalized mutual information. In Table 2 (line 1 and 3 where $NMI = 1$) shows that HGT has exactly discovered the well-known partitioning structures as reported in Zachary (Zachary, 1976) and Lusseau (Lusseau, 2003), and also has obtained better values namely 0.41 for Karate club and 0.51 for Dolphins. However, the best modularity values are respectively 0.42 and 0.52 as found in (Blondel et al., 2008). For networks Football and Polblogs our approach has attained acceptable results in terms of NMI compared to those reported in (Jia & Kuru, 2017) which are respectively 0.96 and 0.57. Regarding modularity, the HGT obtained values are fairly well with the best ones reported in (Jia & Kuru, 2017) namely 0.60 and 0.52.

### 5.4. Quality measure results

In this section, we compare the modularity obtained by our approach with that given by the 10 medium-sized networks most used in the literature, namely: Djazz, E-coli, Email, Power Grid, Wiki vote, Erdos, PGP, CA-HepTh, CA-CondMat and Email-Enron.

The comparisons are made against the best algorithms found in the literature, namely: Eigenvector (EG) (Newman, 2006), Louvain (LV) (Blondel et al., 2008), Label propagation (LP) (Raghavan et al., 2007), Infomap (IM) (Rosvall & Bergstrom, 2008; Rosvall et al., 2009), Multi-level (ML) (Noack & Rotta, 2009) and Fast Greedy (FG) (Clauset et al., 2004). The implementations of the Louvain, label propagation and InfoMap algorithms were retrieved from the igraph library (Csardi & Nepusz, 2006). The other algorithms implementations were retrieved from authors’ implementation websites in particular that of GitHub. Some modularity values are directly retrieved (when they are available) from various publications such as those of Zachary, Dolphins, Football and Polblogs networks. Figure 4 illustrates the results. From this figure, it is clear that our algorithm gives results almost similar to those given by the best algorithms found in the literature. Also, note that, since the results are obtained by performing several computational tests on networks with different sizes from the literature, lead us to say that the method is robust and efficiently reaches good results. To make these results more readable, we calculate the percentage of relative error (Zwillinger, 2002) to the best result (the optimum known modularity value) using the formula:

$$PR_{err} = \frac{1}{n} \sum_{t=1}^{n} \frac{Q_{HGT} - BR}{BR}$$

where $n = 1$ and $Q_{HGT}$ stands for the modularity of our approach (see Table 3). A negative

### Table 2. Performance comparisons with ground truth.

| Networks | Ground truth | HGT |
|----------|--------------|-----|
|          | NC | $Q$ | NC | NMI | $Q$ |
| Karate   | 2  | 0.37| 2  | 1.00| 0.37 found (average is 0.41) |
| Football | 12 | 0.55| 12 | 0.93| 0.59 |
| Dolphins | 2  | 0.38| 2  | 1.00| 0.38 found (average is 0.51) |
| Polblogs | 2  | 0.41| 2  | 0.46| 0.51 |
value for \( P_{err} \) explains the fact that the reference algorithm gives a better modularity than that obtained by our HGT approach, a null value means that the modularity value is the same as the benchmark algorithm. Finally, a positive value means that our result is better than that of the benchmark algorithm. The comparison with the best approach shows that, although being worse in some cases, the results obtained with HGT are in general of the same order of magnitude as those obtained with the best-selected algorithm. Remember that the results presented represent the average of 10 different runs of the algorithm. The algorithm exhibits a constancy of performance, which qualifies it for all the graphs tested. In addition, this consistency is found at each instance of execution since the average is slightly lower than the maximum. Moreover, for all graphs, the average is often close to the best known results. Our interpretation of this stable behaviour is that our algorithm treats all nodes and edges the same and does not rely on any particular type of node to determine communities. As already shown in Section 4.5, some algorithms change qualitative performance and execution time depending on the nature and properties of the network (e.g. sparse, dense, etc.). For instance, algorithms based on link centrality (e.g. edge betweenness) rely on particular links by assuming that these links are located between communities and attempt to remove them to isolate communities. Each algorithm in this category differs in the way in which it selects the particular links to be deleted, hence the sensitivity of the results in relation to this choice.

6. Conclusion

Community detection is a significant problem for many real-word applications. An important number of approaches were proposed to solve it. In this paper, we presented a hybrid algorithm for detecting communities in social networks. The approach is a combination of two well-known efficient metaheuristics, namely GA and TS.

Traditionally, the encoding scheme for CDP uses an integer string to represent a solution; a value \( j \) assigned to the \( i \)th gene is interpreted as a link between the nodes \( i \) and \( j \) which means that these nodes are in the same community. However, this encoding is not direct and requires a decoding step to identify all the components of the corresponding graph. In our approach, the structure of the clustered network is present in the encoding itself. Particularly, this approach is based on an RG string chromosome encoding to overcome the redundancy in solution populations. This encoding is used
Figure 4. Comparison of hybrid genetic and TS approach with best-known algorithm on most known social networks. (a) Djazz, (b) Ecoli, (c) Email, (d) Power Grid, (e) Wiki-Vote, (f) Erdos-Net, (g) PGP, (h) CA-HepTh, (i) CA-CondMat, (j) Email-Enron.
to improve solution quality and to enhance overall approach convergence performance.

The approach uses the modularity metric $Q$, and searches for an optimal clustering of the network by maximizing the modularity metric. All the dense communities present in the network structure are obtained at the end of the algorithm by selectively exploring the search space using GA operation and Tabu lists, without the need to know in advance the exact number of groups.

The approach is executed on real networks of different sizes. Experiments showed the capability of the HGT approach to correctly detect communities with comparable precision with the Louvain algorithm, which is currently the best method to detect communities in social networks, based on maximizing modularity. For medium-sized networks, we achieved good results where we clearly showed the capacity of our approach to correctly detect communities and outperform the best algorithms of the state of the art.

Community detection in social networks is an NP-hard problem. For large networks, the resolution time becomes more expensive. However, the results obtained remain very correct compared to the best algorithms of the literature. Also, final results show that using hybrid metaheuristics combining global and local exploration of the search space, that take into account decisions made in previous iterations for choosing the current movements, is a positive way of tackling the CDP. Recall that this approach is a generic framework, able to handle several problem variations and to tackle different objective functions.

Future perspectives will focus on (1) contribute to the resolution of this problem by proposing parallel methods to detect communities in large complex networks (2) the application of the approach on concrete CDPs (3) take into account the specificity of the data and the user profile to detect the domain or community type in concrete CDPs (4) test the approach by exploiting other modularity functions (5) reduce the number of iterations by developing new link-specific modularity measures that distinguish inter-community links from intra-community links (6) develop the approach on top of a framework that allows for the distributed processing of large data sets across clusters of computers (e.g. Hadoop and spark) to deal efficiently with dynamic networks and Big-networks with millions of nodes.

Finally, the literature review shows that there is a number of community detection approaches that have been proposed over the last two decades for a variety of specific scenarios. In principle, the question is the same and consists of breaking down a graph into several subgraphs. However, the comparison between different community detection algorithms is difficult and unclear. This can be explained by the fact that: there are many modularity functions, unseen details of the structures of medium and large networks (network connectivity, network diameter, betweenness, degree, island nodes, etc.), synthetic data networks generated in different ways, different evaluation criterion’s such as modularity, NMI, (A)RI, purity, etc.

In fact, it is not even clear how to experimentally compare different community detection algorithms. We point out that it would be interesting to do research in this direction to unify use-case settings that allow different algorithms to be compared apples to apples and under the same settings.
Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Cheikh Salmi obtained the Magister degree in Theoretical Computer Science and Information Engineering from M’hammed Bougara University, Boumerdès, Algeria, in 2006, and the PhD degree in Computer Science and Applications from the National Superior School of Mechanics and Aerotechnics, Poitiers, France, in 2017. His research interests are databases and data warehouses and computer applications using AI methodologies such as data mining, machine learning and logical reasoning.

Sara Bouchema received the B.Sc. degree in computer science from Boumerdes University in 2015 and a M.Sc. degree in 2017. Her research interests are exact and meta heuristic algorithms and their application in real problems.

Sara Zaoui received the B.Sc. degree in computer science from Boumerdes University in 2015 and a M.Sc. degree in 2017. Her research interests are social networks, especially, she focuses on community detection problems in the context of big data using AI.

ORCID

Cheikh Salmi http://orcid.org/0000-0001-7131-6158

References

Blondel, V. D.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(4), 10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

Boguñá, Marián, Pastor-Satorras, Romualdo, Diaz-Guilera, Albert, & Arenas, Alex. (2004). Models of social networks based on social distance. Physical review. E, Statistical, nonlinear, and soft matter physics, 70(5), 056122–056130. https://doi.org/10.1103/PhysRevE.70.056122

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. Physical Review E, 70(6), 066111. https://doi.org/10.1103/PhysRevE.70.066111

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal Complex Systems, 1695(5), 1–9. http://www.interjournal.org/manuscript_abstract.php?361100992

Duncan, J. Watts, & Steven, H. Strogatz. (1998). Collective dynamics of ‘small-world’ networks. Nature, 393(6684), 440–442. https://doi.org/10.1038/30918

Falkenauer, E. (1998). Genetic algorithms and grouping problems. USA: John Wiley & Sons, Inc.

Freeman, L. C. (1979). Centrality in social networks conceptual clarification. Social Networks, 1(3), 15–239. https://doi.org/10.1016/0378-8733(78)90021-7

GendreauJean, M., & Potvin, Y. (2007). Tabu search. In M. GendreauJean & Y. Potvin (Eds.), Search methodologies (pp. 165–186). Springer.

Gharaghooshi, S., Zainane, Z. O., Lagerson, C., Mohammad, m. Z., & Liu, C. (2020). Addressing the resolution limit and the field of view limit in community mining. In International symposium on intelligent data analysis (pp. 210–222). Springer.

Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. In L. A. Shepp (Ed.), Proceedings of the National Academy of Sciences of the United States of America (pp. 7821–7826). Rutgers, State University of New Jersey–New Brunswick.

Guimerà, R, Lenon, Danon, Alber, Diaz-Guilera, Frabces, Girlat, & Alex, Arenas. (2003). Self-similar community structure in a network of human interactions. Physical review. E, Statistical, nonlinear, and soft matter physics, 68(6), 065–0103. https://doi.org/10.1103/PhysRevE.68.065103
Handl, J., & Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1), 56–76. https://doi.org/10.1109/TEVC.2006.877146

Imre, D., Gergely, P., & Tamás, V. (2005). Clique percolation in random networks. *Physical Review Letters*, 94(16), 160202. https://doi.org/10.1103/PhysRevLett.94.160202

Jia, H. C., & Kuru, R. (2017). A semi-synchronous label propagation algorithm with constraints for community detection in complex networks. *Scientific Reports*, 7. https://doi.org/10.1038/srep45836

Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9.

Leon, D., Jordi, D., Albert, D.-G., & Alex, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics*, 2005(9), 9008–9019. https://doi.org/10.1088/1742-5468/2005/09/p09008

Leskovec, Jure. (2020). SNAP: https://snap.stanford.edu/data/email-Enron.html

Leskovec, Jure, & Daniel, Huttenlocher. (2010). *Signed networks in social media*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta Georgia USA, April 10 - 15, 2010.

Leskovec, Jure, Kleinberg, Jon , & Christos, Faloutsos. (2007). Graph Evolution: Densification and Shrinking Diameters. *Physics and Society*, 1(1), 1–42. https://doi.org/10.1145/1217299.1217301

Li, Y., Liu, G., & Lao, S.-y. (2013). A genetic algorithm for community detection in complex networks. *Journal of Central South University*, 20(5), 1269–1276. https://doi.org/10.1007/s11771-013-1611-y

Lusseau, D. (2003). The emergent properties of a dolphin social network. *Proceedings of the Royal Society B*, 207(SUPPL_2), 186–188. https://doi.org/10.1098/rsbl.2003.0057

Newman, M. E. (2006). Modularity and community structure in networks. In *Proceedings of the National Academy of Sciences of the United States of America* (pp. 8577–8582). Brian Skyrms, University of California.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3). 10.1103/PhysRevE.74.036104.

Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3). http://dx.doi.org/10.1103/PhysRevE.74.036104

Noack, A., & Rotta, R. (2009). Multi-level algorithms for modularity clustering. In *International symposium on experimental algorithms* (pp. 257–268). Springer.

Orgnet. (2020). http://www.orgnet.com/divided3.html.

Pizzuti, C. (2008). GA-Net: A genetic algorithm for community detection in social networks. In *International conference on parallel problem solving from nature* (pp. 1081–1090). Springer.

Pons, P., & Latapy, M. (2005). Computing communities in large networks using random walks. In *International symposium on computer and information sciences* (pp. 284–293). Springer.

Raghavan, U. N., Reka, A., & Soundar, K. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 036106. https://doi.org/10.1103/PhysRevE.76.036106

Rosvall, M., Axelsson, D., & Bergstrom, C. T. (2009). The map equation. *The European Physical Journal Special Topics*, 178(1), 13–23. https://doi.org/10.1140/epjst/e2010-01179-1

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4), 1118–1123. https://doi.org/10.1073/pnas.0706851105

Ruskey, F. (1993). Simple combinatorial gray codes constructed by reversing sublists. In *International symposium on algorithms and computation* (pp. 201–208). Springer.

Salmi, C., Bouchema, S., & Zaaoui, S. (2020). A hybrid heuristic community detection approach. In 2020 *International conference on innovations in intelligent systems and applications (INISTA)* (pp. 1–7). Springer.

Sivanandam, S., & Deepa, S. N. (2007). *Introduction to genetic algorithms*. Springer.

Souravlas, S., Sifaleras, A., Tsintogianni, M., & Katsavounis, S. (2021). A classification of community detection methods in social networks: A survey. *International Journal of General Systems*, 50(1), 63–91. https://doi.org/10.1080/03081079.2020.1863394
Wei, L., Xingpeng, J., Matteo, P., & Xiaofan, W. (2016). Discovering communities in complex networks by edge label propagation. *Scientific Reports, 6*, 22470–22480. https://doi.org/10.1038/srep22470

Zachary, Wayne W. (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research, 33*(4), 452–473. https://doi.org/10.1086/jar.33.4.3629752

Zachary, W. W. (1976). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research, 33*(4), 452–473. 10.1086/jar.33.4.3629752.

Zhenqi, L., Johan, W., & Arye, N. (2018). Community detection in complex networks via clique conductance. *Scientific Reports, 8*(1), 5982. https://doi.org/10.1038/s41598-018-23932-z

Zwillinger, D. (2002). *CRC standard mathematical tables and formulae*. CRC Press.

The erdos number project, oakland University. (2020). https://oakland.edu/enp/thedata/.