A Hybrid Heuristic Community Detection Approach

Salmi Cheikh
LIMOSE Laboratory
Université M'hamed Bougarra
Boumerdès, Algeria
C.salmi@univ-boumerdes.dz

Bouchema Sara
Université M'hamed Bougarra
Boumerdès, Algeria
Sarah.Bouchema@gmail.com

Zaoui Sara
Université M'hamed Bougarra
Boumerdès, Algeria
Sarah.Zaoui@gmail.com

***Abstract*—Community detection is a very important concept in many disciplines such as sociology, biology and computer science, etc. Nowadays, a huge amount of data is produced by digital social networks. In fact, the analysis of this data make 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 ($\mathcal{CDP}$) 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 that does not need any prior knowledge about the number or the size of each community to tackle the $\mathcal{CDP}$. This approach is evaluated on real world networks and the result of experiments show that the proposed algorithm outperforms many other algorithms according to the modularity ($\mathcal{Q}$) measure.**

***Index Terms*—community detection, social networks, modularity, metaheuristics, hybridization, genetic algorithm, tabu search**

I. Introduction

The concept of network is omnipresent in many disciplines (sociology, chemistry, biology, etc.), in particular in several research fields in computer science. Networks are modelled via graphs which makes 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. In a graph representing a social network, 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. These groups are called communities and they are sets of connected nodes whose link density is higher than in other regions in the graph. 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. Therefore, it could be possible to predict the behaviour of individuals by analysing 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 attempt to find optimal/near optimal communities by maximizing the modularity metric of the entire graph representing the social network as an objective function. The paper presents genetic algorithm ($\mathcal{GA}$) based tabu search ($\mathcal{TS}$) method referred as $\mathcal{HGT}$ for community detection in social networks. The TS algorithm is used as local search technique, by this way, the exploitation ability of the GA will be improved since the $\mathcal{HGT}$ algorithm has been taken the properties of the genetic algorithm ($\mathcal{GA}$) and the $\mathcal{TS}$ algorithm. These properties consist of the large space exploration of $\mathcal{GA}$ and the neighbourhood exploration and prohibitions of $\mathcal{TS}$. The organization of paper is as following: Section 2 introduces the community detection problem. The proposed approach is presented in section 3. Section 4 covers the implementation and tests. In section 5 we scan the related work. Finally, section 6 concludes this paper.

II. Related Work

As mentioned before, the HGT approach is compared with 4 algorithms, louvain, greedy (Greedy), the MENSGA genetic algorithm and the G-N Edge betweenness centrality algorithm. The choice of these algorithms is dictated by the fact that they represent on the one hand the different types of approaches (agglomerative, divisive and metaheuristic) of community detection, and on the other hand, they are well known as the best algorithms and considered as good benchmarks for a performance comparison in terms of the accuracy in community detection algorithms development.

1) The Louvain algorithm [1] 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

978-1-7281-6799-2/20/$31.00 ©2020 IEEE

density. initially, every vertex belong 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 hours).

2) Greedy is one of the agglomerative approaches. It has been introduced by Girvan and Newman. At each step, the algorithm tries to merge communities in order to increase the value of modularity Q [2]. Initially, each vertice is considered as a community. Then, a merging of pairs of neighbouring communities is performed to maximize the modularity Q. However, the algorithm do not merge pairs of communities between 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.
3) The Edge betweenness centrality algorithm [3] starts by calculating the centrality for each edge. Then, the edge which 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. For the choice of the best level of partition from a dendrogram, we use modularity Q. As for each partition obtained, the value of modularity Q is recalculated. The drawbacks of this algorithm is shown in calculating the measure of centrality where it is a process too slow because a course by all possible paths between all the pairs of vertices must be made for each link.

4) MENSGA [4] is a genetic algorithm 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.

III. Community Detection Problem

The objective of community detection is to partition the graph representing the 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, we use the modularity Q [5] 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 $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes and $E = \{e_1, e_2, ..., e_m\}$ is the set of links. The objective is to find a partitioning which gives the best community structure $C = \{c_1, c_2, ..., c_k\}$, i.e. a maximum value for the Q function. The modularity cost function is defined by

$$Q = \frac{1}{2*m} \sum_{ij} \left(B_{ij} - \frac{d_i d_j}{2*m} \right) \delta(K_i, K_j) \qquad (1)$$

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 δ is the Kronecker delta function which evaluates to one if nodes i and j belong to the same community and zero otherwise. The community detection problem has been showed to be NP-hard [6]. We omit details due to lack of space.

IV. Hybrid Genetic-Tabu for Community Detection

As mentioned previously, we investigate the use of a hybrid genetic algorithm ($\mathcal{GA}$) and tabu search ($\mathcal{TS}$) for the community detection problem in social networks. Therefore, a novel hybrid genetic approach called ($\mathcal{HGT}$) is proposed and compared with the state-of-the-art approaches. The basic idea of $\mathcal{HGT}$ is to refine the solution found by $\mathcal{GA}$ [7] using $\mathcal{TS}$ [8] which is an evolutionary heuristic that updates a single solution. Hence, the rationale behind the $\mathcal{HGT}$ approach is to start from an already good solution given by $\mathcal{GA}$ and successively move it to one of its current neighbours using $\mathcal{TS}$ always with the aim of improving the modularity $\mathcal{Q}$. Figure 1 shows a global view of the proposed approach. Its key modules are described in detail in the following sections.

A. Genetic Algorithm for community detection

Genetic algorithms are optimization algorithms based on techniques derived from genetics and natural evolution. Genetic algorithms are population based optimisation. A population is set of elements called chromosomes. A genetic algorithm is used to determine the extrema(s) of a function defined on a search space. 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 (crossover, mutation, selection, etc.) (5) design parameters (population size, total number of generations, probabilities of application of crossover and mutation operators.

1) Chromosome Encoding: Chromosome encoding is a very important step in $\mathcal{GA}$ based approaches. It allows to describe how to associate a chromosome to solution. In $\mathcal{HGT}$, 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 $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\}$, where each $x_{i,k}$ 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.

This encoding scheme is simple but it is not bijective. This means that the same community can be represented by several chromosomes.

B. Tabu-search for community detection

Tabu search 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 evolves 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 this intensification and diversification strategies, three lists are are maintained:

- $\mathcal{LTI}$: a tabu intensification list which is a medium term memory in order to avoid cycles in a local space.
- $\mathcal{LTD}$: 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.
- $\mathcal{LC}$: a candidates list used to store neighbour solutions if they are not already in the $\mathcal{LTI}$ and $\mathcal{LTD}$ lists.

1) 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.

C. HGT: the hybridization approach

The proposed hybridization consists of alternating the stages of the $\mathcal{GA}$ and $\mathcal{TS}$ global and local search processes to diversify and intensify the solutions. In fact, the $\mathcal{HGT}$ 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 solution are selected and inserted in the $\mathcal{LTI}$ list if they don't exist in the tabu list $\mathcal{LTD}$. Then, the local search is called again to generate a set of neighbours for each element of the $\mathcal{LTI}$ list. Solutions that do not already exist in this list are stored in the candidates list $\mathcal{LC}$. Meanwhile, two best solutions are select, one from $\mathcal{LTI}$ and the other from $\mathcal{LC}$ (representing the parents) to perform 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 (Figure 2) are discussed in the following steps:

1) Generation of an initial solution: $\mathcal{HGT}$ starts from an initial solution s_0. This solution can be generated by different methods, either from metaheuristics, heuristics, exact methods or by using a random solution.
2) 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 IV-B1). At each time a community is randomly changed (structurally we change the gene value). The resulting set forms an population P (the first generation) which may contains both optimal and not-optimal communities.
3) Evaluation and Selection of the best solutions: population P 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 $\mathcal{LTI}$ list if they do not exist in the $\mathcal{LTD}$ list. Recall here, that $\mathcal{LTI}$ (tabu list of intensification) is used to avoid being trapped in a local minima and $\mathcal{LTD}$ (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).
4) Generation of neighbours for each element of $\mathcal{LTI}$: after putting the solutions selected in the tabu list $\mathcal{LTI}$, 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 $\mathcal{LTI}$ and $\mathcal{LTD}$ lists, then it will be stored in the $\mathcal{LC}$ list (candidate list), else, an other neighbour is selected. This step allow to have new solutions that do not already exist.
5) Selection of parent solutions : in this step, we choose the two best solutions (individuals) as parents for next generations respectively from the $\mathcal{LTI}$ and $\mathcal{LC}$ tabu list. Recall here, that both parents should not be in the tabu list $\mathcal{LTD}$.
6) 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^*, S^*

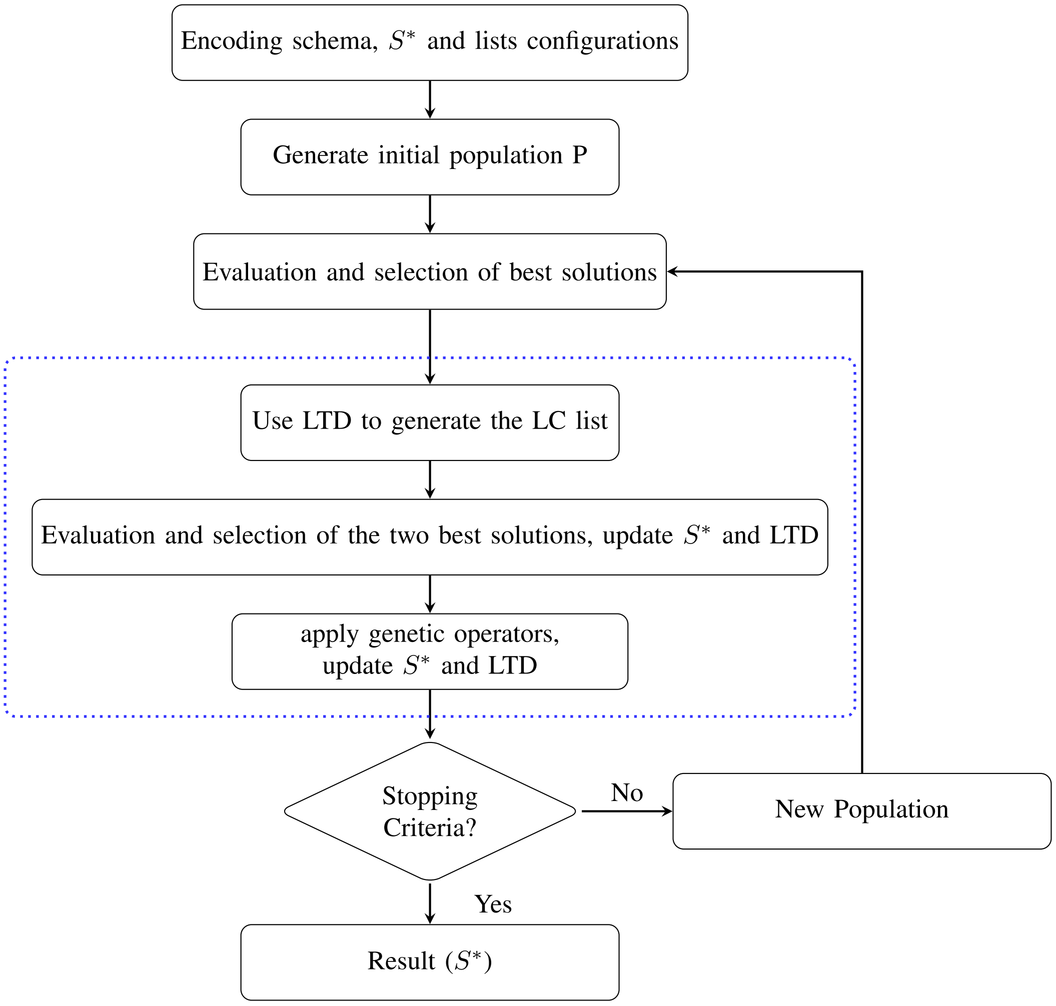
Encoding schema, S^* and lists configurations
Generate initial population P
Evaluation and selection of best solutions
Use LTD to generate the LC list
Evaluation and selection of the two best solutions, update S^* and LTD
apply genetic operators, update S^* and LTD
Stopping Criteria?
No
New Population
Yes
Result (S^*)

Fig. 1: HGT approach global view.

is updated and the the best parent added in the taboo list $\mathcal{LTD}$. Otherwise the process continue to the next step.

7) Updating tabu list $\mathcal{LTD}$: the tabu $\mathcal{LTD}$ list makes it possible to prohibit to return to an already visited solution. Therefore, at each iteration, the best solutions are saved in $\mathcal{LTD}$ with a precise duration where this tabu list must be updated by releasing certain prohibited solutions. This update of $\mathcal{LTD}$ makes it possible to give opportunities to prohibited solutions to be used in the next iterations.
8) 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).
9) 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 $\mathcal{LTD}$ list is updated and the best solution is accepted if its modularity is better than that of the currentS^*.
10) 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 $\mathcal{LC}$ list and the children solutions. Then, the process continue to the next iteration.

V. Implementation and Tests

To evaluate the effectiveness of present community detection system $\mathcal{CDS}$ at this stage, we it has been compared compared to other $\mathcal{CDS}$s. The approach is developed using python, optimized by speed, and run on an intel i3 processor, 2 Ghz with 4Go for RAM. The initial solution S is generated randomly using a Gaussian distribution via the algorithm proposed in [9]. Each resulting community have a size based on a variance of the community size distribution (T/V) where T is the average size and V is a shape parameter. Inside the same community, the vertices are connected with P_{in} probability and between communities with probability P_{out}. The number of communities depends on T; V and N (graph size). In our implementation we set the P_{in} value to 0.75 and

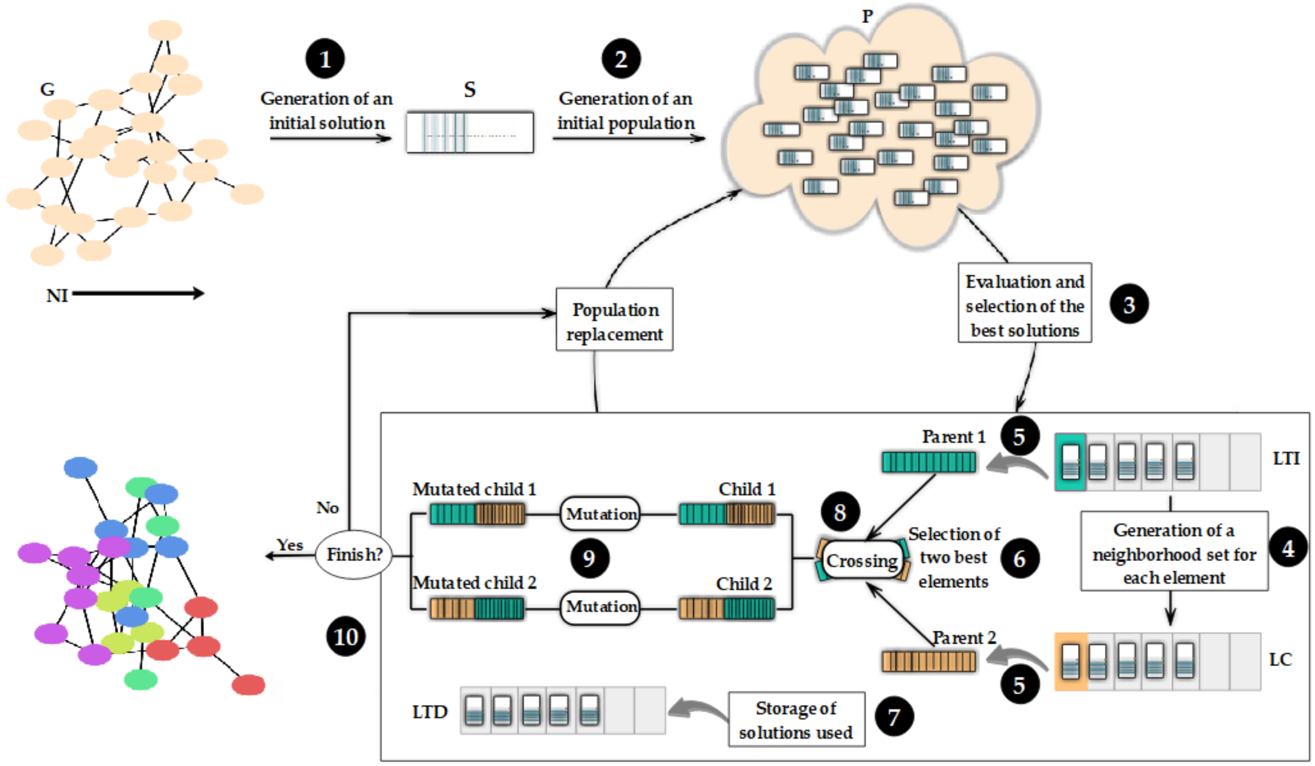
G
NI
1
Generation of an initial solution
S
2
Generation of an initial population
P
Evaluation and selection of the best solutions
3
Population replacement
No
Yes
Finish?
Mutated child 1
Mutation
Child 1
8
Parent 1
5
LTI
Selection of two best elements
6
Crossing
9
Mutated child 2
Mutation
Child 2
Generation of a neighborhood set for each element
4
Parent 2
5
LC
10
LTD
Storage of solutions used
7

Fig. 2: illustration of HGT to solve community detection problem

P_{out} value to 0.25. Many tests were conducted to select the best values for T and V according to the size of the network N. Table I shows the obtained values for different network sizes. Intensive tests were carried out on real world networks on whose best partitioning are known. Four networks are used, namely Club of karate of Zachary [10], Lusseau's dolphins [11] and Political books [11]. Figures 3, 4 and 5 depict the results of community clustering after applying $\mathcal{HGT}$ approach. It provides 4, 5 and 5 communities respectively for Karate, Dauphin and political Books. These number of community are the same obtained by the best algorithm of the state of the art.

The results obtained by the $\mathcal{HGT}$ approach are shown in the table II and 6. From the results it can be seen that: for the karate network, the value of the modularity obtained by HGT (Q = 0.420) is the best compared to the other algorithms (Louvain, MENSGA, G-N and Glouton). The community structure found by Glouton is 3 while for all others is 4 communities. Regarding the dolphin network, the modularity of HGT is at 0.519 with 5 communities which is similar to that of GN, greater than that found by Glouton (Q = 0.495 / 4 communities) and Louvain (Q = 0.518 / 5 communities) but lower than MENSGA which has a modularity of 0.527 with 4 communities. For the political book network, the value of Q obtained by HGT is 0.500 with 5 communities which is close to Glouton which has a value 0.502 with 4 communities. The other algorithms (Louvain, G-N, MENSGA) found the values of modularity at (0.527, 0.517, 0.526 resp.) With 5 communities. For the American football network, HGT finds 7 communities with a modularity of 0.48, unlike Louvain and MENSGA which reaches a value of 0.604 one with 9 communities and the other with 12. The G-N approach has partitioned this network in 12 communities with the value of Q equal to 0.592. Regarding the value obtained by Glouton Q is at 0.549 with 6 communities.

TABLE I: N, T and V values.

N	T	V
34	$\sqrt{N}/2$	0.35
62	$\sqrt{N}/2$	0.17
105	$\sqrt{N}/2$	0.15
115	$\sqrt{N}/4$	0.19

To conclude, the best results of the HGT approach shows its ability to detect communities with a population size of only 30 which needs 400 iterations to reach the optimal value of Q in small networks (Karate and dolphins). For large networks (Political books and American football) HGT has found a good partition but is not optimal compared to the other algorithms. From our first experiments, we found that the HGT approach depends on certain parameters such as those of the genetic algorithm and taboo research and in particular the size of the population and the number of iteration. A good configuration can lead to competitive results for large networks.

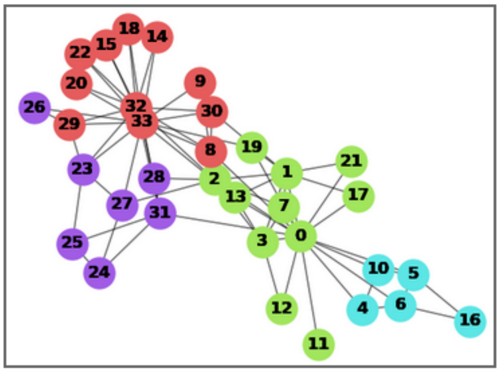

Fig. 3: HGT on the karate network with 34 vertices and 78 edges.

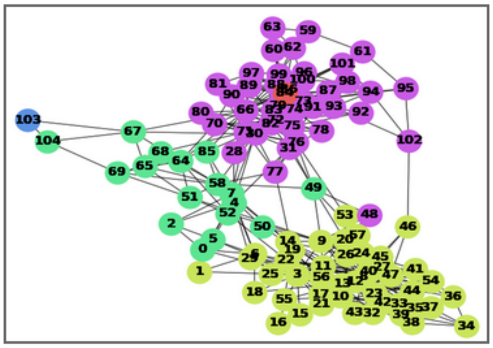

Fig. 4: HGT on the political books network with 105 vertices and 441 edges.

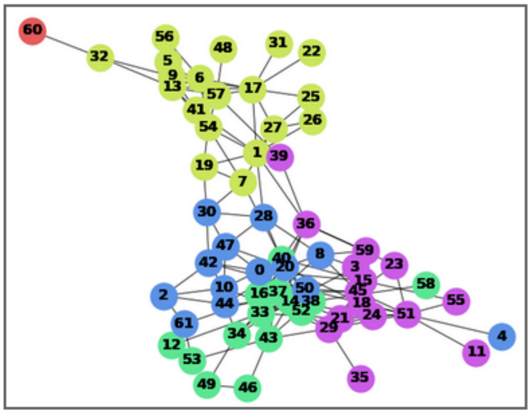

Fig. 5: HGT on the Dauphin network with 62 vertices and 159 edges.

TABLE II: Modularity and number of communities values of the HGT approach compared to state of art algorithms (with the population size equal to 30).

Networks / Methods	C.Karate		L.Dolphins		P.Books	
	Q	Nc	Q	Nc	Q	Nc
Louvain	0.419	4	0.518	5	0.527	5
Greedy	0.380	3	0.495	4	0.502	4
G-N	0.409	4	0.519	5	0.517	5
MENSGA	0.419	4	0.527	4	0.526	5
H-GT	0.420	4	0.519	5	0.500	5

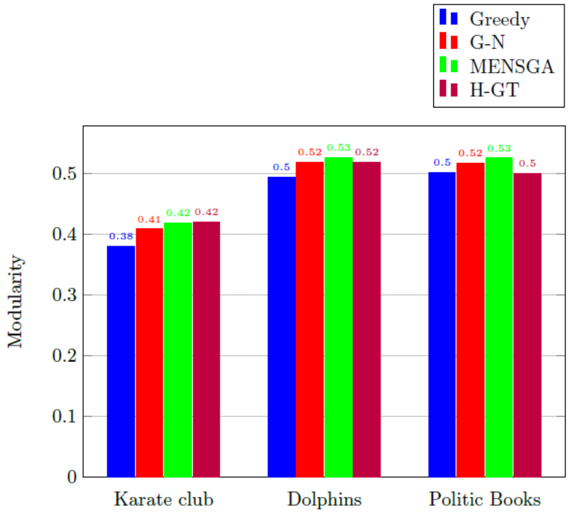

Fig. 6: HGT comparison with state of the art approaches

VI. Conclusion

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. The objective was the optimization of modularity metric Q. the approach was executed on real networks of different sizes. Experiments showed the capability of the HGT approach to correctly detect communities with comparable precision with state-of-the-art approaches. For medium-sized networks, we achieved good results where we clearly showed the capacity of our approach to correctly detect communities an outperforms 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. Future perspectives will focus on (1) results improvement for large scale networks (2) the application of the approach on concrete community detection problems (3) take into account the specificities of the data and the user profile to detect the domain or community type in concrete community detection problems (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. Finally, the literature review shows that there is not a better algorithm in an absolute sense. However, each algorithm can be efficient in very specific cases. It would be desirable to carry out additional empirical tests to better understand the strengths and weaknesses of our approach.

References

[1] V. D. Blondel, J. loup Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” 2008.

[2] Y. Slimani and A. Drif, “Découverte de communautés dans les réseaux complexes,” Oct. 2016, working paper or preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01389844

[3] L. C. Freeman, “Centrality in social networks: Conceptual clarification,” *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979. [Online]. Available: http://dx.doi.org/10.1016/0378-8733(78)90021-7

[4] Y. Li, G. Liu, and S.-y. Lao, “A genetic algorithm for community detection in complex networks,” *Journal of Central South University*, vol. 20, no. 5, pp. 1269–1276, May 2013. [Online]. Available: https://doi.org/10.1007/s11771-013-1611-y

[5] M. E. Newman, “Modularity and community structure in networks,” *Proc Natl Acad Sci U S A*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[6] R. Karp, “Reducibility among combinatorial problems,” in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[7] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[8] M. Gendreau and J.-Y. Potvin, “Tabu search,” 2007.

[9] U. Brandes, M. Gaertler, and D. Wagner, “Experiments on graph clustering algorithms,” vol. 2832, 11 2003.

[10] W. Zachary, “An information flow model for conflict and fission in small groups1,” *Journal of anthropological research*, vol. 33, 11 1976.

[11] G. Meiselwitz, *Social Computing and Social Media: 7th International Conference, SCSM 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2015. [Online]. Available: https://books.google.dz/books?id=CrQ0CgAAQBAJ