Heuristic search approach to community detection problem using ILS and UMDA algorithms

Javier Aguirre
University of the Basque Country
Irun, Spain
javiagu13@gmail.com

ABSTRACT

Community detection is a problem where in a given interconnected network different communities are found and analysed, nowadays, due to the increase on the use of social networks it is a very relevant problem. Being able to recognize communities allows companies to be able to make better recommender systems. Since CDP problem is NP-hard problem two different heuristic approaches are made. First, a non population based algorithm is used and second, a population based one. Iterated local search algorithm and univariate marginal distribution algorithm approaches will be made. It has been concluded that for CDP problem ILS has a better performance than UMDA.

ACM Reference Format:
Javier Aguirre. 2021. Heuristic search approach to community detection problem using ILS and UMDA algorithms. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Communities are part of broader network where a particular network may have multiple communities where nodes inside a community are densely connected. Nodes are strongly connected in communities and weakly connected between communities. This allows the detection of different communities.

Nowadays, in the case of social networks, a social network is a wide network made of communities, the detection of each community is of high importance in order to determine the community each individual belongs to, thus, allowing the creation of more accurate recommender systems and increasing the usage of the corresponding social network. This way the most accurate recommendations can be made enhancing the user experience by showing content interesting to the user.

Community detection is a complex problem therefore there is no exact method capable of solving it in an efficient time. An heuristic search approach is proposed since heuristics can find optimal solutions. This occurs because an "intelligent" type of search is used instead of trying all possible combinations without any kind of intuition.

Some heuristic approaches are better than others depending on the problem that is trying to be solved. A population based approach and a non population based will be tested to see the difference in performance.

2 COMMUNITY DETECTION

In this problem communities will be represented as a graph where each node will be an entity. Each node will be linked to other nodes and the nodes that are more densely linked among each other will create a community, as an example, a person will interact with another person and the people who interact the most among each other will form a community. As explained previously the goal is to find those communities.

For a long time different algorithms have been used, as an example: Kernighan-Lin [2] algorithm, spectral partitioning or hierarchical clustering. However, these algorithm are not the most efficient, therefore, more recently another algorithm has been widely used [1]. It performs a greedy search which enhances the results and is faster than Girvan and Newman [4] algorithm.

In order to find the different communities, modularity will be used which describes the separation of a network where maximum value will represent a greater separation. The aim is to maximize the modularity in our problem.

In our specific problem, a node will represent a researcher and the edges will represent the articles created among different researchers. The aim is to find the highest modularity so that the communities of researchers are found.

2.1 Community Detection Algorithm

As seen in [1] modularity is a property of a network and a specific proposed division of that network into communities. It measures when the division is a good one, in the sense that there are many edges within communities and only a few between them. The mathematical definition of modularity can be seen in the following equation where a_i is the fraction of ends of edges that are attached to vertices in community and e_{ii} is the fraction of edges that join vertices in community i to vertices in community i.

$$Q = \sum_i (e_{ii} a_i^2). \tag{1}$$

The objective is to maximize Q which means obtaining a broader community network. In order to get an in depth explanation of the algorithm review the paper [1].

2.2 Heuristic Search Approach

CDP is an NP-hard problem, therefore, two heuristic search approaches are proposed, iterated local search and univariate marginal distribution algorithm which is one type of estimation of distribution algorithms.

It has been proven that iterated local search can get a good performance in CDP [3] this is the reason why it was chosen. Iterated local search is an algorithm that uses a single solution and iterates it. The objective is to improve the quality of local optima. In this paper the initial solution will be generated at random within the predefined search space, and the stopping criterion will be a predefined number of evaluations of the objective function.

It has also been proven that EDA algorithms perform well in community detections as seen in [5]. Univariated marginal distribution algorithm is a population based algorithm, it uses a probability model to "evolve" the population this occurs by applying the model to the population and updating the model each evolution. A fixed number of population size describe the possible initial solutions and in the stopping criterion will be a predefined number of evaluations of the objective function.

3 CONTRIBUTION

On this paper the implementation of UMDA Algorithm and ILS Algorithm for community detection problem has been done. After the analysis, it has been concluded that iterated local search has a better performance than univariated marginal distribution algorithm for this particular problem.

4 EXPERIMENTATION

First hiperparameter tuning will be made for each of the algorithms and after, a comparison among algorithms will be made.

4.1 UMDA Hiperparameter Tuning

Let us tune the parameters of the UMDA algorithm in order to find the best combination. The aim is to achieve a certain combination of parameters that perform well for most community sizes.

Experimental setup

The maximum amount of evaluations performed by the algorithm will be set to 300 and the amount of individuals evaluated for building the probabilistic model of the UMDA will be set to 10.

The main setup will consist on testing all combinations of population sizes (25, 50, 75) and selection sizes (10, 15, 20) of the algorithm. Each combination will be executed 3 times and the average of the execution will be taken as the final value of the given execution.

Different community sizes (25, 50, 100) will be also taken into account to see if the algorithms hiperparameters get affected by it.

Hypothesis

It is expected that the performance of the algorithm gets affected by the amount of communities. However, certain similarities among executions are also expected, therefore, providing some general optimal hiperparameters.

Results

Let us see Figure 1, as expected similarities can be found in all three different heatmaps. It can be seen that the lower left corners are more brighter than the lower right corners. Also, in average lower selection size has a better performance than a higher selection size.

Conclusions

In conclusion, it is very clear that the combination of population size of 25 with a selection size of 15 has the best results in modularity, therefore, this is the combination that will be used.

4.2 ILS Hiperparameter Tuning

Let us tune the parameters of the ILS algorithm in order to find the best combination. The aim is to achieve a certain combination of parameters that perform well for most community sizes.

Experimental setup

The maximum amount of evaluations performed by the algorithm will be set to 300. An initial solution will be generated at random and using swaps the solution will be perturbed. The strength of the perturbation will define how many swaps are performed to the solution.

The main setup will consist on testing different perturbations [0 - 200]. Each perturbation will be executed 3 times and the average of the execution will be taken as the final value of the given execution.

Different community sizes (25, 50, 100) will be also taken into account to see if the algorithms hiperparameters get affected by it.

Hypothesis

It is also expected that the performance of the algorithm will vary depending on the number of communities but similarities among executions are expected in order to define general optimal hiperparameters.

Results

Let us see Figure 2, a clear similarity is found among plots and 25 can be chosen as the best perturbation scoring 0.032 of modularity. Modularity increases as perturbation intensity increases until arriving to 25 swaps. Then as perturbation intensity grows modularity decreases among executions.

Conclusions

25 is chosen as the best intensity of perturbation since it is the best performing value.

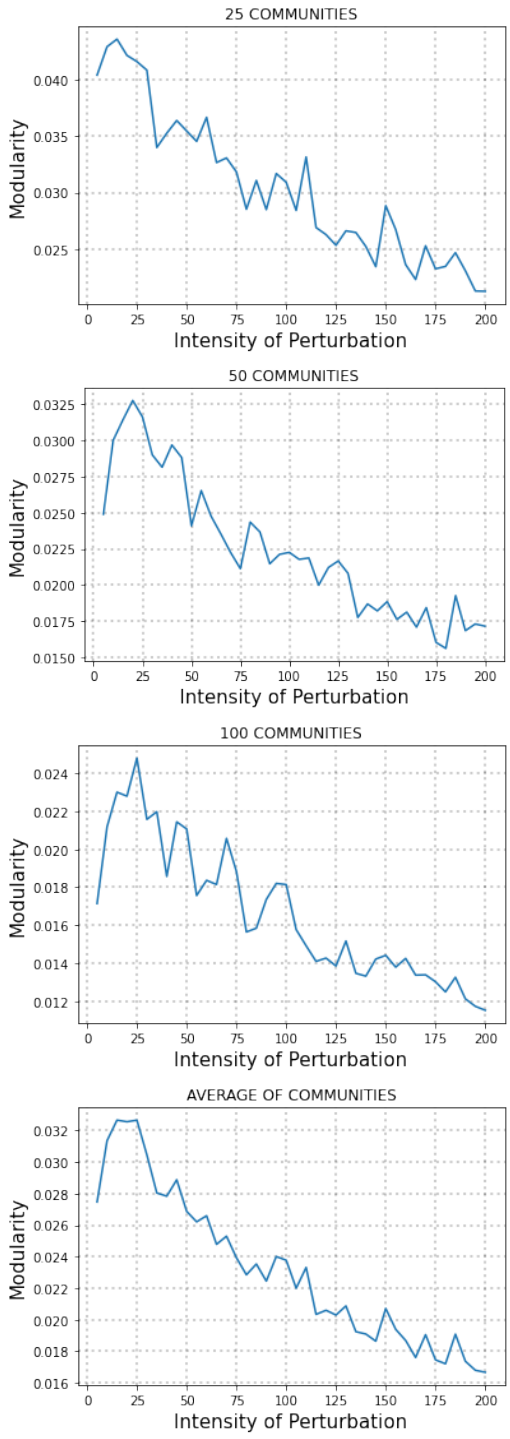

Figure 1: Executions with three community sizes are represented in a plot. The last plot is the average of all the graphics above. 25 is the best performing perturbation, thus, the chosen one.

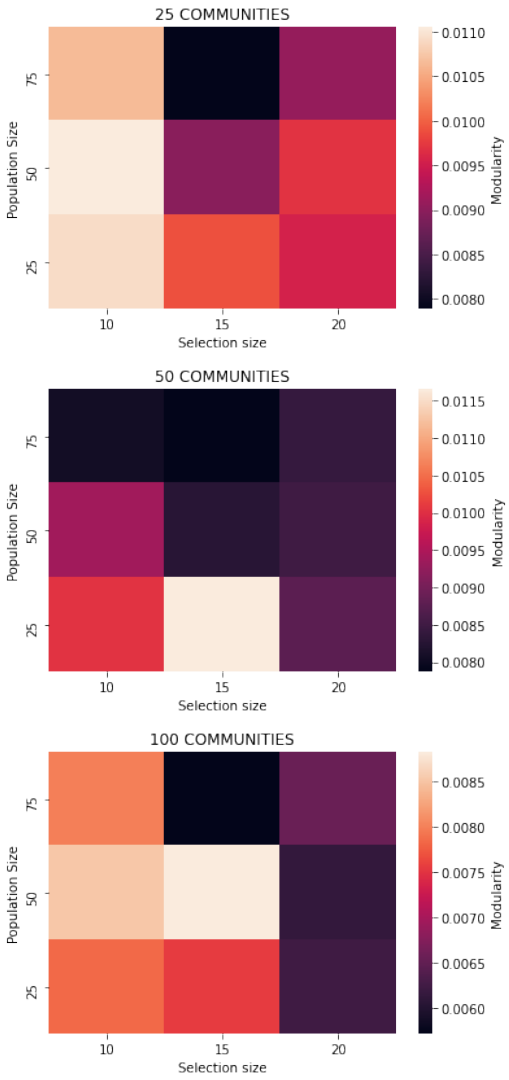

Figure 2: Executions with three community sizes are represented in a heat map. In all of them it can be seen that the best combination is a population size of 25 with a selection size of 15.

4.3 Algorithm Comparison

The aim of the algorithm comparison is to analyze the differences in the behaviours of both algorithms, to see if one is better than the other overall and to gain a better insight on why this may occur.

Experimental setup

For this experiment, the same amount of maximum evaluations 10^4 will be set. Under those conditions the performace of both algorithms will be evaluated. At each of the following number of communities [10,20,40,60,80,100] each algorithm will be run 5 times and the average of those 5 times will be plotted.

Also random search algorithm will be analysed as a baseline algorithm.

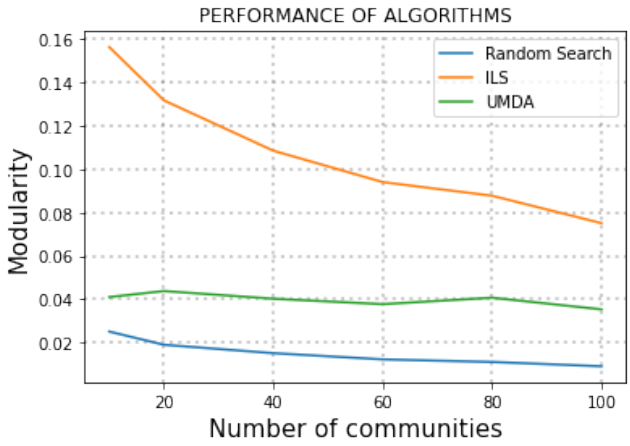

Figure 3: Algorithm comparison UMDA outperforms ILS and random search.

Hypothesis

Definitely ILS and UMDA are expected to perform better than random search.

Results

As seen in Figure 3 ILS performs better than UMDA and UMDA performs better than random search as expected.

Conclusions

In conclusion ILS performs better than UMDA for CDP problem. Also ILS performs two times better than UMDA.

5 CONCLUSIONS

A heuristic approach to the CDP problem has been proposed, for that ILS and UMDA metaheuristics have been used and the comparison of their performace have been made after hiperparameter tuning. As a future work a more thorough parameter tuning of UMDA should be made in order to truly confirm that ILS is better than UMDA.

REFERENCES

[1] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Phys. Rev. E* 70 (Dec 2004), 066111. Issue 6. https://doi.org/10.1103/PhysRevE.70.066111

[2] Brian W Kernighan and Shen Lin. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal* 49, 2 (1970), 291–307.

[3] Chao Liu, Qinma Kang, Hanzhang Kong, Wenquan Li, and Yunfan Kang. 2020. An iterated local search algorithm for community detection in complex networks. *International Journal of Modern Physics B* 34, 04 (2020), 2050013.

[4] Mark EJ Newman. 2004. Fast algorithm for detecting community structure in networks. *Physical review E* 69, 6 (2004), 066133.

[5] Fahong Yu, Meijia Chen, Kun Deng, Xiaoyun Xia, Bolin Yu, Huiming Gao, Feng He, Longhua Ma, and Zhao-Quan Cai. 2017. Community detection in the textile-related trade network using a biased estimation of distribution algorithm. *Journal of Ambient Intelligence and Humanized Computing* (2017), 1–10.