Analysis of the relation between the minLA problem and the visualisation of a graph spectrogram

Thibaut Eloy
IMT Atlantique
Brest, France
thibaut.eloy@imt-atlantique.net

Anthony Frion
IMT Atlantique
Brest, France
anthony.frion@imt-atlantique.net

ABSTRACT

The Minimum Linear Arrangement problem (MinLA) has been widely studied over the previous decades. On the other hand, graph signal processing through graph Fourier transform is still an emerging field with a lot of open questions : see e.g. [9]. The question treated in this paper is the choice of a permutation of columns to visualise a computed spectrogram. We show that this problem can be addressed by solving the MinLA problem and vice versa.

1 INTRODUCTION

The minLA problem consists in labeling the nodes of a graph with unique integers from 1 to the number of nodes in the graph, in a way that minimises the bandwidth sum, defined as the sum of absolute differences of labels for each pair of neighboring nodes. Therefore, a solution of this problem is equivalent to a permutation of the graph's nodes.

Similarly, one of the issues of graph spectrograms compared to spectrograms of classical signals is that, unlike the set of real numbers, the nodes of a graph have no trivial partial order. Thus, when plotting the spectrogram as a matrix, one has to choose the order in which the columns will be arranged. Since each column of the spectrogram corresponds to a node of the graph, this is again equivalent to finding a permutation of the graph's nodes.

Because of this similarity, one can very easily see a solution to the minLA problem as a permutation to be used for the spectrogram and vice versa. Thus, we are interested in checking if a good solution to one of these problem is good as well for the other problem. Apart from the simple ability to do so, there are intuitive ideas that let us think that doing so could be interesting. Indeed, a good solution to the minLA problem will likely have small differences between labels of neighboring nodes, and, conversely a spectrogram permutation is meant to highlight similarities between some node's spectrums, which often happen when they are neighbors in the graph.

The remaining of this article will be organised as follows :

In part 2, we explain exactly in which way we want to address the problem of finding an appropriate permutation to plot the specrogram. Indeed, unlike the minLA problem which has a clear mathematical formulation, the visualisation of a graph spectrogram has no objective function and is rather a problem that satisfies the human intuition to try to classify nodes. We will therefore define the graphs and signals that we use to compute our spectrograms, as well as a similarity measure that will enable us to quantify how good a permutation of columns is for a given spectrogram.

In part 3, we will first recall the formal definition of the minLA problem as well as some heuristics used to address it. We will then apply these heuristics and see how well the obtained solutions optimise our similarity measure from part 2.

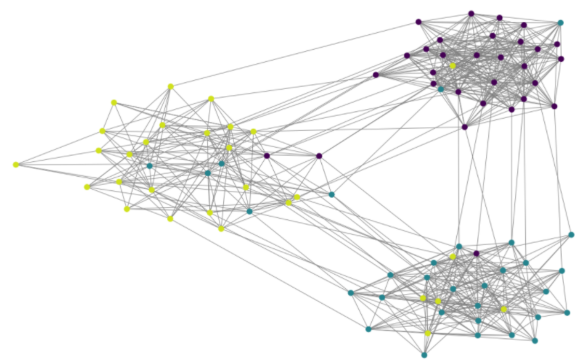

Figure 1: SBM graphs with 3 blocks each color is a group

In part 4, we will proceed in the opposite direction by trying to directly optimise our similarity measure for permutations of spectrograms, and seeing if the permutations that we obtain are also interesting to address the minLA occurrences on the concerned graphs.

2 PROBLEM FORMULATION AND DESCRIPTION OF THE METHOD

2.1 Graph model and signal used

We worked on graphs generated by a stochastic block model (SBM graphs) Figure 1. It is a generative model for random graphs which produces graphs containing blocks. Two vertex in the same block are more likely to be connected. We used three different blocks.

Then to generate a signal on this graph, we created 3 different groups of vertices. On each group we generated a signal with a given frequency, so that the signal is a mix between 3 frequencies. We could have chosen the three blocks as our three groups but we didn't because this method couldn't be generalized to other graph structures. So we built the three groups with a Bread-First Search (BFS) from three random vertices. An example of spectrogram obtained with this model can be visualized in Figure 2. We can clearly visualize the three frequencies. A random permutation of columns is used here, so the columns are not well arranged. One issue is then to find a 'good' permutation of the columns of the spectrogram.

2.2 Similariy indicator

In order to quantify the meaning of a 'good' permutation, we defined an indicator. This indicator must be high when we observe coherent bands in the spectrogram and must be low when the high value pixels are sparse along the spectrogram.

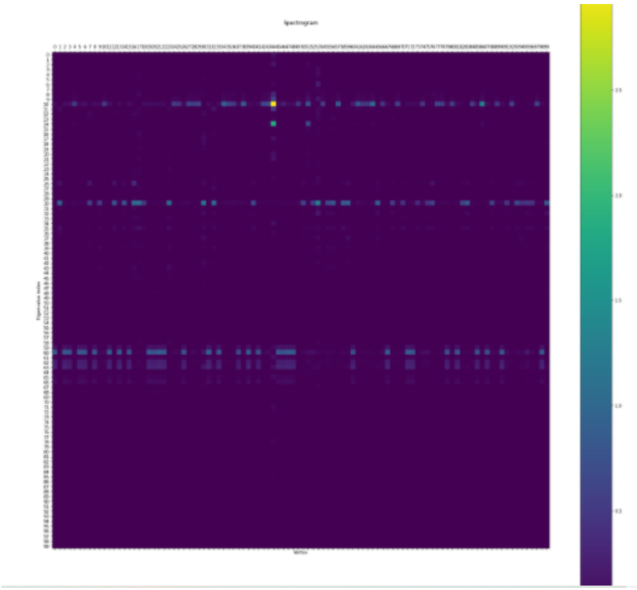

Figure 2: example of spectrogram on SBM with three frequencies. A random permutation of columns is used

We defined it as the sum of similarities between consecutive columns of the spectrogram. The similarity between 2 columns is 1 minus the distance between them. Mathematically the indicator can be expressed as :

$SM(S,P) = \sum_{j=1}^{N-1} \sum_{i=1}^{N} (1 - |S(i,P(j)) - S(i,P(j+1))|^a)$

With S the spectrogram obtained with permutation [1,2,...,N]. P the permutation used to reorder the columns. N the number of vertices and a the order of the norm used, we chose $a = 1$

This indicator favors putting similar columns next to each other so that we can observe bands.

2.3 Correlation between bandwidth and Similarity measure

In our research we tried to figure out whether the MinLA problem is equivalent to finding a good permutation of the column of a spectrogram. In other words, are the similarity indicator and the bandwidth sum correlated ?

3 DEFINITION AND OPTIMISATION OF THE MINLA PROBLEM

3.1 Definition

The minLA problem has been first described in [4]. It belongs to the category of graph layout problems (see e.g. [3]), which means that, for a given graph G = (V, E) with |V| = n, it consists in finding a bijective function

$\phi : V \longrightarrow [\![1, n]\!]$

to label the nodes of a graph G (or equivalently a permutation of the nodes) that minimises a cost function. The cost function to minimise here is the sum of absolute value differences between the labels of each pair of neighboring nodes :

$\mathrm{LA}(\phi) = \sum_{(i,j) inE} |\phi(i) - \phi(j)|$

This cost function is called the bandwidth sum of the arrangement ϕ. For this reason, the minLA problem is sometimes referred to as the bandwidth sum problem.

Note that the minLA problem is NP-complete, which means that, apart from some particular cases, its exact solution cannot be computed in a reasonable time. It is therefore solved through heuristics.

3.2 Heuristics

Along the years, many different algorithms have been proposed to solve the minLA problem. Some of them are quite simple and require little computing time. Let us cite the Cuthill-McKee algorithm [1], the spectral sequencing [5] and the McAllister algorithm [6].

The best performing algorithms to optimise the minLA are costly meta-heuristics such as simulated annealing and genetic algorithms : see e.g. [7] and [8]. However, since our goal here is not to recompute the best known solutions to minLA instances, we rather used Cuthill-McKee (CM) algorithm, spectral sequencing (SS) and McAllister algorithm.

We will not explain the principle of these algorithms here, but we encourage the interested reader to find it in the referenced articles.

First, spectral sequencing consists in retrieving the Fiedler vector, which is the eigenvector that corresponds to the second smallest eigenvalue of the graph's Laplacian matrix (the smallest eigenvalue being always 0) and ordering the nodes according to their coefficients in this eigenvector.

Cuthill-McKee algorithm consists in an improved breadth-first search (BFS) of the graph's nodes. Indeed, the idea is to start the BFS from the node (or one of the nodes) with the lowest degree and iteratively add nodes to your list of explored nodes using BFS, with the additional ideas that, at each iteration of the search, you don't add the neighbors of the current node in your list of nodes with a random order, but you instead sort them according to the index at which the earliest already visited node is located in your list. In case of tie, the algorithm prioritises nodes with the lowest degrees.

Finally, McAllister algorithm is an 'iterative construction' algorithm of another type. It again starts from the node of lowest degree (the original formulation suggests to start from a random noise, but we observed that making a nonrandom choice often gave better results) and gives it label '1'. Then, at each step, it looks at all the unlabeled neighbors to the labeled nodes and selects the one that minimises the following function : $\mathbf{sf(v) = d_U(v) - d_L(v)}$ $d_U(V)$ being the number of unlabeled neighbors of v, and $d_L(V)$ the number of labeled neighbors of v.

We also incorporated a way to quickly improve an existing solution, which has been described in [2]. The idea is to take an existing set of labels and, for each vertex of the graph, compute the median of the labels of its neighbors. Then, sort the vertices according to its medians and retrieve the corresponding permutation of the vertices to use it as a new set of labels.

3.3 Respective performances of the heuristics

We present here the performances of CM, SS and McAllister for addressing the minLA problem. We tested all of these heuristics on a set of 40 graphs randomly generated using the stochastic block model described in section 2.

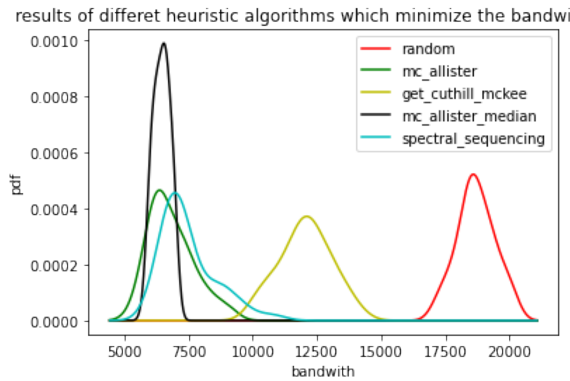

Figure 3: distribution of bandwidth sums found by the different algorithms on 40 different graphs

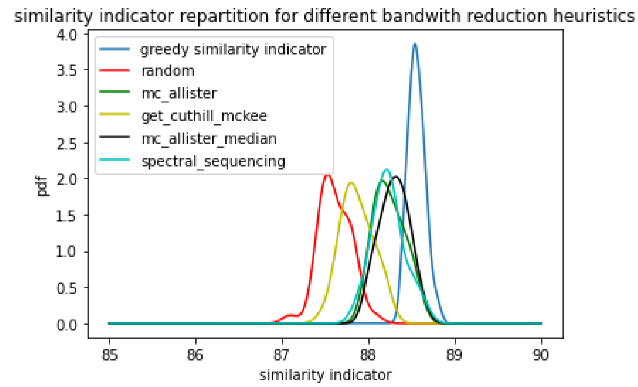

Figure 4: distribution of bandwidth sums found by the different algorithms on 40 different graphs

The results are summarized in figure 3. Note that we actually have a discrete set of 40 results for each algorithm, and we extrapolated continuous densities from these results using kernel density estimators (KDE).

Along with the Cuthill-McKee, spectral sequencing and McAllister results, we computed the results from the algorithm named 'mc_allister_median' which consists in performing a median improvement, as described in the previous subsection, on the set of labels obtained with the McAllister algorithm. We also used for each graph a random arrangement of vertices, which enables us to better compare the heuristics.

We can notice from this figure a clear hierarchy of performances between the presented heuristics : the best one is the McAllister improved with medians, followed by simple McAllister, then SS and finally CM. The poor results of CM are not so surprising since this heuristic is rather designed to address the bandwidth problem (another graph layout problem) yet this heuristic is undeniably better than a random choice of labels.

3.4 Usage of minLA solutions to plot spectrograms

In this subsection, we try using the solutions that we obtained with our minLA heuristics as permutations of the columns to plot spectrograms of the graphs.

It is first worth noticing that the minLA problem is fully defined by the graph's structure and involves no graph signal while a graph's spectrogram is always related both to a graph and to a signal that is applied to it.

That being said, we plotted on figure 4 the densities of similarity indicators with the permutations obtained from different minLA heuristics, on the same 40 graphs as in the previous subsection, on which were applied the signals described in section 1. We also plotted on this figure the density indicators of a random permutation generator and of a heuristic that actually has access to the signal's information and minimises the spectrogram's similarity indicator by greedily constructing a permutation. This greedy algorithm will be described with more details in section 4.

We can immediately notice from this graph that the minLA heuristics clearly give better results than random permutations for the spectrogram similarity measure, and that this measure actually increases with the quality of the heuristic : in other words, the better a heuristic performs on the minLA problem, the better it performs at finding good permutations for graph spectrograms. Indeed, we can see that the least effective heuristic for maximising the similarity measure is the Cuthill-McKee heuristic while the most effective one is the McAllister heuristic with median improvement.

This observation tells us that using a minLA heuristic on a graph should give us a permutation of its nodes that is likely to perform well when it comes to plotting the spectrogram of a signal on this graph.

3.5 Local Search

In order to highlight that the optimization of the bandwidth has a good influence on the ordering of the spectrogram columns, we performed local search on the bandwidth and looked at the evolution of the similarity indicator during the optimization process.

The results of the local search on four different graphs are summarized in Figure 5. We can notice from these graphs that, while the bandwidth is decreasing during the local search, the similarity indicator increases, which suggests again that the bandwidth and the similarity indicator are correlated.

4 OPTIMISATION OF THE SIMILARITY INDICATOR

We saw in the previous section that optimizing the bandwidth had a positive effect on the ordering of the columns of the spectrogram, which means that it also optimizes the similarity indicator. We will adress in this section the opposite question : if we find a good order for the columns, does it improve the bandwidth ? In other words, does the optimisation of the similarity indicator also optimise the bandwidth sum ?

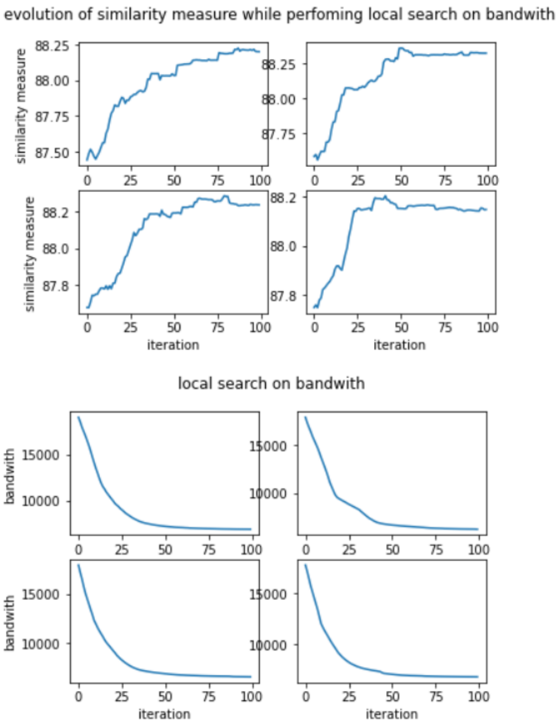

Figure 5: local search on bandwith

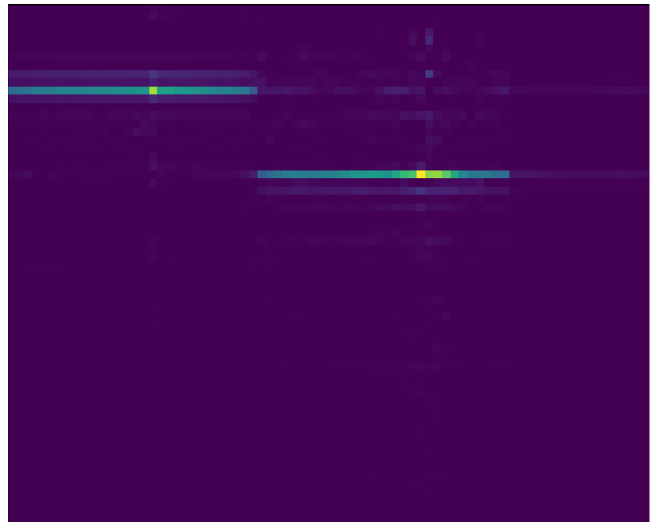

Figure 6: spectrogram with optimal solution for the similarity indicator

4.1 Algorithms

We developed 2 types of algorithms to optimize our indicator : algorithms which do not directly optimize it but are supposed to improve its value, and algorithms which directly optimise it

4.1.1 indirect optimisation. We can notice that most of the relevant information in the spectrogram is contained in the N rows of highest intensity with N the number of groups, each of them representing a frequency. We can then use the information in this N (3 in our case) rows to find a good permutation.

The first simple algorithm we implemented consists in sorting of the highest intensity row. We then reorder the columns of the spectrogram according to this order. It gives significantly better results than a random permutation but it is not ideal since we do not use the information of the other frequencies.

A first approach to be able to use the information of all the N highest intensity rows is to sum them and sort the results, and again use this permutation to reorder the column.

A second approach is to sort these N rows independently and to combine the information of the N permutations found. We combined it by keeping the first halve of each permutation.

4.1.2 direct optimisation. The problem of optimizing our indicator is similar to the travelling salesman problem (TSP). However there is a small difference : our problem does not impose to come back to the starting vertex at the end.

We first solved this problem with a greedy algorithm : we start our solution with the column of highest intensity and then build it greedily by iteratively adding the column which has the highest similarity with the previous one.

We then solved it exactly using a Mixed Integer Programming model that we solved with the Gurobi solver. The input is the adjacency matrix of our problem, i.e the matrix which contains at position (i, j) the similarity between columns i and j. The output of this algorithm is the exact solution of the corresponding TSP, which imposes to come back to the initial vertex. As a results it can give solutions in which a spectral band starts at the right of the spectrogram and ends at the left. To overcome this issue we slighty modified the solution by applying a translation to it so that the similarity indicator is optimized.

4.2 Results

Similarly to what we did in Section 3, We tested our algorithms on 40 randomly generated SBM graphs. We then plotted the distribution of the similarities found using a kernel density estimator Figure 7. In another figure we plotted the bandwidth distribution of the corresponding permutations. Figure 8. To have a reference, we also plotted on this figure the bandwidth repartition found by the McAllister algorithm presented in section 3.

The first important point to notice is that the results on the bandwidth are significantly better than the one we get with a random permutation. The second point is that the algorithms which give good results on the similarity indicator are also the one which give good results on the bandwidth. From the best to the worst results the order is : best permutation - greedy permutation - sorting sum of rows - sorting highest intensity row - first halves rows. It is the same order for the similarity indicator as for the bandwidth sum.

This observation tells us that using a permutation which performs well in arranging the column of a spectrogram is likely to result in a good solution for the minLa of the corresponding graph.

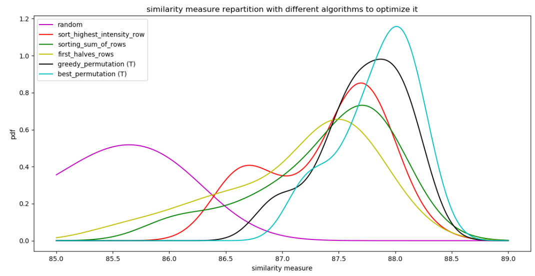

Figure 7: repartition of similarities found by the different algorithms on 40 different graphs

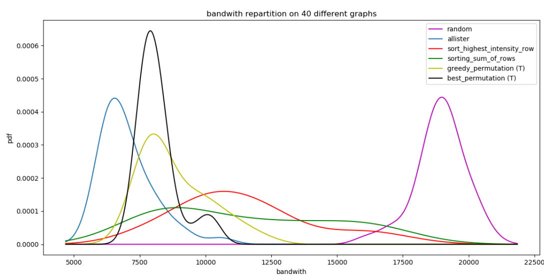

Figure 8: distribution of bandwidths obtained with different algorithms optimising the similarity indicator

5 CONCLUSION

In this article, we defined an indicator to quantify the quality of a given permutation of columns for a graph spectrogram, and showed that this indicator was strongly negatively correlated with the bandwidth sum of the associated permutation for any signal of the structure that we used. Moreover, there is no one-sense causality in this correlation : indeed, all of the algorithms that we used to address one of the problems was also good at solving the second problem, and their respective performances on the other problem were always ranked in the same way as their performances for solving the problem for which they were originally defined.

Using this correlation, one could use very fast minLA algorithms to quickly choose a permutation to visualise a spectrogram or identify groups of vertices in a graph with a given signal.

Also, even though our algorithms for optimising the spectrogram did not outperform the McAllister algorithm for the minLA problem, it got sufficiently close to it to give us hope that, with a well chosen set of signals, optimising our similarity indicator for the spectrograms might be a very good way to solve the minLA problem, as a whole heuristic or as an auxilary to a more complex metaheuristic.

ACKNOWLEDGMENTS

We want to thank Bastien Pasdeloup, researcher in IMT Atlantique, for suggesting us to work on this subject and helping us throughout our research.

REFERENCES

[1] E. Cuthill and J. McKee. 1969. Reducing the Bandwidth of Sparse Symmetric Matrices. In *Proceedings of the 1969 24th National Conference (ACM '69)*. Association for Computing Machinery, New York, NY, USA, 157–172. https://doi.org/10.1145/800195.805928

[2] A Duarte, JJ Pantrigo, V Campos, R Martí, and ETSII-URJC Madrid. [n.d.]. Heuristics for the Minimum Linear Arrangement Problem. ([n. d.]).

[3] Josep Díaz, Jordi Petit, and Maria Serna. 2002. A Survey of Graph Layout Problems. *ACM Comput. Surv.* 34 (09 2002), 313–356. https://doi.org/10.1145/568522.568523

[4] L. H. Harper. 1964. Optimal Assignments of Numbers to Vertices. *J. Soc. Indust. Appl. Math.* 12, 1 (1964), 131–135. https://doi.org/10.1137/0112012 arXiv:https://doi.org/10.1137/0112012

[5] Martin Juvan and Bojan Mohar. 1992. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics* 36, 2 (1992), 153 – 168. https://doi.org/10.1016/0166-218X(92)90229-4

[6] Andrew McAllister et al. [n.d.]. A new heuristic algorithm for the linear arrangement problem. ([n. d.]).

[7] Jordi Petit. 2003. Combining spectral sequencing and parallel simulated annealing for the MinLA problem. *Parallel Processing Letters* 13, 01 (2003), 77–91.

[8] Eduardo Rodriguez-Tello, Jin-Kao Hao, and Jose Torres-Jimenez. 2005. Memetic algorithms for the MinLA problem. In *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, 73–84.

[9] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30, 3 (2013), 83–98. https://doi.org/10.1109/MSP.2012.2235192