User guide *insert_name* Visual Tools

by Daniel-Stefan Radu

This is a manual for the use of ***insert_name* Visual Tools**. Each of them works independently and does not share memory space with the others. This manual will explain in detail each of the tools' uses and provides a basic understanding of the process. In order to fully understand the topic, knowledge is required regarding LDPC (Low Density Parity Checks), PEG (Progressive Edge Growth) as well as what an H matrix is in this context. The following abbreviations will be used:

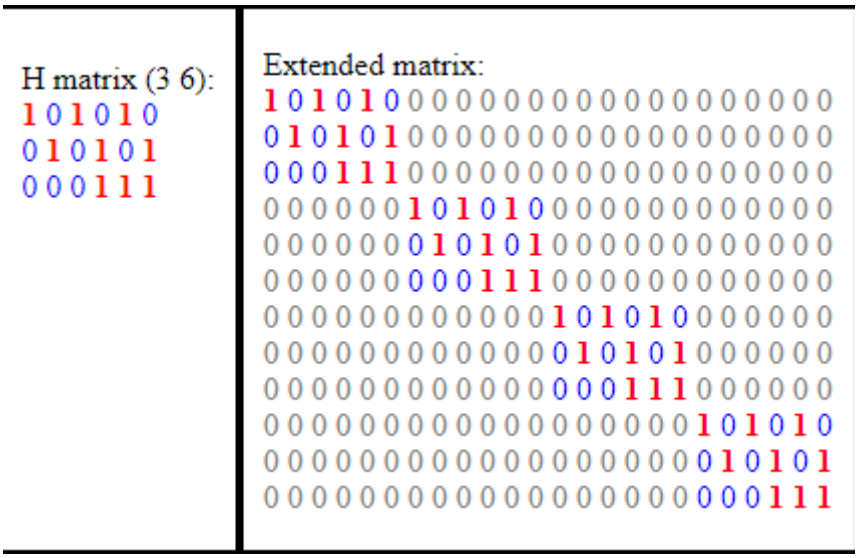
- **LDPC** = Low Density Parity Checks
- **PEG** = Progressive Edge Growth
- **HM =** H matrix, a matrix formed with only 1's and 0's that represent the bonds between nodes and factors, the nodes are the columns and the factors are the rows. If the location [i, j] is a 0, then no bond between factor i and node j exists. Similarly if it's a 1, then a bond (connection) exists.
- **EM** = Extended Matrix, a larger matrix formed by copying the HM diagonally and filling the empty space with zeros.
- **SM_x** = "Shifted Matrix step x"; an EM will have multiple steps of row-shifts applied to it, rerouting the factor graph to eventually make the final product. SM_0 is identical to the EM, SM_1 shows the first change, SM_2 shows the next change, and so on.
- **Module** = As you will read later, each change in the factor graph is "mirrored", or modular, by design. The term Module refers to the smaller part of the factor graph that directly concerns the nodes and factors from the original HM. (As a reminder, the EM is initialized from multiple copies of an HM; these are the modules, and any change made to a module will be done for all modules in parallel).

One example of the process and abbreviations:

An H matrix (HM):

1	0	1	0	1	0
0	1	0	1	0	1
0	0	0	1	1	1

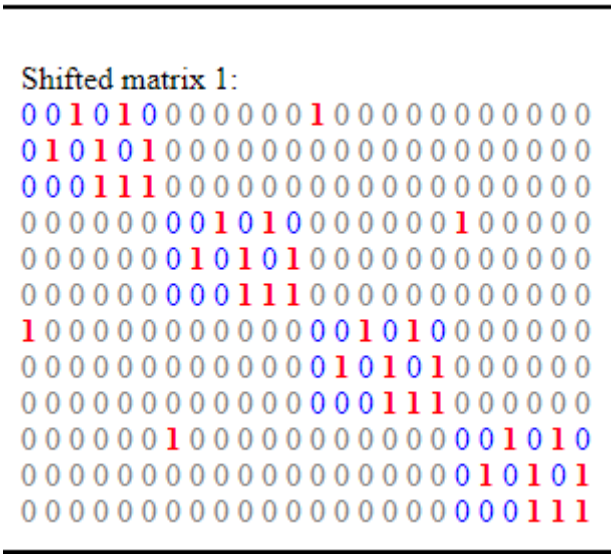
An Extended Matrix (EM), using 4 repetitions of the H matrix (HM) (R=4):



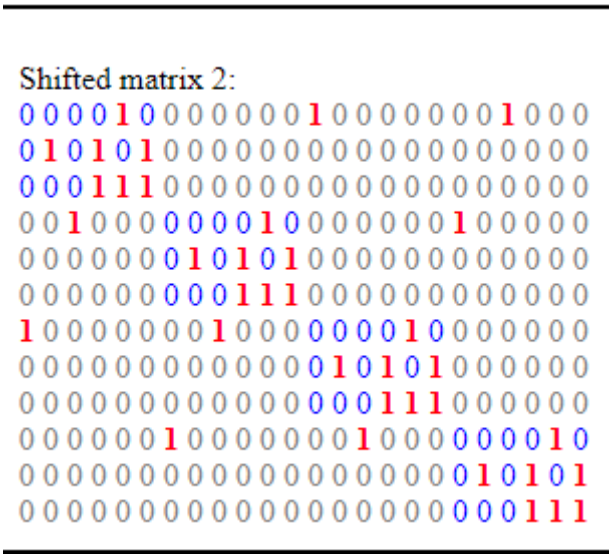
Note: At this step we can see the Modules clearly.
The modular nature is parallel at first; they do not interact with each other. Applying shifts will make these modules more integrated.

Now: Taking the above Extended Matrix (EM)
First: Find the edge connecting node 1 to factor 1 and apply a shift of 2.
Next: Find the edge connecting node 3 to factor 1 and apply a shift of 1.

This would be represented as SM_1: (row 1, col 1, S=2)



Followed by SM_2: (row 1, col 3, S=1)



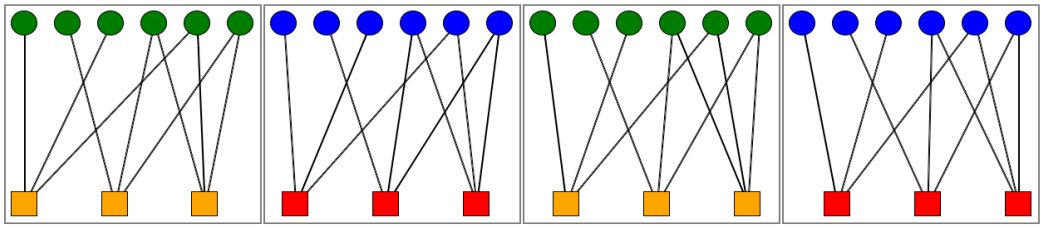
Note: SM_0 is identical to the Extended Matrix (EM).
Note: The selected row and column cannot be a zero. Since these matrices represent edges in a factor graph, shifts can only be applied to nodes and factors that are connected.

From now on, all the explanations will be related to the usage of the tools to manipulate and interpret the defined matrices.

1. Nodes and Factors Graph

This tool takes a text (**.txt**) file as input, containing two numbers on the first line, separated by a space which represent the dimensions of the matrix (**rows followed by columns**) and the matrix (on the following lines), containing only values of **0 or 1**. The rows represent the factors and the columns represent the nodes. The matrix is an H-matrix (see above). If and only if the value at row = x and column = y is 1, then factor x is **bonded/connected** with factor y. Any other input aside from 0 and 1 in the matrix is considered invalid and will result in an error. Once the input has been validated the user may proceed further. The aim of this tool is to display the matrices as text, as well as draw their graphs. A graph is a line of **nodes (circles)** that may be connected via edges (lines) to **factors (squares)**. Below there is a list and explanations of the rest of the features:

A. **"Set number of repetitions"**: sets how many times the root **H matrix** will be repeated when generating the **Extended Matrix**, or in other words, how many H matrix modules will be in the **EM**. Must be a positive integer.
B. **"Run"**: executes the input (if it is valid) and draws the graph according to the settings. For the graph, nodes will be represented by **circles** and factors by **squares**. On the right side of the panel the **H**, **EM,** and **SM**s, will be displayed as text.



C. **"Shift bond at […]"**: takes the input from the **"node = "** field, **"factor = "** field and the **"steps"** field and checks if the given node and factor are connected. If they are connected and the steps are a valid input (positive integer), then a shift will be applied and a new Shifted Matrix will be displayed in the right side panel. For execution, click **"Apply Shift"**.

- Advanced settings begin here, to display click the "show advanced settings" button -

D. **"Focus on central module"**: enables a highlight of the middle module and its components on the graph.
E. **"Hide noise to remove bonds"**: updates the graph to only display bonds connected to the nodes and factors in the middle/central module (includes connections leaving the module).
F. **"Hide Extended Matrix"**: hides the right side panel containing the E matrix.
G. **"Hide Shifted Matrix"**: hides the right side panel containing the S matrix.
H. **"Please select which matrix you would like to export"**: presents a list of options from the generated shifted matrices for exporting. Select an option from the dropdown list and click **"Export"** to download a text file containing

the chosen shifted matrix. Please see sections I, J and K below for additional options.

I. **“Choose the name of the file to export”**: allows the naming of the exported file. By default, it is “Matrix_” followed by the number of the shifted matrix (seen at the dropdown list mentioned at point H).

J. **“Use encoding”**: enables the usage of encoding on the exported matrix; enabled by default.

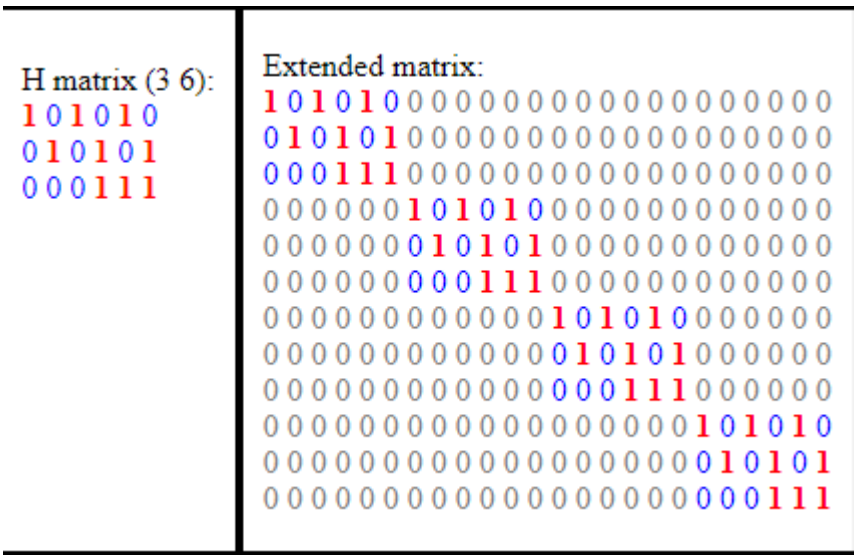
The encoded string follows the format:
num_rows/num_cols/encoded_values

num_rows is an integer, the number of rows in the matrix being exported.
num_cols is an integer, the number of columns in the matrix being exported.

encoded_values is a compressed string representing the values within the matrix. It can be composed from combinations of the following components.

Any subsequence of 1’s and 0’s that has less than three consecutive repeating digits may simply be written as a binary string (cannot contain 000 or 111). Whenever three consecutive repeating digits are encountered, compression should be used; a subsequence consisting of all 1’s (111…1) will be written as “/subseq_length/” and a subsequence of all 0’s (000…0) will be written as “.sebseq_length.”; all of this is written on a single line with no whitespace.

An example of encoding:



12/24/10101.20.10101.21./3/.24.10101.20.10101.21./3/.24.10101.20.10101.21./3/.24.10101.20.10101.21./3/

Notes:

- Hovering over a factor or a node on the graph will highlight it and all the directly connected factors/nodes to it.

- On each step of graph drawing, the most recent edge drawn will be colored in red, to distinguish it from the many other black edges.
- Checking/Unchecking any options would update the display accordingly.
- Each module is marked with a gray square border.
- The “targeting” of nodes and factors for shifting is done in natural language (index 1 represents the first element)

2. Matrix Wizard (Encoder/Decoder)

These tools allow for a quick view of the graph of an input matrix or the encoding/decoding of a matrix file respectively.

I. Matrix Encoder:

This tool takes a text (**.txt**) file as input, containing two numbers on the first line, separated by a space which represent the dimensions of the matrix (**rows followed by columns**) and the matrix (on the following lines), containing only values of **0 or 1**. Any other input aside from 0 and 1 in the matrix is considered invalid and will result in an error. Once the input has been validated the user may proceed further. The aim of this tool is to display the matrices as text and/or export its encoded version (see below). This is a simplified version of the tool “Nodes and Factors Graph”, for a deeper understanding, please read above. Below there is a list and explanations of the rest of the features:

A. **“Run”**: displays the matrix as text allows the encoding of the provided data.
B. **“Export”**: generates and downloads a text file with the encoded version of the matrix. For the full explanation of how this works, read the documentation or, for a short version, read at point 1.J. (above).
C. **“Choose a name [...]”**: allows for naming the file to be downloaded, by default it is “Encoded_Matrix_”.

II. Matrix Decoder

This tool takes a text (**.txt**) file as input, containing the encoded version of a matrix (see above). It is a mirror tool of the “Matrix Encoder”. Once the input has been validated the user may proceed further. The aim of this tool is to display the matrices as text and/or export its decoded version (see below) or display a simplified graph (see documentation or point 1.). This is a simplified version of the tool “Nodes and Factors Graph”, for a deeper understanding, please read above. Below there is a list and explanations of the rest of the features:

A. **“Run”**: displays the matrix as text allows the decoding of the provided data.
B. **“View Graph”**: displays a graph of the given matrix (see point 1. on how that works, this is a simplified version).
C. **“Export”**: generates and downloads a text file with the decoded version of the matrix. For the full explanation of how this works, read the documentation or, for a short version, read at point 1. first paragraph (above) on how the decoded file would look like.
D. **“Choose a name [...]”**: allows for naming the file to be downloaded, by default it is “Decoded_Matrix_”.

3. PEG Generator

This tool takes a text (**.txt**) file as input, containing a single line. Once the input has been validated the user may proceed further. The input is as follows: **number_of_nodes**: a list of degree distributions for each node, (positive integer list, delimited with the space-character), followed by "/" and **number_of_factors**: a list of degree distributions for each factor, (positive integer list, delimited with the space-character) followed by another "/". All of this is on a single line. Lastly, note that some 'impossible' inputs may be run; in these cases the algorithm will do its best to generate the H matrix as usual, but will flag any issues (like any unused degrees). The aim of this tool is to generate an H matrix based on the degree of distribution for a given number of nodes and factors. Below there is a list and explanations of the rest of the features:

A. **"Select the algorithm"**: contains a list of the currently implemented algorithms. Please check Attachment 1 for a full explanation on how each of them works. Once one has been selected, it will be applied to the input. Should the algorithm perform "badly" an error code will be given in the console stating the issue (bonds overlapping).
B. **"Run"**: executes the selected algorithm and displays the H matrix and its graph (*see point C) (see point 1).
C. **"View Graph"**: displays the graph, which will not automatically be displayed on **"Run"** if the input is on the larger side and could cause a larger computation time due to drawing elements.
D. **"Export"**: generates and downloads a text file with the matrix (not encoded).
E. **"Choose a name [...]"**: allows for naming the file to be downloaded, by default it is "PEG_Matrix_".