CODECHECK certificate 2025-003

https://doi.org/10.5281/zenodo.15333601

Item	Value
Title	Optimal Graph Stretching for Distributed Averaging
Authors	Florine W. Dekker (0000-0002-0506-7365), Zekeriya Erkin (0000-0001-8932-4703) and Mauro Conti (0000-0002-3612-1934)
Publication	https://doi.org/10.48550/arXiv.2504.10289
Publication repository	https://doi.org/10.4121/e64c61d3-deb5-4aad-af60-92d92755781f
Codecheckers	Aysun Urhan (0000-0001-8584-4736) and Yasel Quintero (0009-0005-9240-7370)
Date of check	2025-03-25
Summary	All figures in the manuscript were successfully reproduced, with the process requiring both an HPC cluster and a valid MATLAB license.
Codecheck repository	https://github.com/codecheckers/certificate-2025-003

Table 1: CODECHECK summary

Summary

The author provided detailed instructions on how to reproduce the results in the manuscript. The instructions include descriptions of the directory structure, and the files available to reproduce the results. As part of this CODECHECK, we used the scripts provided by the author to submit SLURM jobs (which ran MATLAB scripts) to first generate the results and then replicate the figures in PNG format. The codecheckers were able to accurately reproduce all the figures in the manuscript in a straightforward manner following the instructions provided by the author.

Output	Comment
fig1a_Proportion-of-Edges-Removed-During-Stretching.png	manuscript Figure 1(a)
fig1b_Number-of-Leaves-After-Stretching.png	manuscript Figure 1(b)
fig2_Metrics-After-Stretching.png	manuscript Figure 2
fig3_Convergence-Time-After-Stretching.png	manuscript Figure 3
fig4_Number-of-Leaves-After-Minimisation.png	manuscript Figure 4
fig5_Edges-Added-During-Leaf-Minimisation.png	manuscript Figure 5
fig6_Convergence-Time-After-Leaf-Minimisation.png	manuscript Figure 6
fig7_Edges-Changed-During-Optimisation.png	manuscript Figure 7
fig8_Convergence-Time-After-Optimising.png	manuscript Figure 8

Table 2: Summary of output files generated

CODECHECKER notes

The authors have made available all the necessary code and data required to reproduce figures in the manuscript. These resources can be accessed in the 4TU.ResearchData repository: https://doi.org/10.4121/e64c61d3-deb5-4aad-af60-92d92755781f.

Reproducing the figures required substantial computational resources, which made the use of a high-performance computing (HPC) cluster necessary. The author has included SLURM job scripts to facilitate running the application, computing the results, and generating the figures. Although these scripts were originally configured for TU Delft's DAIC HPC cluster, the codecheckers executed them on the DelftBlue HPC cluster. Adapting the scripts to this environment was straightforward.

Following the provided instructions, the codecheckers executed the SLURM scripts located in the *SlurmExamples* directory of the repository. To accommodate the limited resource access on the DelftBlue HPC cluster, modifications were made to the SLURM parameters in each script, including job runtime, number of CPU cores, and memory per CPU. The table below summarizes the SLURM jobs submitted by the codecheckers:

Job Script	Job Wall-clock time	Nodes	ntasks	CPU Cores	Memory per CPU
RunExperiments1.sbatch	00:18:42	1	1	32	1536M
RunExperiments2.sbatch	07:10:31	1	1	32	1536M
RunExperiments3.sbatch	19:38:38	1	1	32	1536M
RunExperiments4.sbatch	09:00:00	1	1	32	1536M
CreateFigures.sbatch	00:51:06	1	1	8	10240M

Additionally, the `srun` command was updated to comply with the MATLAB execution conventions on DelftBlue:

```
module load matlab/R2021b
SLURM_ARRAY_TASK_MAX=20 srun matlab -batch
"run('/scratch/$USER/codecheck/StretchSim/StretchSim/StretchSim.m'); exit;"
```

The SLURM job submission scripts used by the codechekers are available in the repository for this CODECHECK report.

The author also provided detailed instructions for reproducing the figures, including a list of third-party MATLAB dependencies and guidance on how to install them.

During the CODECHECK process, the author introduced a feature allowing users to select the desired output format for the figures. This feature removed the dependency on the matlab2tikz MATLAB plugin, which is now only required when the output format is set to `bin.tikz`.

Manifest Files

fig1a_Proportion-of-Edges-Removed-During-Stretching.png

Comment: Figure 1 (a) in the manuscript.

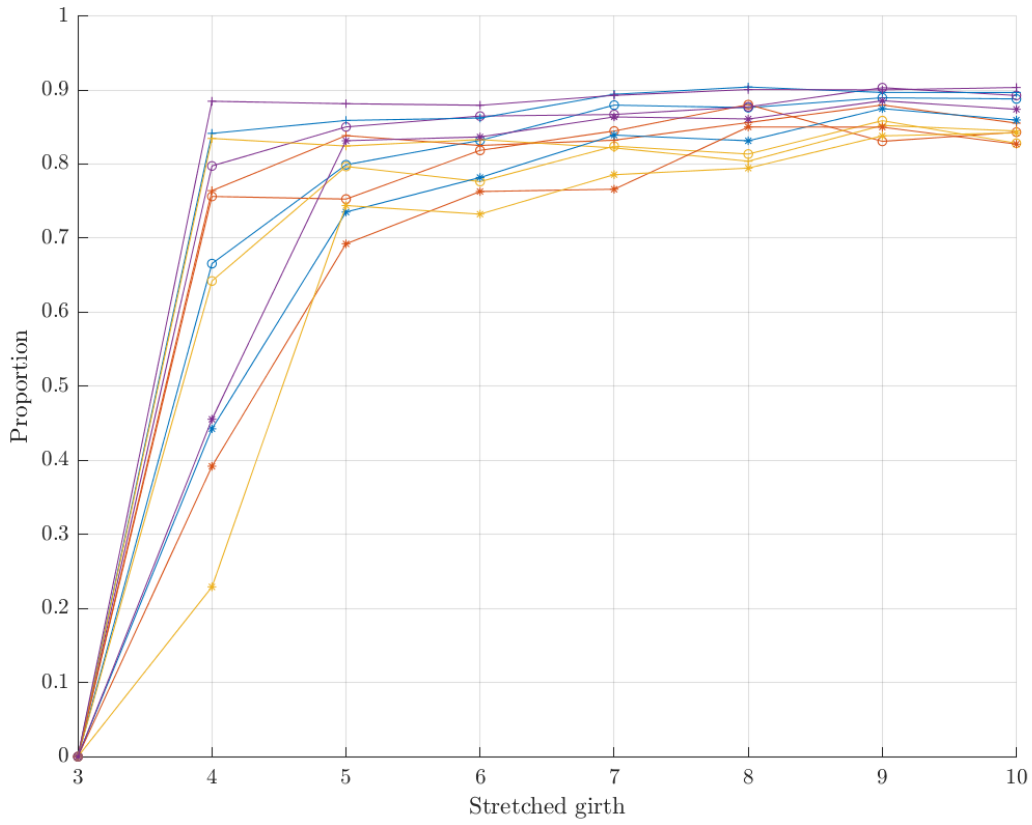


fig1b_Number-of-Leaves-After-Stretching.png

Comment: Figure 1 (b) in the manuscript.

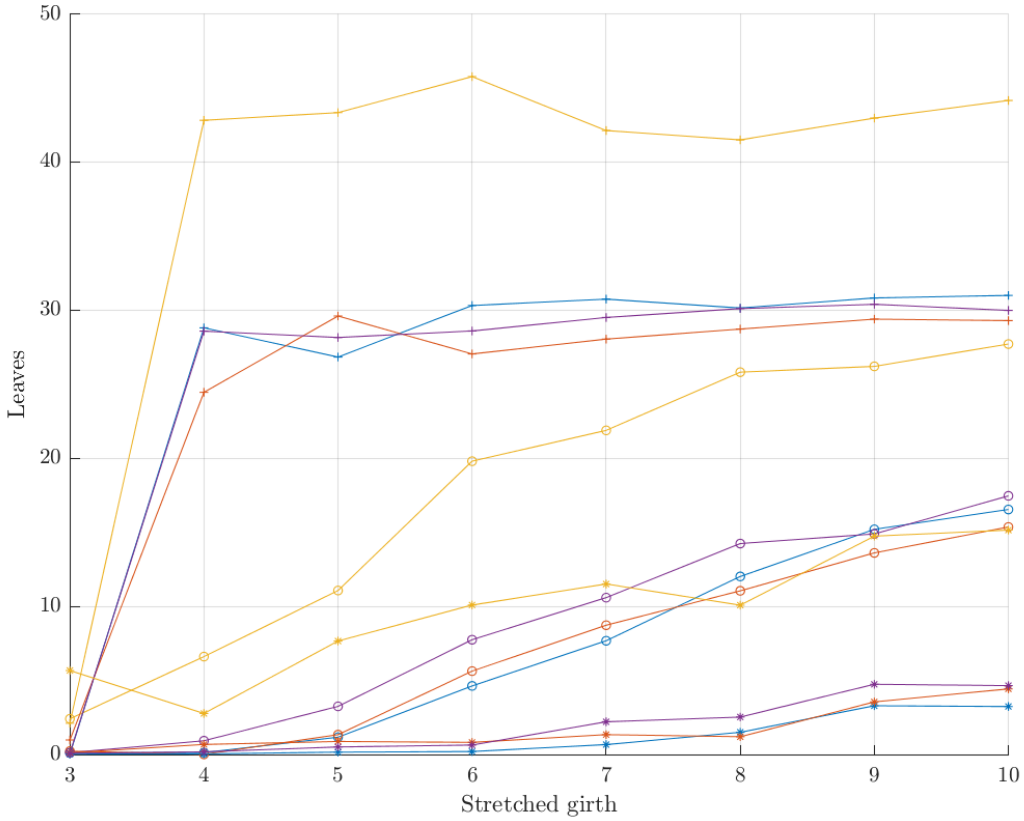


fig2_Metrics-After-Stretching.png

Comment: Figure 2 in the manuscript.

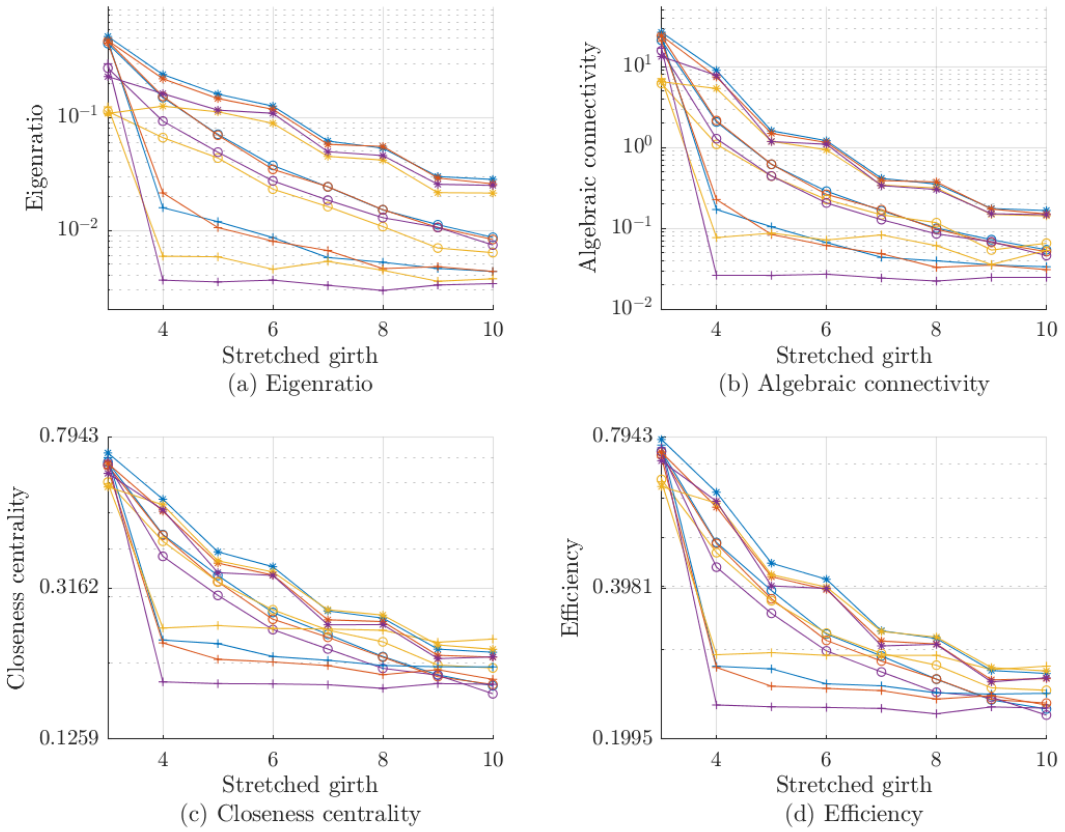


fig3_Convergence-Time-After-Stretching.png

Comment: Figure 3 in the manuscript.

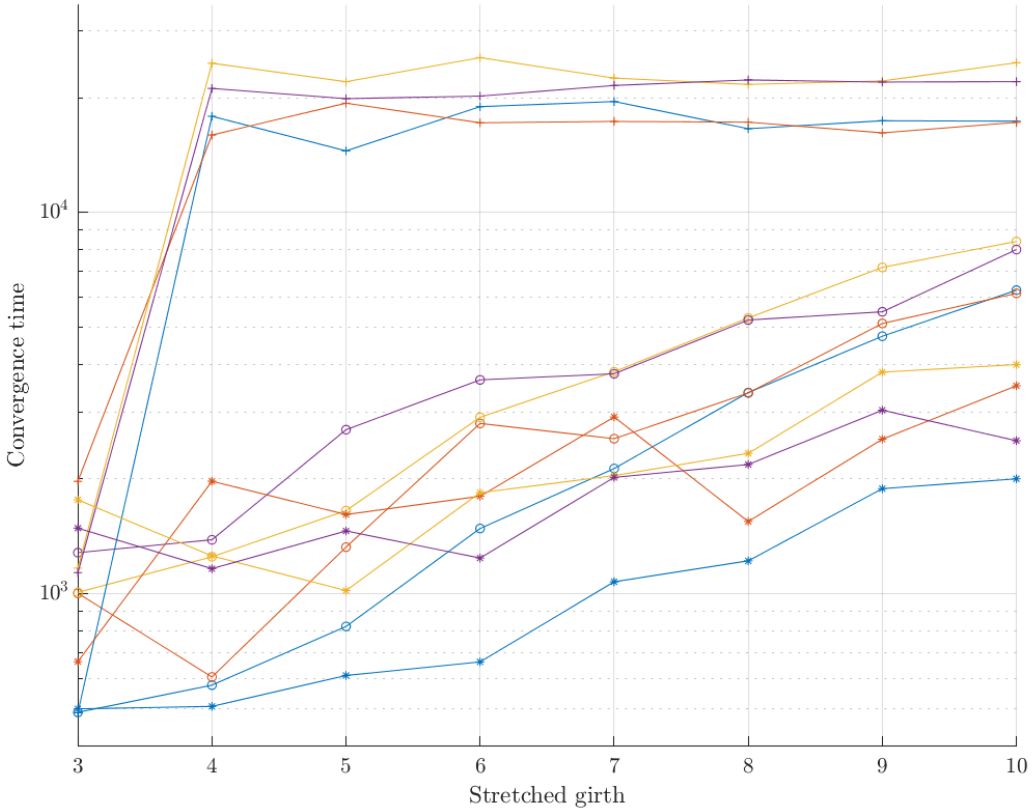


fig4_Number-of-Leaves-After-Minimisation.png

Comment: Figure 4 in the manuscript.

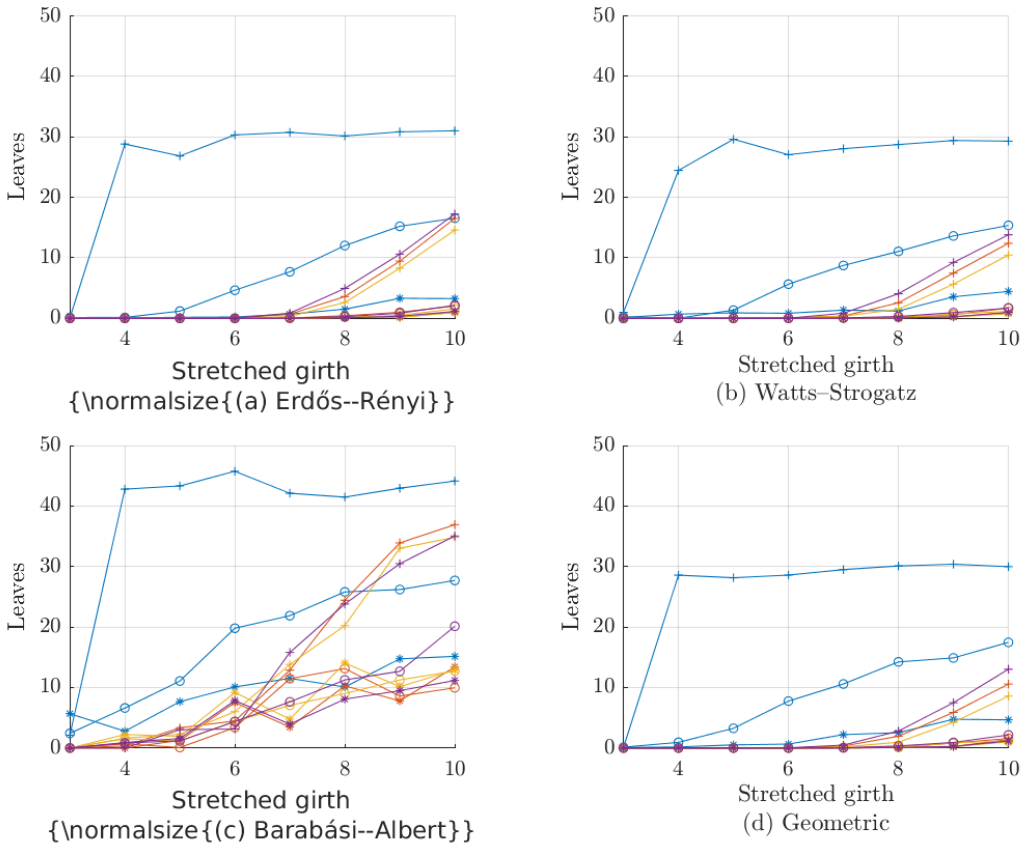


fig5_Edges-Added-During-Leaf-Minimisation.png

Comment: Figure 5 in the manuscript.

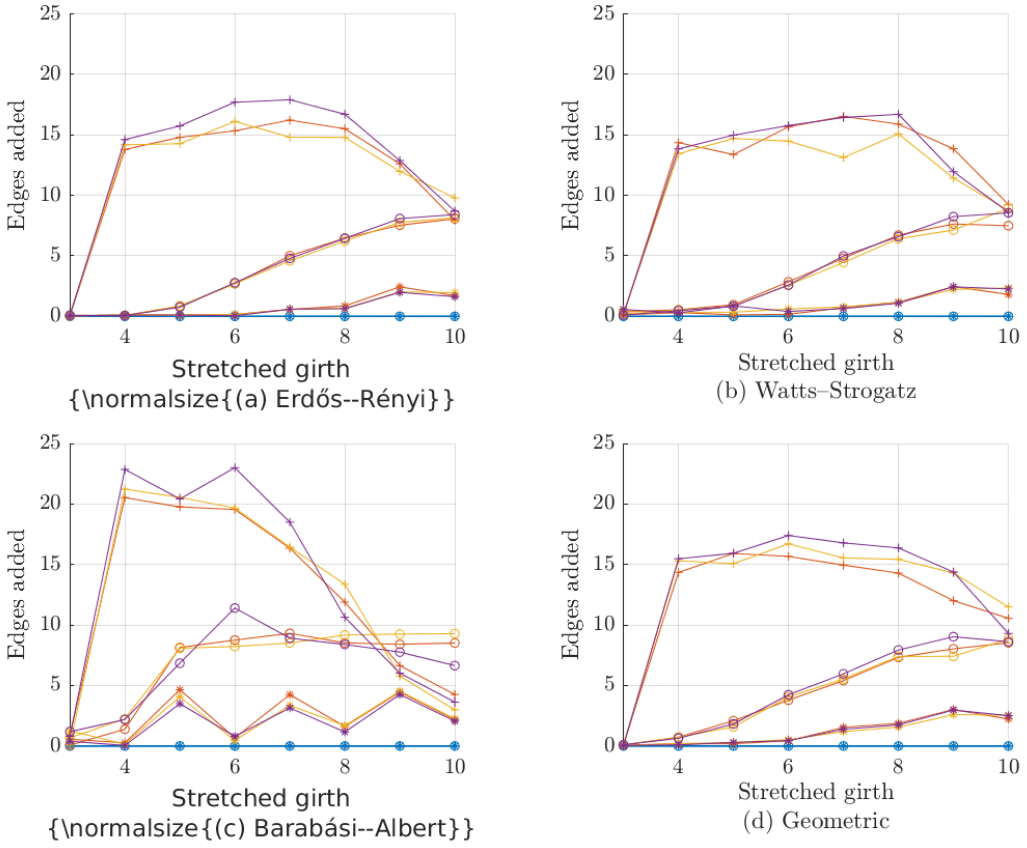


fig6_Convergence-Time-After-Leaf-Minimisation.png

Comment: Figure 6 in the manuscript.

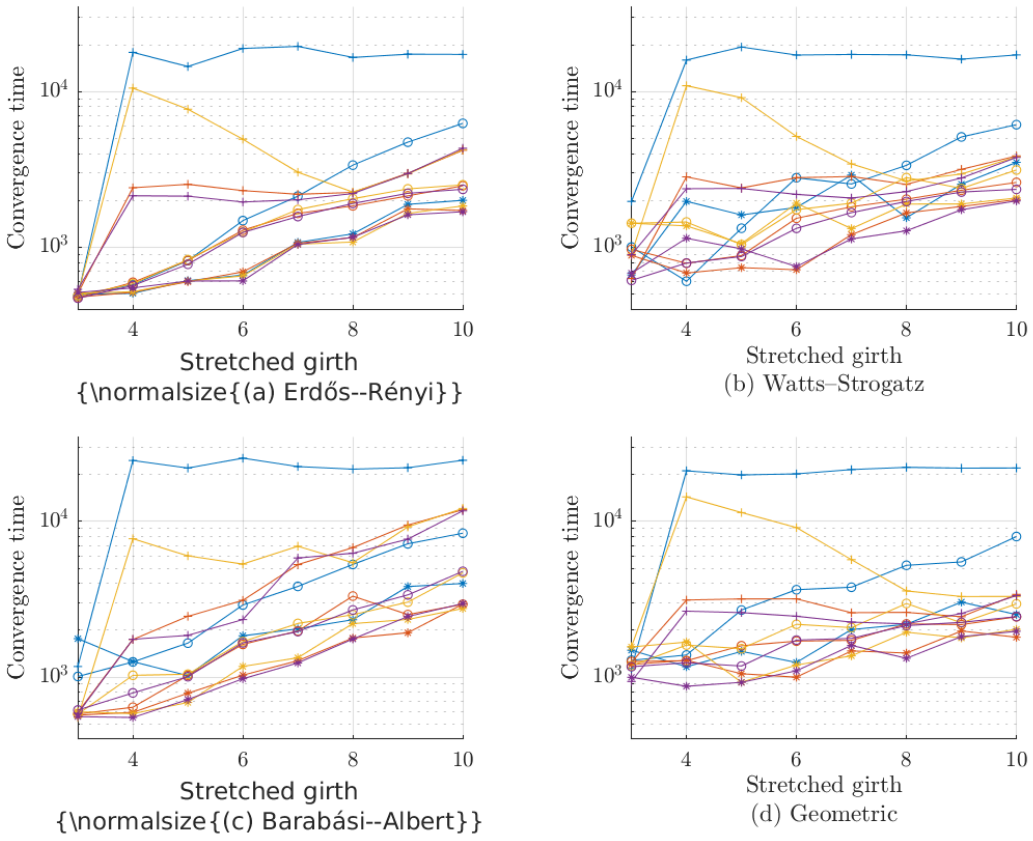


fig7_Edges-Changed-During-Optimisation.png

Comment: Figure 7 in the manuscript.

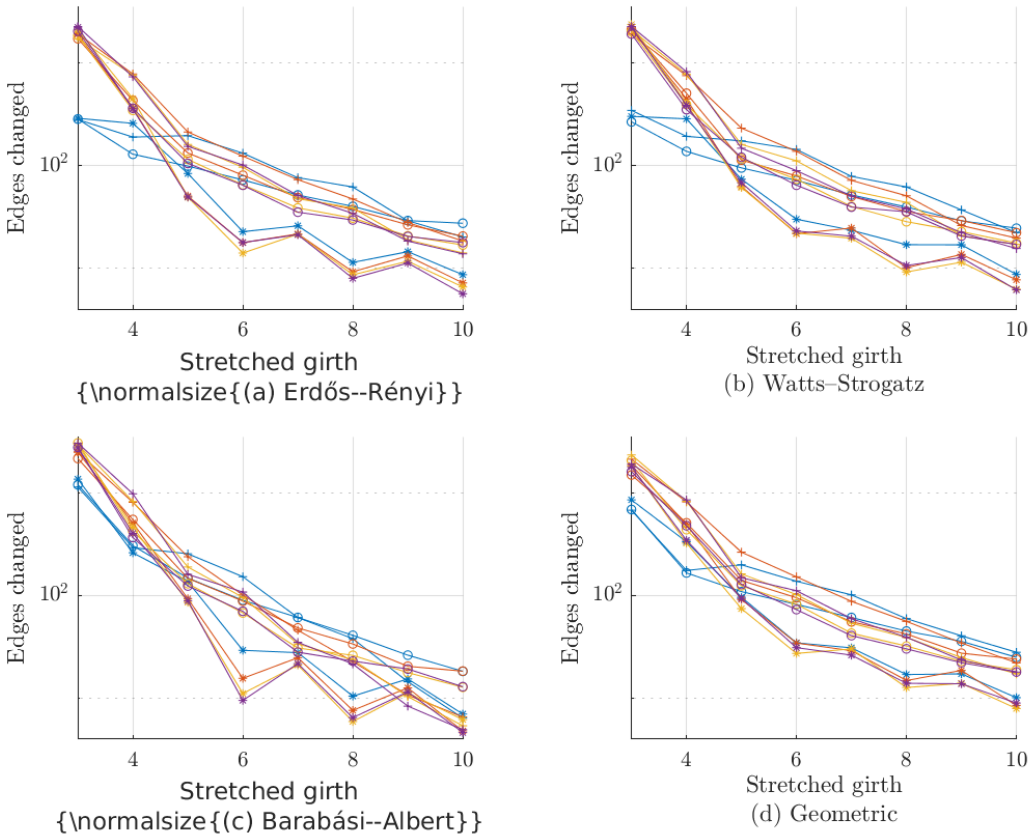
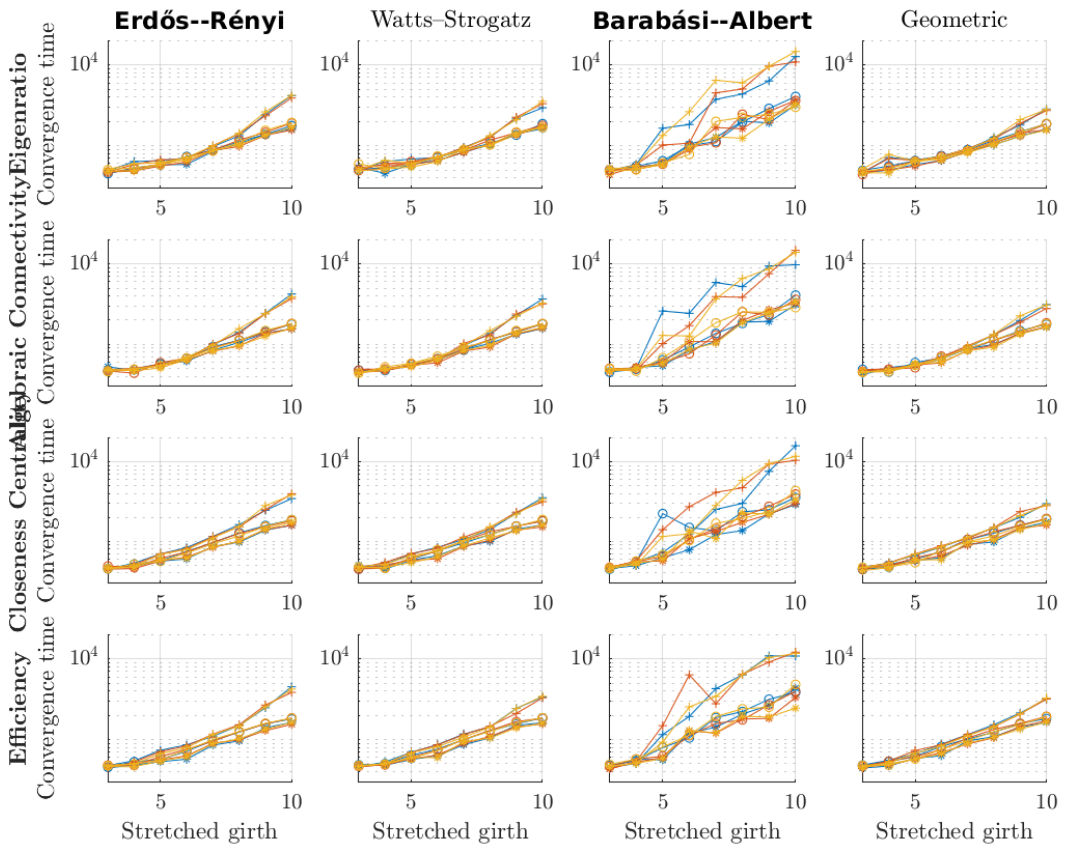


fig8_Convergence-Time-After-Optimising.png

Comment: Figure 8 in the manuscript.



Acknowledgements

This CODECHECK was done as part of the Reproducibility Check initiative led by TU Delft's Digital Competence Centre and 4TU.ResearchData.

The codecheckers would like to thank the author Florine Dekker for promptly answering any questions that came up during the CODECHECK process.

Citing this document

Aysun Urhan and Yasel Quintero (2025). CODECHECK certificate 2025-003. CODECHECK. https://doi.org/10.5281/zenodo.15333601

About CODECHECK

This certificate confirms that the codecheckers could independently reproduce the results of a computational analysis given the data and code from a third party. A CODECHECK does not check whether the original computation analysis is correct. However, as all materials required for the reproduction are freely available by following the links in this document, the reader can then study for themselves the code and data.