

# grapherator: A modular multi-step graph generator

06 December 2017

## Summary

Performance of heuristic graph algorithms for hard graph problems, e.g., Travelling-Salesperson-Problem or multi-criteria Minimum-Spanning-Tree-Problem, is typically studied by performing experiments on random graphs.

The R (R Core Team 2017) package **grapherator** implements different methods for random graph generation. The focus is on weighted graphs with one or more weights per edge. Grapherator thus targets researchers who study single- or multi-criteria optimization problems on graphs. The technical pipeline (see Figure 1) follows a three-step approach: 1) node generation (e.g., lattice, uniform etc.), 2) edge generation (Erdos-Renyi (Erdős and Rényi 1959), Waxman-model (Waxman 1988) etc.) and 3) weight generation (distance-based, random, correlated etc.). Each step may be repeated multiple times with different generator functions allowing for high flexibility (see Figure 2 for some examples). The set of predefined generator functions can be easily expanded with custom functions.

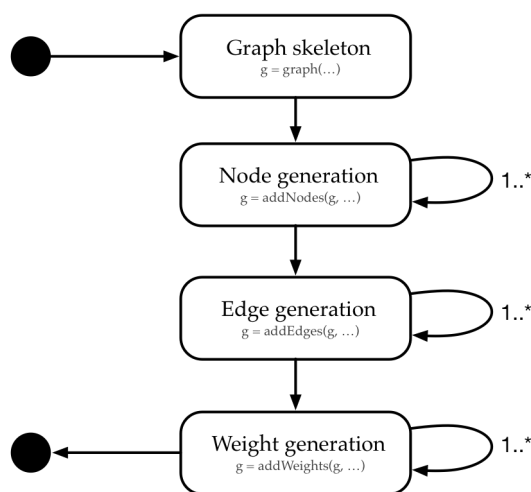


Figure 1: The grapherator workflow

## Support

Bug reports and feature requests are highly appreciated via the GitHub issue tracker (<https://github.com/jakobbossek/grapherator/issues>).

## References

- Erdős, P., and A. Rényi. 1959. “On random graphs, I.” *Publicationes Mathematicae (Debrecen)* 6: 290–97.
- R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R

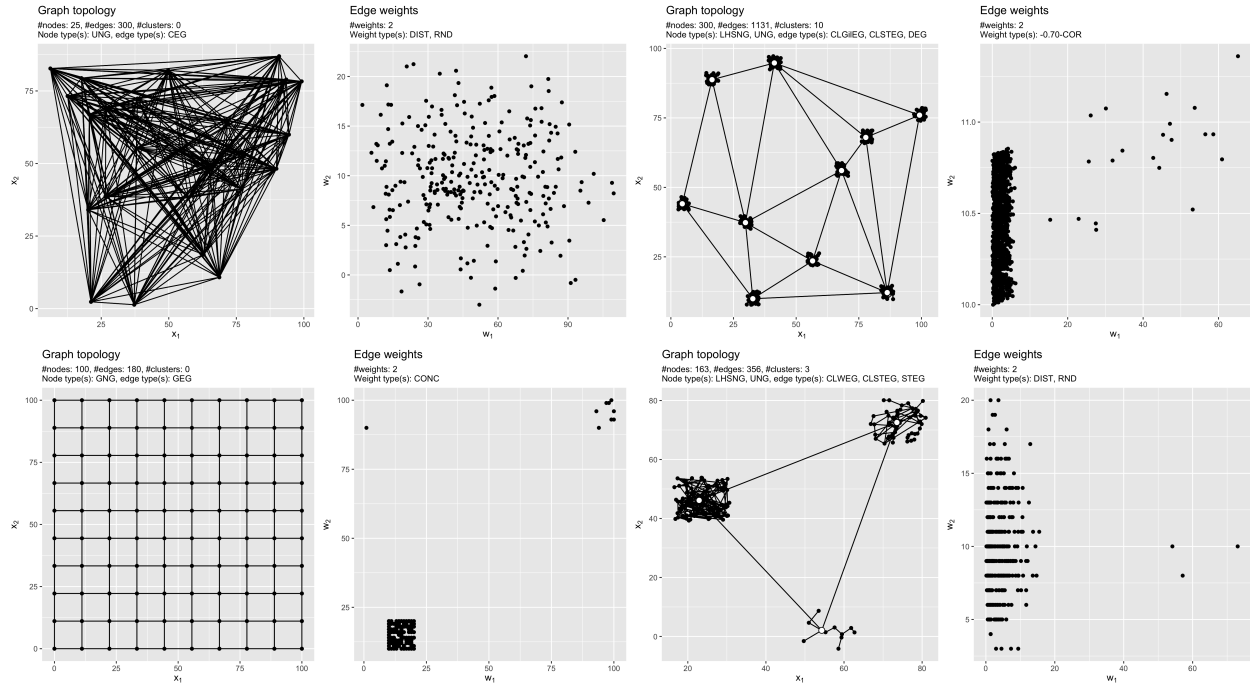


Figure 2: Example graphs with two weights per edge. Each the graph topology and a scatterplot of the edge weights is shown.

Foundation for Statistical Computing. <https://www.R-project.org/>.

Waxman, B. M. 1988. "Routing of Multipoint Connections." *IEEE Journal on Selected Areas in Communications* 6 (9): 1617–22. doi:10.1109/49.12889.