Heuristic Bandwidth Reduction Algorithms for Weighted Graphs

Luis M. B. Varona[1,2,3]

Joint Work with **Alexander Matheson**[2]

East Coast Combinatorics Conference 2026

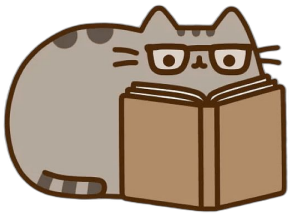
[1] *Department of Politics & International Relations, Mount Allison University*
[2] *Department of Mathematics & Computer Science, Mount Allison University*
[3] *Department of Economics, Mount Allison University*

Graph Bandwidth and Applications

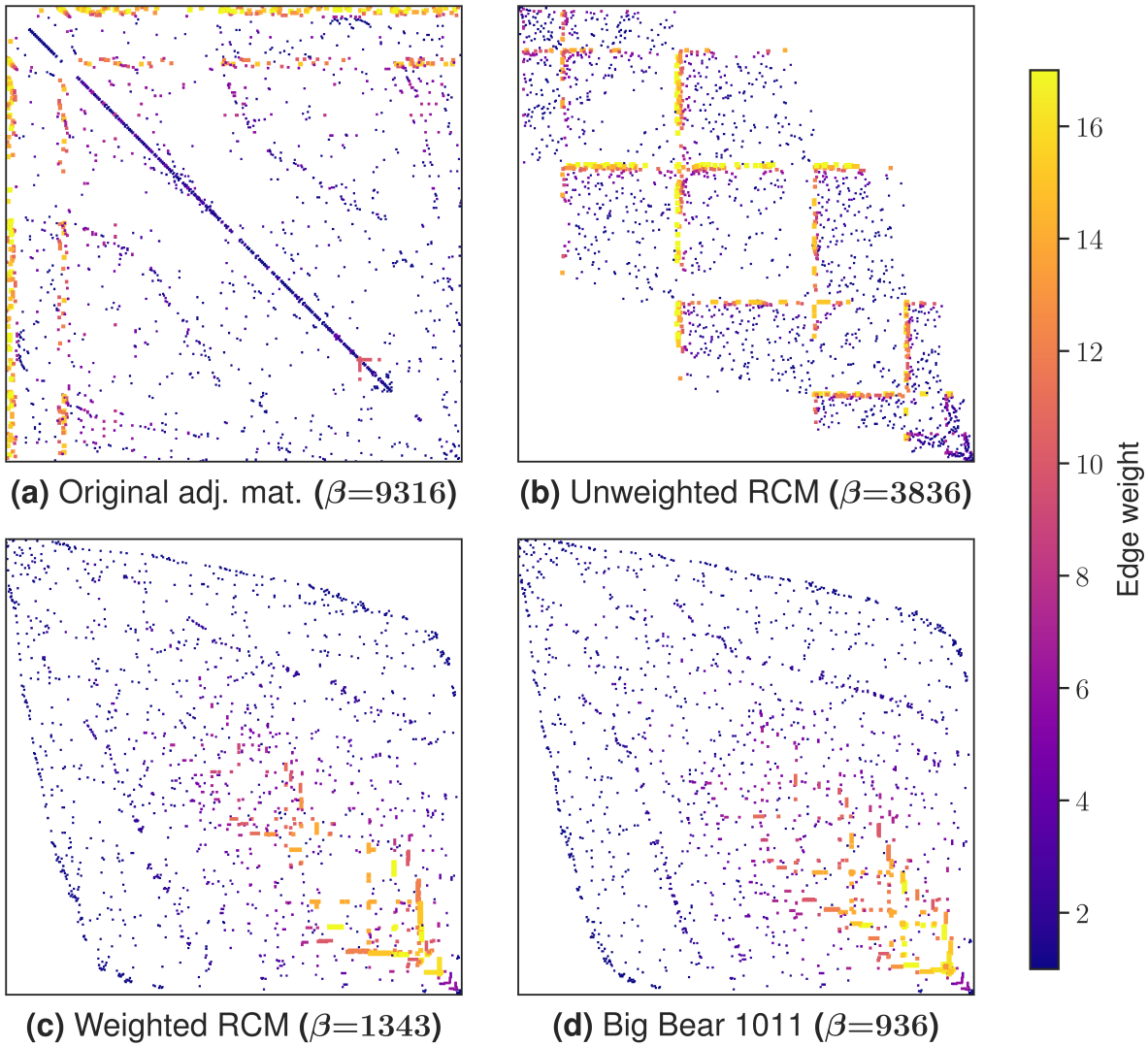
- **Layout** of G: a bijection $\pi : V(G) \to \{0, 1, \ldots, |V(G)| - 1\}$
- **Bandwidth** of G under layout π: $\beta_\pi(G) = \max_{\{u,v\} \in E(G)} |\pi(u) - \pi(v)|$
- Interested in finding **low-bandwidth layouts** because of applications in optimizing circuit layout, speeding up PDE solvers, etc.
- e.g., paths have **minimum bandwidth** $\beta(P_n) = 1$, stars have $\beta(S_n) = \lfloor n/2 \rfloor$, and complete graphs have $\beta(K_n) = n - 1$
- For a **weighted graph**: $\beta_\pi(G) = \max_{\{u,v\} \in E(G)} w(u, v) \cdot |\pi(u) - \pi(v)|$, where w is the weight function
- Most algorithms are for unweighted graphs only; these neglect, e.g., non-uniform **timing criticality** when designing circuits

Designing Algorithms for Weighted Bandwidth

- **Reverse Cuthill–McKee** (RCM) and other heuristics *can* be trivially adapted to weighted graphs, but this is rather suboptimal
- My (very early-stages) **"Big Bear 1011"** algorithm:
  1. Takes a linear forest with high-weight edges, then contracts all paths (computing the weights of the new edges is nontrivial)
  2. Runs both weighted RCM and a custom RCM where all edges below a certain weight threshold are (temporarily) deleted
  3. Re-expands the super-nodes and returns the better layout
- Rmk.: Custom RCM wins on "heavy-backbone" graphs (e.g., **VLSI circuits** or **caterpillars**); weighted wins on fairly uniform graphs



Observe: Big Bear in Action!



(a) Original adj. mat. (β=9316)

(b) Unweighted RCM (β=3836)

(c) Weighted RCM (β=1343)

(d) Big Bear 1011 (β=936)

Figure: "Big Bear 1011" reduces β by >30% more than weighted RCM on a 561-**cell circuit** benchmark (s1196, ISCAS 1989). [Code: github.com/Luis-Varona/eccc-2026-weighted-bw.]