Parallel Coarsening of Graph Data with Spectral Guarantees

Christopher Brissette[1] Andy Huang[2] George M. Slota[1]

[1]Rensselaer Polytechnic Institute
[2]Sandia National Laboratories

SIAM Data Mining
Applications of Topological Data Analysis Workshop

25 February 2022

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND2022-5199 C.

Spectrum of a graph

An important object of study in graph theory and specifically spectral graph theory is the graph Laplacian.

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

- This is analogous to the Laplacian from physics.
- Can determine certain topological and geometric features from its spectrum.
  - The number of 0 eigenvalues equals the number of connected components.
  - The first eigenvector can be used to approximate the Cheeger-cut of the graph. (Chung. 1997)

Spectral partitioning

We would like to be able to partition a given graph $G = (V, E)$ into equally weighted portions. A powerful tool for this is spectral partitioning as it provides us with useful global information and guarantees on cut quality.

- Requires $O(n^3)$ operations in the dense case.
- Requires $O(m)$ operations in the sparse case, with limited room for parallelism, and possible convergence issues.

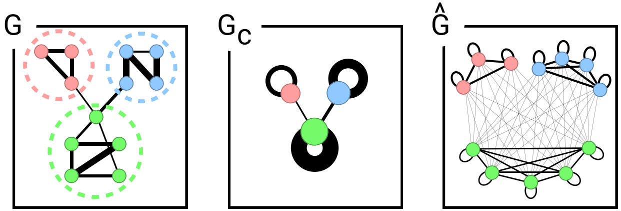
Is there a way to coarsen a graph while closely preserving this information?
Yes! However, they tend to be slow!
(Lescoat et al. 2020, Loukas 2019, Jin. Loukas 2020, Liu et al. 2019)

The lift

Note: Comparing spectra of G and G_c is difficult!

- Eigenvectors are of different dimensions
- There are different numbers of eigenvalues. How do we align them?

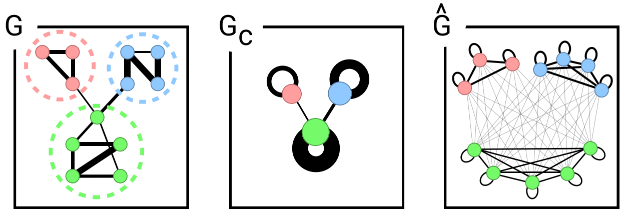
For this purpose we note the **lift** of the graph.



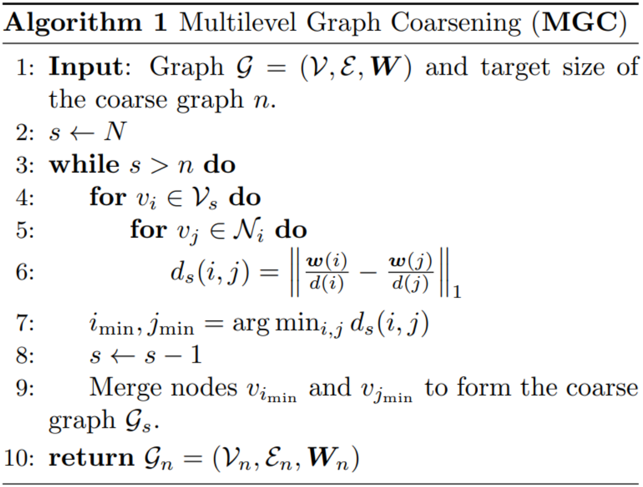
A spectral bound

A useful result (Jin. Loukas 2020) states the following...
If Λ and $\hat{\Lambda}$ are sorted vectors containing the eigenvalues of L and $\hat{L}$ respectively. Then if we merge two nodes u, v 2 V such that $k\frac{w_u}{d_u} \quad \frac{w_v}{d_v}k_1 \quad \epsilon$, we have,

$$k\Lambda \quad \hat{\Lambda}k_1 \quad \epsilon$$



Explicit greedy method

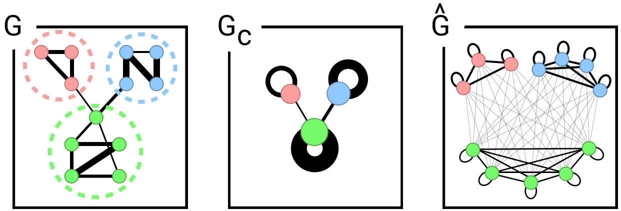


Algorithm 1 Multilevel Graph Coarsening (**MGC**)

```
1: Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ and target size of
   the coarse graph $n$.
2: $s \leftarrow N$
3: while $s > n$ do
4:     for $v_i \in \mathcal{V}_s$ do
5:         for $v_j \in \mathcal{N}_i$ do
6:             $d_s(i, j) = \left\| \frac{\boldsymbol{w}(i)}{d(i)} - \frac{\boldsymbol{w}(j)}{d(j)} \right\|_1$
7:     $i_{\min}, j_{\min} = \arg\min_{i,j} d_s(i, j)$
8:     $s \leftarrow s - 1$
9:     Merge nodes $v_{i_{\min}}$ and $v_{j_{\min}}$ to form the coarse
   graph $\mathcal{G}_s$.
10: return $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n, \boldsymbol{W}_n)$
```

It's too slow!

Their algorithm requires $O(n^2 m)$ work with dense vector operations, or $O(n^2 \langle d^2 \rangle)$ work with sparse vectors.
We propose a heuristic to reduce this to $O(n \langle d^2 \rangle)$ work where $\langle d^2 \rangle$ is the second moment of the degree distribution. Furthermore our method is parallelizable and can be performed in $O(\frac{n}{p} \langle d^2 \rangle)$ parallel time.



An observation

We can bound our spectral difference purely based on the norm differences of the uncoarsened graph.

For s levels of coarsening where each merge has $k\frac{w_u}{d_u} \quad \frac{w_v}{d_v}k_1 \quad \epsilon$ in the original graph we know the following is true of the spectrum.

$$k\Lambda \quad \hat{\Lambda}k_1 \quad \frac{s(s+1)}{2}\epsilon$$

- This essentially comes from an iterative application of the triangle inequality
- For more details, see the proof in the workshop paper submission.

An observation

This observation allows us to never have to recompute norm-differences!

I We can obtain a spectral approximation of our graph using only the norm-differences $k\frac{w_u}{d_u} \qquad \frac{w_v}{d_v}k_1$ in the original graph.

I This avoids the outer-loop in their explicit greedy method, and reduces the complexity by a factor of n, giving us a $O(n\left\langle d^2\right\rangle)$ work algorithm.

Our algorithm

```
Input: G = ( V, E ), r, p
Result: coarsened graph $G_c$
G$_c$    $G$
merge_fitness    $;$
$f$E$_1$,    $, E_p g$    $edgePartition(E, p)$
for $i$=1 to $p$ in parallel do
    for $j$=1 to $jE_i j$ do
        u    $E_i[j][1]$
        v    $E_i[j][2]$
        merge_fitness    $merge\_fitness$ [ $(k\frac{w_u}{d_u}$    $\frac{w}{d_v}k_1, E_i[j])$
    end
end
ascendingSort( merge_fitness )
for  $i$=1 to $jVj$    $r$ do
    G$_c$    $merge(G_c, merge\_fitness[i][0], merge\_fitness[i][2])$
end
return: G$_c$
```

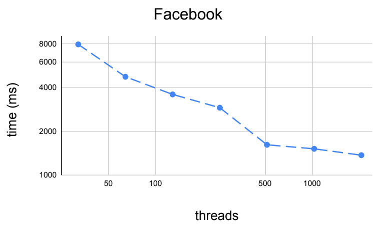
Our algorithm

As mentioned, out algorithm without parallelization has serial time $O(n \left\langle d^2 \right\rangle)$, and with parallelization the edge-weight calculations are reduced to $O(\frac{n}{p} \left\langle d^2 \right\rangle)$.

- We then perform a sort that requires O(mlog(m)) work. Can be parallelized to $O(\frac{m}{p} log(m))$ time.
- The merges must be done sequentially, and require O($n \quad r$) work.

Overall for this leaves us with an $O(\frac{n}{p} \left\langle d^2 \right\rangle + \frac{m}{p} log(m) + (n \quad r))$ parallel time algorithm.

Timing Results



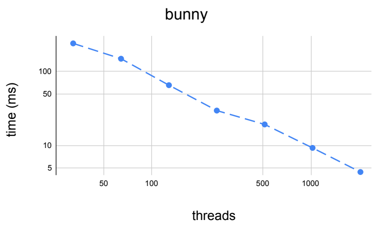
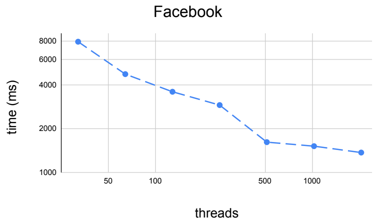
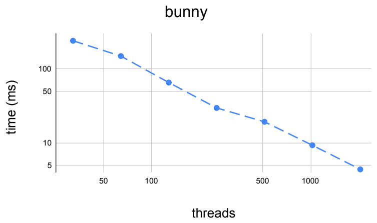


Figure: Scaling results for the Facebook and Standford bunny graphs from 32 to 2048 threads.

Timing Results





Note that, while the Bunny graph achieves proper parallel scaling, the Facebook graph does not. This is due to degree irregularity.

- In our current algorithm we do not parallelize within each norm computation.
- This means some threads have to compute far more expensive sparse norms than others. This leads to dead time for some threads while others are still computing many norms.

Approximation Results

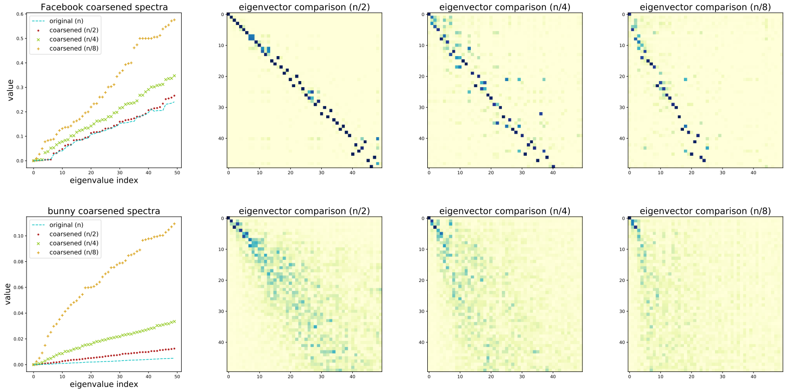


Figure: Comparisons of spectral properties between the Facebook graph, and the Stanford bunny at three levels of coarsening.

Eigenvectors

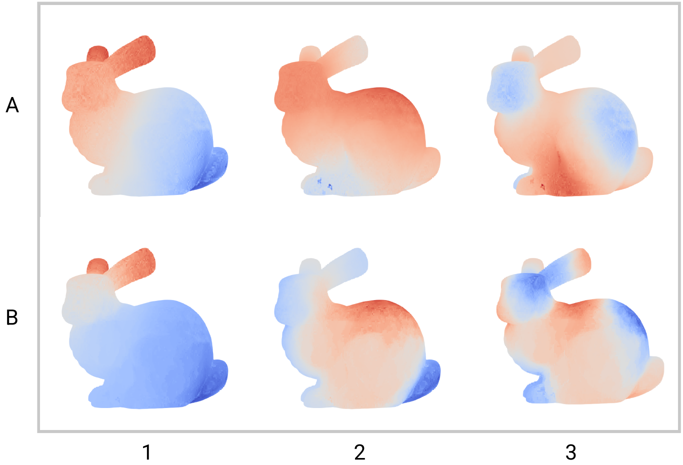


Figure: Eigenvector comparisons between uncoarsened (**A**) and half coarsened (**B**) bunny meshes. We compare the first, fifth, and tenth eigenvectors in columns 1,2, and 3 respectively.

Future Work

- As mentioned, the merges currently have to be done serially. However, this isn't necesarily true if a small adjustment is made. If we have disjoint merges they can be parallelized. Therefore, by weighting edges inversely to their fitness function and performing a maximal matching we can further speed up an approximation of our algorithm via parallel merges.
- This method relies on storing several arrays in GPU memory. This quickly leads to spatial constraints. Large graphs will require distributing the structure over multiple GPUs.
- We currently only compute the norms with respect to the original graph, but by recomputing norms after some level of coarsening we should closer approximate the explicit greedy algorithm and tighten our errors.

Acknowledgements

- This work was supported by the U.S. Department of Energy, Office of Science, under award DE-AC52-07NA27344 (FASTMath SciDAC Institute).
- This work was also supported under Laboratory Directed Research and Development (LDRD) program, Project No. 218474, at Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.
- This presentation describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Rensselaer

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

FASTMATH

Summary and Thanks

- We would like to be able to coarsen graphs while also preserving information via the Laplacian spectrum.
- Current methods for doing this are rather slow.
- We present a highly parallel method for performing this task.
- The speed up of our method is highly contingent of the degree irregularity of the graph.

Thank you!

Contact: brissc@rpi.edu