

Eric Buras Thesis: Methodology

March 14, 2016

1 Graph Partitioning

In studying 'small world' networks proposed by Watts and Strogatz [7] Fan Chung and Linyuan Lu propose an algorithm that separates a graph into a locally connected component and a global component. Finding a local subgraph is akin to finding a locally connected portion of a graph. Given integers $k \geq 2$ and $l \geq 2$, a (k, l) locally connected graph will have at least l paths connecting any given two nodes with distinct edges in each path. The length of each path can be at most k edges for this pair. A grid network can be described locally with $k = 3, l = 3$, and $k = 5, l = 9$ and is a good example of how connected these types of graphs are. Given graph G , its maximum locally connected subgraph is the union of all locally connected subgraphs within the entire graph. It is important that this maximum is unique, and can be found through edge deletion [3]. Thus we are able to split a graph into two unique components.

(Picture of grid graph with random edges and graph split)

2 Laplacian Solver

I have partitioned a graph into its locally connected subgraph, P , and the graph of the teleportation edges, T . These subgraphs can be converted to Laplacian form with P_L matrix of connections and degree information for the locally connected part, and T_L , similar for the teleportation part. With a minor diagonal operation on T_L , we have $L = P_L + T_L$ where L is the Laplacian for the entire graph. To solve a Laplacian linear system $Lx = b$ we now solve the two subgraph Laplacians and use linear algebra.

3 Algebraic Multigrid of Locally Connected Subgraph

The locally connected subgraph is planar, meaning it can be drawn on a piece of paper without any edges crossing (proof it is planar?). Using work beginning

with Gary Miller [6], we know that an Algebraic Multigrid solver is optimal for this planar graph laplacian matrix as the solution space is split into multiple cycles of coarsened solving [2]. This solves $P_L y = b$ where b is the right hand side of the original linear system. Previous approaches to using multigrid to solve Laplacian linear systems do not have a systematic method of graph coarsening. Thus they are prone to losing edge information in the multiple coarsening cycles. (do i need to cite?)

4 Low Rank SVD of Teleportation Subgraph

For graphs I am interested in, the number of edges in the Teleportation subgraph is very small relative to the size of the original graph. This causes the laplacian matrix, T_L , to have very low rank structure. I can then take the low rank singular value decomposition (cite?) to solve this small portion. This part of the algorithm takes negligible time relative to the multigrid cycles on the planar portion.

5 Linear Algebra: Woodbury Matrix Identity

To combine the SVD and Multigrid solves, I utilize the Woodbury matrix Identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

with A replaced by P_L , U and V component matrices of the SVD of T_L , and C replaced by a diagonal of the singular values of T_L .

6 Summary of Solving the Laplacian Linear System

$$\begin{aligned} Lx &= b \\ x &= L^{-1}b \\ x &= (P_L + T_L)^{-1}b \\ x &= (P_L + USV)^{-1}b \\ &\text{Use Woodbury matrix identity} \\ x &= (P_L^{-1} - P_L^{-1}U(S^{-1} + VP_L^{-1}U)^{-1}VP_L^{-1})b \\ x &= P_L^{-1}b - P_L^{-1}U(S^{-1} + VP_L^{-1}U)^{-1}VP_L^{-1}b \end{aligned}$$

7 Complexity of Graph partitioning and Linear System Solve

NEED THIS

8 NetworkX and PETSc

NetworkX is a valuable library for graph theory and analysis written in python. This library has many functions for working with data, and specifically use the `kl_connected_subgraph` function to partition the graph into the locally connected subgraph and the teleportation subgraph. [5] This function operates according to Chung and Lu's greedy algorithm. Currently this function operates in serially over the graph edges, and I believe there is ample opportunity to speed it up.

The Portable, Extensible Toolkit for Scientific Computation (PETSc) is primarily written in C for solving partial differential equations, however there is a library for python users called `petsc4py` [4]. I utilize the algebraic multigrid solvers in PETSc for the locally connected subgraph solves as mentioned above [1].

References

- [1] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.6, Argonne National Laboratory, 2015.
- [2] Achi Brandt and Oren E Livne. *Multigrid techniques: 1984 guide with applications to fluid dynamics*, volume 67. SIAM, 2011.
- [3] Fan Chung and Linyuan Lu. *Complex Networks*, chapter The Small World Phenomenon in Hybrid Power Law Graphs, pages 89–104. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [4] Lisandro D Dalcin, Rodrigo R Paz, Pablo A Kler, and Alejandro Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.
- [5] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.
- [6] Gary L Miller and Joseph Naor. Flow in planar graphs with multiple sources and sinks. *SIAM Journal on Computing*, 24(5):1002–1017, 1995.
- [7] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.