Distributed Laplacian Solver

Rahul V

September 13, 2019

Problem Statement

- Solve $Lx = b$ where $L = D - A$.
- Properties of L:
  1. Symmetric, diagonally dominant
  2. Singular, Positive semi-definite
  3. Sum of each row is 0
- One sink: $b_n = -\sum_{i=1}^{n-1} b_i$

Approach: Random walk

- Solve Data Collection Problem (DCP)
  - Each node generates a packet with probability βJ_u
  - Random packet from queue and a random neighbor v with probability directly propotional to w_{uv}
  - The packet is then transmitted to this neighbor
  - Packets sunk at the sink immediately
- At stationarity, $\eta^T(I - P) = \beta J^T$ or $(\eta^T D^{-1})L = \beta J^T$ holds
- Ergodic iff $\beta < \beta^*$, and we have a lower limit for β^*. So we binary search on β.
- Scale and shift x for canonical solution, i.e., $< x, 1 >= 0$

Status quo: Practical concerns

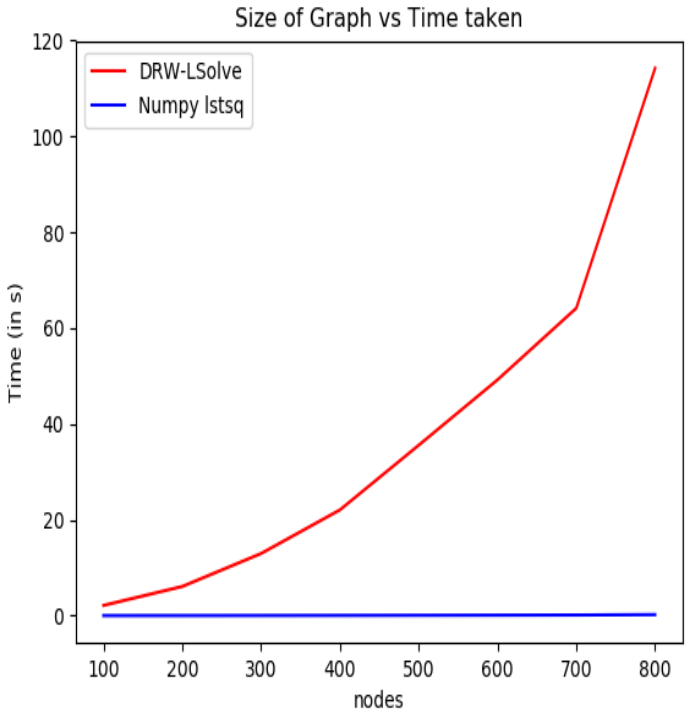
Testing

- Guarantee by the algorithm:
  $|x_u - \hat{x_u}| < (\epsilon_1 + \epsilon_2)x_u$, whenever $\kappa < x_u d_u$
- Test set construction:
  - $m =$ random from $[n-1, n(n-1)/2]$
  - Generate Random MST with random weights in $[0, 100]$
  - Generate random edges by selecting 2 random nodes
  - b_i is chosen randomly from $[0, 1000]$

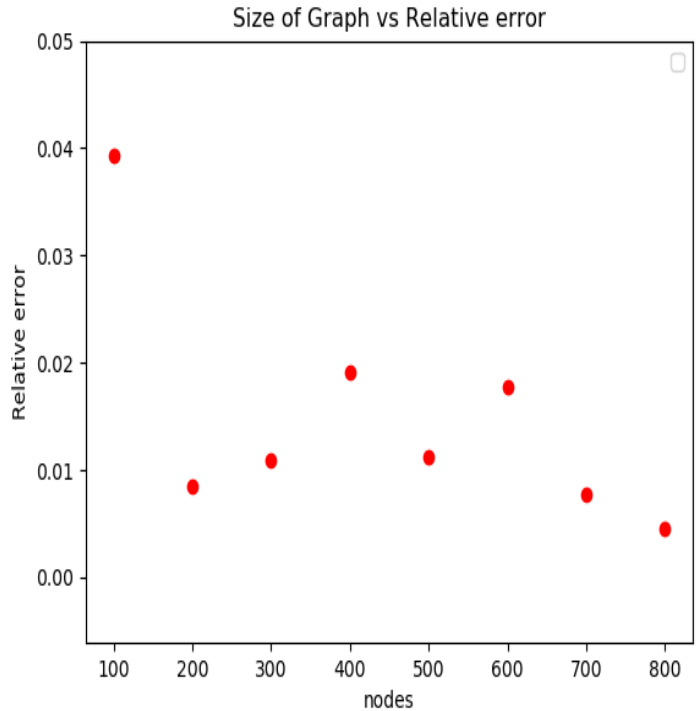
Status quo: Practical concerns

Performance

- $(\kappa, \epsilon_1, \epsilon_2) = (0.1, 0.1, 0.1)$
- Shared memory model with 4 threads
- Fixed number of rounds ($O(nlogn)$), sampling from the start



(a) Performance



(b) Accuracy

Near future

- Use Message Passing instead of Shared memory model
- Graph partitioning to reduce communications, who should be at the boundary:
  - Work as long as possible without need for checking queue
  - Get very few packets in the first place

Thank you