Parallel graph decompositions using random shifts

Gary L. Miller, Richard Peng, Shen Chen Xu

Presenter: Jessica Shi

6.886 Algorithm Engineering
Spring 2019, MIT
Introduction
Graph decomposition

- **Graph decomposition**: Partition vertices of a graph such that:
  - Subsets satisfy some connectivity property
  - There are few edges between subsets
- **Diameter**: Maximum length of a shortest path between any two vertices
- **Low diameter graph decomposition**
Motivation

- Key subroutine in many (distributed) algorithms:
  - Low-stretch embedding of graphs into trees [1]
  - Shortest path approximations [2]
  - Symmetric diagonally dominant (SDD) linear system solvers [3]
    - Applications: Max flow, negative-length shortest path [4]
    - Issue: Polylog ($\log^O(1) n$) work factor b/c of low diameter decomposition alg to generate tree embeddings

---

Aside: Diameter

- **Strong diameter**: Diameter of the induced subgraph on the subset $S$
- **Weak diameter**: Diameter of the subset $S$ where shortest paths may use vertices outside of $S$
  - Quadratic work factor for parallel low diameter decompositions $[5]$
- **Note**: Take “diameter” to mean “strong diameter”

![Figure: The strong diameter of the blue vertices is 3, but the weak diameter is 2.](image)

Main results
Main results

Main problem

- A \((\beta, d)\) decomposition is a partition of \(V\) into subsets \(S_i\) such that
  - Each \(S_i\) has diameter \(\leq d\)
  - Number of edges between subsets \(\leq \beta m\).
- **Note:** Usually (optimally), \(d = O\left(\log \frac{n}{\beta}\right)\)
Main results

Related work

- **Sequential**: $(\beta, O(\log n/\beta))$ decomposition:
  - $O(m)$ time
- **Previous** [6]: $(\beta, O(\log^4 n/\beta))$ decomposition:
  - Expected $O(\log^3 n/\beta)$ depth, $O(m \log^2 n)$ work
- **This work**: $(\beta, O(\log n/\beta))$ decomposition $(\beta \leq 1/2)$:
  - Expected $O(\log^2 n/\beta)$ depth, $O(m)$ work
  - Work-efficient!

---

Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm ($\beta = 1/2$)
Sequential (ball-growing) algorithm \((\beta = 1/2)\)
Sequential algorithm (overview)

- Choose a vertex $v$ and start a subset ("ball") $S_v = \{v\}$
- Use BFS to add vertices to $S_v$
- Stop when ($\#$ edges on the boundary of $S_v) < \beta \cdot (\#$ of edges in $S_v$)
- Delete all vertices in $S_v$
- Repeat until all vertices have been deleted (partitioned)
Sequential algorithm (crossing edge analysis)

- All subsets $S_v$ satisfy $(\# \text{ edges on the boundary of } S_v) < \beta \cdot (\# \text{ of edges in } S_v)$ upon creation
- $\therefore$ only $\beta m$ edges total cross subsets
Sequential algorithm (diameter analysis)

- Let $i$ denote BFS iterations
- Let $m_i$ denote # edges in $S_v$ after step $i$
- At step $i$:
  - Increase diameter by at most 2
  - Must have added all vertices from step $(i - 1)$:
    - # edges on frontier at start of step $(i - 1) \geq \beta m_{i-2}$
    - $m_{i-1} \geq (1 + \beta) \cdot m_{i-2}$
- Since diameter increases at most linearly with $i$, the diameter of a subset is bounded by $O\left(\log n / \beta\right)$
**Main results**

**Blelloch et al.’s algorithm (sketch)**

- Randomly sample a subset of vertices to be “centers”
- Grow balls starting from the centers in parallel
- If two balls overlap, choose which ball to place overlapping vertices based off of distance to center (with an additive random shift factor)
- Repeat until all vertices have been partitioned
This algorithm

- Each \( u \in V \) picks \( \delta_u \) indep. from an exp. distr. w/mean \( 1/\beta \)
- Let \( \delta_{\text{max}} \) denote the max \( \delta_u \)
- Start an instance of parallel BFS, with \( v \) s.t. \( \delta_{\text{max}} = \delta_v \)
- When the vertex at the head of the queue has dist > \( \delta_{\text{max}} - \delta_u \), start parallel BFS with \( u \) (add to queue) if it has not yet been visited (as a center)
- Assign each vertex \( u \) to the center that visited it in the BFS
- Note: Think of \( \delta_u \) as randomized start times for \( u \) to begin its own ball
Preliminaries
Simplification: Take diameter to be the max distance from a designated center $u$ of subset $S_u$ to any $v \in S_u$

- Bounds diameter up to factor of 2

**Shifted distance**: Define $\text{dist}_{\delta}(u, v) = \text{dist}(u, v) - \delta_u$
Preliminaries

- **Exponential distribution:**
  
  - **PDF:** \( \text{Exp}(\gamma) = f(x, \gamma) = \begin{cases} 
  \gamma e^{-\gamma x} & \text{for } x \geq 0, \\
  0 & \text{otherwise}
  \end{cases} \)
  
  - **CDF:** \( F(x, \gamma) = \Pr[\text{Exp}(\gamma) \leq x] = \begin{cases} 
  1 - e^{-\gamma x} & \text{for } x \geq 0, \\
  0 & \text{otherwise}
  \end{cases} \)
  
  - **Mean:** \( 1/\gamma \)
  
- **\( i^{th} \) order statistic** of RV \( \{X_i\}_{i\in[n]} \): \( X_{(i)}^n \) = value of \( i^{th} \) smallest \( X_{(1)}^n \) and consecutive differences \( X_{(k+1)}^n - X_{(k)}^n \) are indep.
  
  - **PDF of** \( X_{(1)}^n \): \( \text{Exp}(n\gamma) \)
  
  - **PDF of** \( X_{(k+1)}^n - X_{(k)}^n \): \( \text{Exp}((n - k)\gamma) \)
Analysis (correctness)
This algorithm

- Each $u \in V$ picks $\delta_u$ indep. from an exp. distr. w/mean $1/\beta$
- Let $\delta_{\text{max}}$ denote the max $\delta_u$
- Start an instance of parallel BFS, with $v$ s.t. $\delta_{\text{max}} = \delta_v$
- When the vertex at the head of the queue has dist $> \delta_{\text{max}} - \delta_u$, start parallel BFS with $u$ (add to queue) if it has not yet been visited (as a center)
- Assign each vertex $u$ to the center that visited it in the BFS
- **Note**: Think of $\delta_u$ as randomized start times for $u$ to begin its own ball
**Modified algorithm**

- Each $u \in V$ picks $\delta_u$ indep. from $\text{Exp}(\beta)$
- Assign each vertex $v$ to $S_u$ where $u$ minimizes $\text{dist}_\delta(u, v)$ (break ties lexicographically)
- These form the partitions $S_u$
Lemma

If \( v \in S_u \) and \( v' \) is the last vertex on the shortest path from \( u \) to \( v \), then \( v' \in S_u \) as well.

Proof.

Assume \( v' \in S_{u'} \):

- **Shortest path:** \( \text{dist}_{-\delta}(u, v) = \text{dist}_{-\delta}(u, v') + 1 \)
- **Adjacent:** \( \text{dist}_{-\delta}(u', v) \leq \text{dist}_{-\delta}(u', v') + 1 \)
- **Cases:**
  - \( v' \) closer to \( u' \) than to \( u \) \( \Rightarrow \) \( v \) is closer to \( u' \) than to \( u \), so \( v \in S_{u'} \)
  - \( v' \) is the same distance from \( u' \) and \( u \), but \( u' \) is lexicographically before \( u \) \( \Rightarrow \) \( v \) is the same distance from \( u' \) and \( u \), so \( v \in S_{u'} \)
Modified algorithm (diameter analysis)

- **Note:** Since we may have $v \in S_v$, diameter is bounded above by $\delta_{\text{max}} = \max_u \delta_u$

**Lemma**

The expected value of the max shift is $H_n/\beta$, where $H_n$ is the $n^{th}$ harmonic number. With high probability (by failure parameter $d$), $\delta_u \leq O(\log n/\beta)$.

**Proof.**

- Expected value of max shift: Sum over differences of order statistics:
  - $E[\delta_{\text{max}}] = E[\delta^n_{(n)}] = \frac{1}{\beta} \sum_{i=1}^n \frac{1}{i} = H_n/\beta$
- Bound all $\delta_u$: Use CDF and union bound:
  - $Pr[\delta_u \geq (d + 1) \cdot \ln n/\beta] \leq n^{-(d+1)}$
Modified algorithm (crossing edge analysis)

Lemma

Let edge \((u, v)\) have midpoint \(w\). If \(u \in S_{u'}\) and \(v \in S_{v'}\) \((u' \neq v')\), then \(\text{dist}_{-\delta}(u', w)\) and \(\text{dist}_{-\delta}(v', w)\) are within 1 of the min shifted distance to \(w\).

Proof.

- Let the arg min shifted distance to \(w\) be \(w'\)
- Since \(w\) to \(u\) is \(1/2\), \(\text{dist}_{-\delta}(w', u) \leq \text{dist}_{-\delta}(w', w) + 1/2\)
- If \(\text{dist}_{-\delta}(u', w) > \text{dist}_{-\delta}(w', w) + 1\),
  \[
  \text{dist}_{-\delta}(u', u) \geq \text{dist}_{-\delta}(u', w) - 1/2 \\
  > \text{dist}_{-\delta}(w', w) + 1/2 \quad \text{(substitute)} \\
  \geq \text{dist}_{-\delta}(w', u),
  \]
  but \(u'\) minimizes shifted dist to \(u\)
Modified algorithm (crossing edge analysis)

Main idea: For every edge \((u, v)\):

- Consider all shifted distances to midpoint \(w\)
- If the min + second min of these aren’t within 1 of each other, then \(u\) and \(v\) must be in same subset
- Bound the probability \(p\) that min + second min are within 1 of each other

\[\therefore pm\text{ is expected number of edges across subsets}\]

- Represent shifted distances as \(d_i - \delta_i\), where \(d_i\) is arbitrary and \(\delta_i\) is from \(\text{Exp}(\beta)\)
Modified algorithm (crossing edge analysis)

Proof sketch:

- $d_i$ indicates when a light bulb is turned on (time goes from high to low), $\delta_i$ is lifespan
- $\min(d_i - \delta_i) =$ time when last light burns out
- Want to bound diff $\Delta$ b/w when last light burns out + second last light burns out
- Exp distr is memoryless $\Rightarrow$ last light follows exp distr after second last light burns out
- $Pr[\Delta < c]$ is bounded by CDF $1 - e^{-c\beta} \approx c\beta$ (for small $c\beta$)
- Case: If last light not on yet when second last light dies, $Pr[\Delta < c]$ can only be less than the above
Modified algorithm (crossing edge analysis)

Lemma

$$Pr[\Delta \leq c]$$ is at most $$O(\beta c)$$.  

Proof.

- More convenient to consider $$-(d_i - \delta_i)$$ ⇒ let $$d'_i = -d_i$$
- Let $$X_i = d'_i + \delta_i - d'_1$$, let $$X(i)$$ be $$i^{th}$$ order stat of $$X_j$$
  - Note: $$X_i$$ follows exp distr w/mean $$1/\beta$$
- WTS: $$Pr[X(n) - X(n-1) > c] \geq e^{-\beta c}$$
- For $$S \subseteq [n]$$, let $$\varepsilon_S$$ be the event where $$X_i \geq 0$$ iff $$i \in S$$
- $$Pr[X(n) - X(n-1) > c] = \sum_S Pr[X(n) - X(n-1) > c | \varepsilon_S] Pr[\varepsilon_S]$$
Modified algorithm (crossing edge analysis)

Proof.

- Since $X_1 = \delta_1 \geq 0$, if $1 \notin S$, then $Pr[\varepsilon_S] = 0$

- Case: $|S| = 1$: $S = \{1\}$:
  - $Pr[X_1 > c] \geq e^{-\beta c}$
  - Since $X_n \geq X_1$ and $X_{n-1} < 0$, we have
    \[
    Pr[X_n - X_{n-1} > c|\varepsilon_S] \geq e^{-\beta c}
    \]

- Case: $|S| \geq 2$:
  - By order statistics, $Pr[X_n - X_{n-1} > c|\varepsilon_S] \geq e^{-\beta c}$

In total:

\[
Pr[X_n - X_{n-1} > c] \geq e^{-\beta c} \Rightarrow Pr[\Delta < c] \leq 1 - e^{-\beta c} < \beta c
\]
Analysis (work/depth)
This algorithm

- Each \( u \in V \) picks \( \delta_u \) indep. from an exp. distr. w/mean \( 1/\beta \)
- Let \( \delta_{\text{max}} \) denote the max \( \delta_u \)
- Start an instance of parallel BFS, with \( v \) s.t. \( \delta_{\text{max}} = \delta_v \)
- When the vertex at the head of the queue has dist > \( \delta_{\text{max}} - \delta_u \), start parallel BFS with \( u \) (add to queue) if it has not yet been visited (as a center)
- Assign each vertex \( u \) to the center that visited it in the BFS
- **Note:** Think of \( \delta_u \) as randomized start times for \( u \) to begin its own ball
Implementation improvements

- Simulate \(-\delta_u\) shifts by using super source \(s\) with dist \(-\delta_u\) to each \(u\)
- Fix negative edge lengths by adding \(\delta_{\text{max}}\)
- Only non-integral path lengths are from \(s\)
  - Use fractional parts from \(s\) as tie-breakers
  - Can also replace these with a random permutation
- Delayed processing of edges so can use unweighted BFS
Work/depth analysis

- Generating $\delta_u$: $O(1)$ depth and $O(n)$ work
- BFS: $O(\Delta \log n)$ depth and $O(m)$ work (where $\Delta$ is max distance) [7]
  - Each center to vert in subset has max distance $O(\log n/\beta)$
  - In total: $O(\log^2 n/\beta)$ depth and $O(m)$ work
- Verify correctness: $O(\log n)$ depth and $O(m)$ work
- In total: $O(\log^2 n/\beta)$ depth and $O(m)$ work

Conclusion
Future work

- Actual implementation?
- Weighted low diameter decomposition
  - Difficult to bound depth
- Other kinds of decompositions, e.g., low weak diameter block decomposition
  - $O(\log^2 n)$ depth and $O(n \log^2 n)$ work for $(\log n, \log n)$ decom [8]

Thank you!