LITERATURE REVIEW: Reimplementation and Analysis of p-Stepping Algorithms for Parallel Shortest Paths

Tian Xia
School of Electrical Engineering and Computer Science (EECS)
University of Ottawa
Ottawa, Canada K1S 5M2
txia102@uottawa.ca

October 5, 2021

1 Introduction

SSSP(single-source shortest path problem) is well-known graph algorithms, which has been proved can be implemented with time complexity $\mathcal{O}(n \log n + m)$ on a graph with n nodes and m edges using the well-known Dijkstra algorithm. As for the Parallel version of SSSP algorithms, the Δ-stepping algorithm[9] and the Radius-stepping[3] algorithm are two existing solutions for this problem. However, both of these 2 algorithms have their problem. The Δ-stepping algorithm[9] can not find the worst-case bound on general graphs, whereas, the Radius-stepping[3] doesn't have any efficient implementation available. So this paper[5] described a new stepping algorithm framework, and a new data structure called lazy-batched priority queue (LaB-PQ) to implement a better version of parallel SSSP algorithm with non-trivial worst-case bounds.

2 Literature Review

2.1 Sequential Algorithms

Let a directed graph with $|V| = n$ nodes and $|E| = m$ edges $G = (V, E)$, let s become a vertex ("source") of the graph. The object of SSSP is get the shortest path between any nodes v and our source node s. We denote path between node v and s as $dist(v, s)$, or abbreviated as $dist(v)$. If v is unreachable from u, we'll set $dist(v, u) = \infty$. To solve SSSP problem, plenty of methods have been created, most famous and most widely used sequential algorithms are Djkstra[4] and Bellman-Ford[2][6][10]

2.1.1 Dijkstra[4]

To find SSSP of an arbitrary directed graph with n nodes, m edges and non-negative edge weights sequentially, the algorithm maintains a priority queue(min-heap) minQ to save the unvisited nodes and their estimating shortest path $tent(v)$ as key values. It then repeatedly extracts the vertex u with the minimum $tent(u)$ from minQ and relaxes all incident edges from u to any vertex in minQ, i.e., for an $edge(v, u) \in E$ the algorithm sets $tent(v) := min\{tent(v), tent(u) + c(v, u)\}$. When a vertex is extracted from minQ and all relaxations through it are complete, the algorithm treats this vertex as visited and will not visit it again. Dijkstra's[4] algorithm either stops when minQ is empty or when every vertex has been visited. The time complexity of priority queue-based Dijkstra algorithm is $\mathcal{O}(n + n\log m)$.

2.1.2 Bellman-Ford[2][6][10]

Unlike Dijkstra's algorithm that greedily selects node u with minimum $tent(u)$ that has not yet been processed and does this for all its outgoing edges, the Bellman-Ford[2][6][10] algorithm focuses on the edges rather than the nodes, more specifically Bellman-Ford[2][6][10] just relaxes all the edges and does this $|V| - 1$ times, where $|V|$ is the number of vertices in the graph. To be more specific, it first calculates the shortest distances which have at-most one edge in the path. Then, it calculates the shortest paths with at-most 2 edges, and so on. After the ith iteration of loop, the shortest paths with at most i edges are calculated, the loop will stop after $|V|$–1 time (There are at most $|V|$–1 edges in the graph). The time complexity of Bellman-Ford[2][6][10] algorithm is relative high, it can achieve $\mathcal{O}(mn)$ time complexity, in case $m = n^2$, $\mathcal{O}(n^3)$.

2.2 Parallel Algorithms To Solve SSSP Problem

With the development of multi-core CPUs and other parallel hardware, parallel algorithms can now transfer from theory to practice. The state of art algorithms solving SSSP problem nowadays is Δ-stepping algorithm. Before it, most parallel SSSP Algorithms are based on randomized parallel breadth-first search(BFS) algorithm.

2.2.1 Algorithms Before Δ-stepping Algorithm

Plenty of parallel SSSP Algorithms are implemented using Ullman and Yannakaki's randomized parallel breadth-first search(BFS) algorithm[12]. The native implementation version of BFS algorithm performs $\mathcal{O}(\sqrt{n}{\cdot}log(n))$-limited searches from $\mathcal{O}(\sqrt{n})$ randomly and parallely chosen distinguished nodes.Then it creates an auxiliary graph of nodes with edge weights found from the random search. This native BFS algorithm takes $\mathcal{O}(\sqrt{n} \cdot polylog(n))$ time using $\mathcal{O}(\sqrt{n} \cdot m \cdot polylog(n))$ work with high probability. A more advanced implementation can get $\mathcal{O}(t \cdot polylog(n))$ time using $\mathcal{O}((\sqrt{n} \cdot m + n \cdot m/t + n^3/t^4)polylog(n))$ work for any $t \leq \sqrt{n}$ WHP(with high probability).

2.2.2 Δ-stepping Algorithm[9]

The Δ-stepping algorithm[9] is the state of art parallel algorithms to solve SSSP problem, which combines Dijkstra's algorithm[4] and the Bellman-Ford[2][6][10] algorithm. It calculate the shortest distances in increments of Δ, in step i all the vertices with distances in $[i\Delta, (i+1)\Delta]$ are settled down. And it runs Bellman-Ford as substeps in its each step. Sequential Δ-stepping[9] can be implemented to run in time $\mathcal{O}(n + m + L/\Delta + n\Delta + m\Delta)$. For random edge weights, A parallel version can then take $\mathcal{O}(d \cdot L \cdot logn + log^2 n)$ time and $O(n + m + d \cdot L \cdot logn)$ work on average.

2.2.3 Better Implementation of Δ-stepping Algorithm[9]

Theoretically, Δ-stepping algorithm[9] has been analyzed on random graphs, but there is no bounds on the general graphs(can only be analysed on random edge weights graphs). So in the paper[3], author provided a solution to this problem. He provided a advanced version of Δ-stepping algorithm[9] called Radius-stepping algorithm[3]. This algorithms can only run on weighted, undirected graphs. The input includes: a source vertex s, and a target radius parameter $r(\cdot)$, and a weighted, undirected graph g. The output of this algorithm is same as previous SSSP algorithms:the graph distance $\delta(\cdot)$ from s. Radius-stepping[3] is very similar to Δ-stepping algorithm[9], the only different is instead of increasing the search range by a fixed amount($[i\Delta, (i+1)\Delta]$), the radius-stepping taking $\delta(v)+r(v)$ as its new searching range . The the radius stepping algorithm[3] can solve the SSSP problem in $\mathcal{O}(m \log(n))$ work and $\mathcal{O}(\frac{n}{p} \log n \log pL)$ depth, for $p \leq \sqrt{n}$. The preprocessing phase uses $\mathcal{O}(m \log n + np^2)$ work and $\mathcal{O}(p^2)$ depth.

Both of these stepping algorithms have their own advantages, however, the Δ-stepping algorithm can not find the worst-case bound on general graphs, whereas, the Radius-stepping doesn't have any efficient implementation available. So in paper *Efficient Stepping Algorithms and Implementations for Parallel Shortest Paths*[5] the author described a new stepping algorithm framework, and a new data structure called lazy-batched priority queue (LaB-PQ) to implement a better version of parallel SSSP algorithm with non-trivial worst-case bounds. This paper have implemented and tested on several web and social network dataset including: com-orkut(OK)[13], Live-Journal(LJ)[1], Twitter(TW)[7] and Friendster(FT)[13], one web graph WebGraph(WB)[8], and two road graph[11] Road-USA(USA) and Germany(GE). I plan to test this algorithm on some graphs other than web and social network graphs to check whether this algorithm work properly on different kinds of graph datasets.

References

[1] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, 2006.

[2] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[3] Guy E Blelloch, Yan Gu, Yihan Sun, and Kanat Tangwongsan. Parallel shortest paths using radius stepping. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 443–454, 2016.

[4] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[5] Xiaojun Dong, Yan Gu, Yihan Sun, and Yunming Zhang. Efficient stepping algorithms and implementations for parallel shortest paths. *arXiv preprint arXiv:2105.06145*, 2021.

[6] Lester R Ford Jr. Network flow theory. Technical report, Rand Corp Santa Monica Ca, 1956.

[7] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600, 2010.

[8] Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. Web data commons-hyperlink graphs. *URL http://webdatacommons. org/hyperlinkgraph/index. html*, 2012.

[9] Ulrich Meyer and Peter Sanders. δ-stepping: a parallelizable shortest path algorithm. *Journal of Algorithms*, 49(1):114–152, 2003.

[10] E.F. Moore. *The Shortest Path Through a Maze*. Bell Telephone System. Technical publications. monograph. Bell Telephone System., 1959.

[11] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . URL https://www.openstreetmap.org , 2017.

[12] Jeffrey D Ullman and Mihalis Yannakakis. High-probability parallel transitive-closure algorithms. *SIAM Journal on Computing*, 20(1):100–125, 1991.

[13] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.