Brief Announcement: Near Optimal Bounds for Replacement Paths and Related Problems in the CONGEST Model

Vignesh Manoharan
vigneshm@cs.utexas.edu
The University of Texas at Austin
Austin, USA

Vijaya Ramachandran
vlr@cs.utexas.edu
The University of Texas at Austin
Austin, USA

ABSTRACT

We present round complexity results in the CONGEST model for Replacement Paths (RPaths), Minimum Weight Cycle (MWC), and All Nodes Shortest Cycles (ANSC). We study these fundamental problems in both directed and undirected graphs, weighted and unweighted. Many of our results are optimal to within a polylog factor: For an n-node graph G we establish near linear lower and upper bounds for RPaths if G is directed and weighted, and for MWC and ANSC if G is weighted, directed or undirected; near $\sqrt{n}$ lower and upper bounds for undirected weighted RPaths; and $\Theta(D)$ bound for undirected unweighted RPaths. We also present lower and upper bounds for approximation: a $(2-(1/g))$-approximation algorithm for undirected unweighted MWC that runs in $\tilde{O}(\sqrt{n}+D)$ rounds, improving on the previous best bound of $\tilde{O}(\sqrt{ng}+D)$ rounds, where g is the MWC length, and a $(1+\epsilon)$-approximation algorithm for directed weighted RPaths and $(2+\epsilon)$-approximation for weighted undirected MWC, for any constant $\epsilon > 0$, that beat the round complexity lower bound for an exact solution.

CCS CONCEPTS

• **Theory of computation → Problems, reductions and completeness**; **Shortest paths**; **Distributed algorithms**; • **Networks → Network algorithms**.

KEYWORDS

Distributed algorithms; CONGEST lower bounds; replacement paths; minimum weight cycle

ACM Reference Format:
Vignesh Manoharan and Vijaya Ramachandran. 2022. Brief Announcement: Near Optimal Bounds for Replacement Paths and Related Problems in the CONGEST Model. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing (PODC '22), July 25–29, 2022, Salerno, Italy*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3519270.3538473

1 INTRODUCTION

Consider a communication network with two special nodes s and t, and with information transmitted between them along a shortest path P_{st} between s and t in the network. In the distributed setting, it can be important to maintain communication in the event that a link (i.e., edge) on this path P_{st} fails. This is the Replacement Paths (RPaths) problem, where for each edge e on P_{st}, we need to find a shortest path from s to t that avoids e. This problem has been extensively studied in the sequential setting [5, 12]. Another fundamental graph problem related to shortest paths is the Minimum Weight Cycle (MWC) problem, where we need to compute a shortest simple cycle in a given graph. The All Nodes Shortest Cycle (ANSC) problem, where we want to compute a shortest cycle through each node in a given graph, is also relevant to the distributed setting.

Upper and lower bounds in the CONGEST model are known for the round complexity of several fundamental graph problems. However, we are not aware of any nontrivial results for RPaths, MWC or ANSC except for undirected unweighted MWC. In this paper we present lower bounds and algorithms for these problems in the CONGEST model for directed and undirected graphs, both weighted and unweighted. For several of the variants we consider, our upper and lower bounds are within a polylog factor of each other, giving near optimal bounds. Our main results are listed in Table 1, and we present a summary of select results below. In the full paper [8] we give details of all results listed in Table 1, together with several approximation upper and lower bounds, efficient algorithms for distributed construction of replacement paths and cycles, and some additional results. We also defer an overview of related work to the full version [8].

Table 1: Exact Weight/Size for a Graph G on n nodes. D = undirected diameter, h_{st} = number of edges in P_{st}. $SSSP$ and $APSP$ are round complexity of weighted SSSP and APSP. †Denotes deterministic results, all other results are randomized.

Problem	Lower Bound	Upper Bound
	Directed Weighted Graphs	
RPaths	$\Omega\left(\frac{n}{\log n}\right)$	$O(APSP) = O(n)$
MWC, ANSC	$\Omega\left(\frac{n}{\log n}\right)$	$\tilde{O}(n)$
	Directed Unweighted Graphs	
RPaths	$\Omega\left(\frac{\sqrt{n}}{\log n}+D\right)$	$\tilde{O}(\min(n^{2/3}+\sqrt{nh_{st}}+D,$ $SSSP \cdot h_{st}))$
MWC, ANSC	$\Omega\left(\frac{n}{\log n}\right)$	$O(n)^{\dagger}$
	Undirected Weighted Graphs	
RPaths	$\Omega\left(\frac{\sqrt{n}}{\log n}+D\right)$	$O(SSSP+h_{st})$
MWC, ANSC	$\Omega\left(\frac{n}{\log n}\right)$	$\tilde{O}(n)$
	Undirected Unweighted Graphs	
RPaths	$\Omega(D)$	$O(D)^{\dagger}$
MWC, ANSC	$\Omega\left(\frac{\sqrt{n}}{\log n}\right)$ [3, 6]	$O(n)^{\dagger}$ [4]

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
PODC '22, July 25–29, 2022, Salerno, Italy
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9262-4/22/07.
https://doi.org/10.1145/3519270.3538473

1.1 Techniques and Results

1.1.1 Model and Notation. The CONGEST Model. We use the standard CONGEST model [9], where communication links are always bi-directional and unweighted, and can send and receive $\Theta(\log n)$ bits along an edge in each round — other details of the model are in the full version [8].

Notation. Let $G = (V, E)$ be a directed or undirected graph with $|V| = n$ and $|E| = m$. Let edge (u, v) have non-negative integer weight $w(u, v)$ according to a weight assignment function $w : E \rightarrow \{0, 1, \ldots W\}$, where $W = poly(n)$. Let δ_{st} denote the weight of a shortest path P_{st} from s to t and h_{st} denote the number of edges (hop length) on this shortest path. We define the undirected diameter D as the maximum shortest path distance between any two vertices in the underlying undirected unweighted graph of G.

1.1.2 CONGEST Upper Bounds. We use a variety of techniques to establish upper bounds for RPaths and MWC. For directed weighted RPaths, adapting the sequential algorithm directly to the CONGEST model may require up to n SSSP computations, which is not efficient. Instead, our CONGEST algorithm uses a reduction to APSP, which has a $\tilde{O}(n)$-round CONGEST algorithm [2]. For directed unweighted RPaths, we give two algorithms, each of which is efficient for different values of h_{st} and D (which are determined at runtime). One of these algorithms uses sampling to find replacement paths that are long, and BFS to find short replacement paths. This gives us an algorithm with sublinear round complexity when h_{st} and D are $\tilde{o}(n)$.

We use sampling and approximate bounded-hop shortest paths in our $(1+\epsilon)$-approximation algorithm for directed weighted RPaths. We use sampling in conjunction with source detection in our $\tilde{O}(\sqrt{n}+D)$-round $(2-(1/g))$-approximation algorithm for undirected unweighted MWC. The starting point of our algorithm is the randomized $(2-(1/g))$-approximation algorithm in [10] that runs in $\tilde{O}(\sqrt{ng}+D)$ rounds in a graph with MWC length g. We significantly improve this round complexity by removing the dependence on g, and our algorithm compares favorably to the $\tilde{\Omega}(\sqrt{n})$ lower bound for computing a $(2-\epsilon)$-approximation. Using this unweighted algorithm, along with a weight scaling technique and sampling, we present a $(2+\epsilon)$-approximation algorithm for undirected weighted MWC which has sublinear round complexity when D is $\tilde{o}(n^{3/4})$.

1.1.3 CONGEST Lower bounds. Many of our lower bounds use reductions from Set Disjointness, and other graph problems with known CONGEST lower bounds. Our reduction from Set Disjointness for the $\tilde{\Omega}(n)$ CONGEST lower bound for directed weighted RPaths is inspired by a construction in a sparse sequential reduction from MWC to RPaths in [1]. Other lower bounds that we present with reductions from Set Disjointness include those for directed MWC and undirected weighted MWC to get near linear lower bounds, as well as lower bounds for $(2-\epsilon)$-approximation algorithms. Using reductions from graph problems such as s-t Subgraph Connectivity and Weighted s-t Shortest Path that have been shown in [11] to have an unconditional lower bound of $\Omega\left(\frac{\sqrt{n}}{\log n}+D\right)$, we prove lower bounds for directed unweighted RPaths and undirected RPaths.

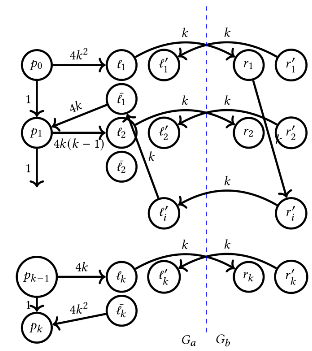

Figure 1: Directed weighted RPaths lower bound construction

2 REPLACEMENT PATHS

2.1 Directed Weighted RPaths Lower Bound

We prove a lower bound of $\Omega(n/\log n)$ for computing RPaths and 2-SiSP in directed weighted graphs. Our proof is a reduction from Set Disjointness using the graph construction $G = (V, E)$ given in Figure 1. Consider an instance of the Set Disjointness problem where the players Alice and Bob are given k^2-bit strings S_a and S_b, and the problem is to determine whether $S_a \cap S_b = \phi$, i.e., that for all indices $1 \leq i \leq k^2$ either $S_a[i] = 0$ or $S_b[i] = 0$. We will reduce the Set Disjointness problem to computing the second simple shortest path in the constructed graph G such that the sets not being disjoint would lead to a shorter replacement path than when the sets are disjoint. This intuition is formally captured in Lemma 2.1.

Lemma 2.1. *If $S_a \cap S_b \neq \phi$, the shortest replacement path from vertex p_0 to vertex p_k in G has weight at most $(4k^2+9k-1)$. Otherwise, the shortest replacement path has weight at least $(4k^2+12k)$.*

To complete the reduction from Set Disjointness, assume that there is a CONGEST algorithm $\mathcal{A}$ that takes $R(n)$ rounds to compute the weight of replacement paths in a directed weighted graph on n vertices. Consider the graph partition G_a, G_b in Figure 1. Any edge of G going from a vertex in G_a to G_b is considered to be in the cut separating the partition, and this cut has $4k$ edges. Note that G_a is completely determined by the string S_a and G_b is completely determined by S_b. Alice and Bob will communicate to simulate running algorithm $\mathcal{A}$ on G. Alice simulates G_a and Bob simulates G_b, and they communicate any information sent across a cut edge. Since there are $4k$ cut edges, and $\mathcal{A}$ can send $O(\log n)$ bits through each edge per round, Alice and Bob can communicate up to $O(4k \cdot \log n)$ bits per round to simulate $\mathcal{A}$ for a total of $O(4k \cdot \log n \cdot R(n))$ bits. Alice and Bob and determine if their sets are disjoint by checking if the shortest replacement path computed by $\mathcal{A}$ has weight more than $(4k^2+9k-1)$. Any communication protocol for Set Disjointness must use at least $\Omega(k^2)$ bits and $n = \Theta(k)$, and hence $R(n)$ is $\Omega\left(\frac{n}{\log n}\right)$.

2.2 Directed Unweighted RPaths Upper Bound

Algorithm 1 computes replacement paths in a directed unweighted graph in $\tilde{O}(n^{2/3}+\sqrt{nh_{st}}+D)$ rounds. In the full version, we also give an algorithm that runs in $O(SSSP \cdot h_{st}) = O\left((\sqrt{n}D^{1/4}+D) \cdot h_{st}\right)$

rounds, which is more efficient for certain ranges of h_{st}. Combining the two, we have an algorithm that runs in $\tilde{o}(n)$ rounds unless either D or h_{st} is $\tilde{\Omega}(n)$.

Algorithm 1 Directed Unweighted RPaths

Let $p = n^{1/3}$ if $h_{st} < n^{1/3}$ and $p = \sqrt{nh_{st}}$ if $h_{st} \geq n^{1/3}$, and let $h = n/p$.
Sample each vertex $v \in G$ into set S with probability $\Theta\left(\frac{\log n}{h}\right)$.
for vertex $v \in P_{st} \cup S$, perform BFS starting from v on $G - P_{st}$ up to h hops to compute unweighted shortest paths $d'(v, u)$. Broadcast $\{d'(v, u) \mid u \in S\}$.
for vertex $v_a, v_b \in P_{st}$, compute the best detour (short or long) from v_a to v_b: $D(v_a, v_b) = \min\big(d'(v_a, v_b),$
    $\min_{u,v \in S} (d'(v_a, v) + d'(v, u) + d'(u, v_b))\big)$
Compute best replacement path for edge e on P_{st} as $\min_{a,b} \delta_{sv_a} + D(v_a, v_b) + \delta_{v_b t}$ for v_a before e and v_b after e.

We compute short detours, detours with hop length at most h, by performing a BFS of h hops from each vertex on P_{st} on the graph $G - P_{st}$. This takes $O(h_{st} + h)$ rounds. To compute the long detours with hop length at least h, we sample each vertex with probability $\Theta((\log n)/h)$ so that we get a vertex set S of size $\tilde{O}(n/h) = \tilde{O}(p)$ with high probability. Also w.h.p., every path of h hops, including all long detours, contains at least one vertex from S. We perform a BFS of h hops from each vertex in S on the graph $G - P_{st}$, which takes $\tilde{O}(p + h)$ rounds. Each node $v \in S$ broadcasts the values $d'(v, u)$, $d'(v_a, v)$ and $d'(v, v_b)$ $\forall u \in S, v_a, v_b \in P_{st}$ to all other nodes. A total of $\tilde{O}(p^2 + p \cdot h_{st})$ weights are broadcasted, which takes $\tilde{O}(p^2 + p \cdot h_{st} + D)$ rounds. With this information, each vertex on the path P_{st} can locally compute the best long detour $D(v_a, v_b)$, and compute the best replacement paths for each edge locally. This gives a total round complexity of $\tilde{O}(p^2 + p \cdot h_{st} + h + D)$, and combining the two parameter choices in the algorithm we get an algorithm with a round complexity of $\tilde{O}(n^{2/3} + \sqrt{nh_{st}} + D)$.

2.3 Approximate Undirected MWC

A CONGEST algorithm in [10] computes a $(2 - \frac{1}{g})$-approximation of MWC length (or *girth*) in $O(\sqrt{ng} + D)$ rounds, where g is the girth. We significantly improve this result by removing the dependence on g to give an $\tilde{O}(\sqrt{n} + D)$ round algorithm (Algorithm 2).

First, we argue correctness: Consider the case when all vertices in a minimum weight cycle C are contained in the $\sqrt{n}$-neighborhood of some vertex v in C. When computing the $\sqrt{n}$-neighborhood of v, an edge of C furthest from v is a non-tree edge and C is computed as one of the candidate cycles — the MWC length is computed exactly. In the other case where no vertex v in C whose $\sqrt{n}$-neighborhood contains C, our algorithm computes a $(2 - (1/g))$-approximation. This $\sqrt{n}$-neighborhood of v must have size $\sqrt{n}$ and contain a sampled vertex w at a hop distance at most $\lfloor g/2 \rfloor$. The BFS from this vertex w detects cycle C as a candidate cycle of length at most $2g$. By a more refined argument for even-length cycles, we show a $(2 - (1/g))$-approximation guarantee.

In line 1 we use an $(R + h)$-round algorithm in [7] to compute for each vertex its R closest neighbors within h hops (the *source detection* problem); here we have $R = \sqrt{n}$ and $h = D$ so line 1 runs

Algorithm 2 Undirected Unweighted MWC 2-Approximation

0: Sample each vertex $v \in G$ into set S with probability $\Theta\left(\frac{\log n}{\sqrt{n}}\right)$
1: **for** each vertex $v \in G$, find $\sqrt{n}$ closest vertices to v. For each non-tree edge in the partial shortest path tree computed, record a candidate cycle.
2: **for** vertex $w \in S$, perform BFS starting from w. For each non-tree edge in the BFS tree, record a candidate cycle.
3: Return the minimum among all recorded candidate cycles.

in $O(\sqrt{n} + D)$ rounds. The set S computed in line 2 has size $\tilde{O}(\sqrt{n})$ w.h.p. in n and we perform a $|S|$-source BFS, which takes $\tilde{O}(\sqrt{n} + D)$ rounds. The global minimum computation in line 3 takes $O(D)$ rounds, giving a total round complexity of $\tilde{O}(\sqrt{n} + D)$.

We use this algorithm along with a weight scaling technique to compute a $(2 + \epsilon)$-approximation of undirected *weighted* MWC in $\tilde{O}\left(\min(n^{3/4}D^{1/4} + n^{1/4}D, n^{3/4} + n^{0.65}D^{2/5} + n^{1/4}D, n)\right)$ rounds. This algorithm has sublinear round complexity when D is $\tilde{o}(n^{3/4})$ and compares favorably with our $\tilde{\Omega}(n)$ lower bound for $(2 - \epsilon)$-approximation of undirected weighted MWC. This algorithm, and other approximation results, are presented in the full version [8].

ACKNOWLEDGMENTS

This work was supported by NSF Grant CCR-2008241.

REFERENCES

[1] Udit Agarwal and Vijaya Ramachandran. 2018. Fine-grained complexity for sparse graphs. In *Proc. of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 239–252.
[2] Aaron Bernstein and Danupon Nanongkai. 2019. Distributed exact weighted all-pairs shortest paths in near-linear time. In *Proc. of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 334–342.
[3] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. 2014. On the power of the congested clique model. In *Proc. of the 2014 ACM Symposium on Principles of Distributed Computing*. 367–376.
[4] Stephan Holzer and Roger Wattenhofer. 2012. Optimal Distributed All Pairs Shortest Paths and Applications. In *Proc. of the 2012 ACM Symposium on Principles of Distributed Computing*. 355–364.
[5] Naoki Katoh, Toshihide Ibaraki, and Hisashi Mine. 1982. An efficient algorithm for k shortest simple paths. *Networks* 12, 4 (1982), 411–427.
[6] Janne H. Korhonen and Joel Rybicki. 2017. Deterministic Subgraph Detection in Broadcast CONGEST. In *21st International Conference on Principles of Distributed Systems (OPODIS 2017)*, Vol. 95. 4:1–4:16.
[7] Christoph Lenzen, Boaz Patt-Shamir, and David Peleg. 2019. Distributed distance computation and routing with small messages. *Distributed Computing* 32, 2 (2019), 133–157.
[8] Vignesh Manoharan and Vijaya Ramachandran. 2022. Near Optimal Bounds for Replacement Paths and Related Problems in the CONGEST Model. (2022). arXiv:2205.14797 [cs.DS] https://arxiv.org/abs/2205.14797
[9] David Peleg. 2000. *Distributed computing: a locality-sensitive approach.* SIAM.
[10] David Peleg, Liam Roditty, and Elad Tal. 2012. Distributed Algorithms for Network Diameter and Girth. In *Automata, Languages, and Programming*. Springer Berlin Heidelberg, 660–672.
[11] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. 2012. Distributed verification and hardness of distributed approximation. *SIAM J. Comput.* 41, 5 (2012), 1235–1265.
[12] Jin Y Yen. 1971. Finding the k shortest loopless paths in a network. *Management Science* 17, 11 (1971), 712–716.