DOS Assignment 2 : Test Results and Analysis

Team members: Anirudh Subramanian (UFID:94453124), Divya Ramachandran (UFID:46761308)

Gossip Algorithm

Approach: For the gossip algorithm, the termination process is by determining a certain number of rounds after which the algorithm must stop executing. The initial algorithm is designed to produce a feedback on a per-node basis in order to know whether each node has received the message at least once. In order to determine the number of rounds required, the algorithm was tested for different topologies with varying number of nodes. This gave rise to different equations as a function of the total number of nodes that could be used for each topology.

Initial data collection:

The buffer signifies the percentage of nodes that we will count in addition to the original value of the required termination rounds observed in the collected data. We have set the buffer to be 5% i.e.

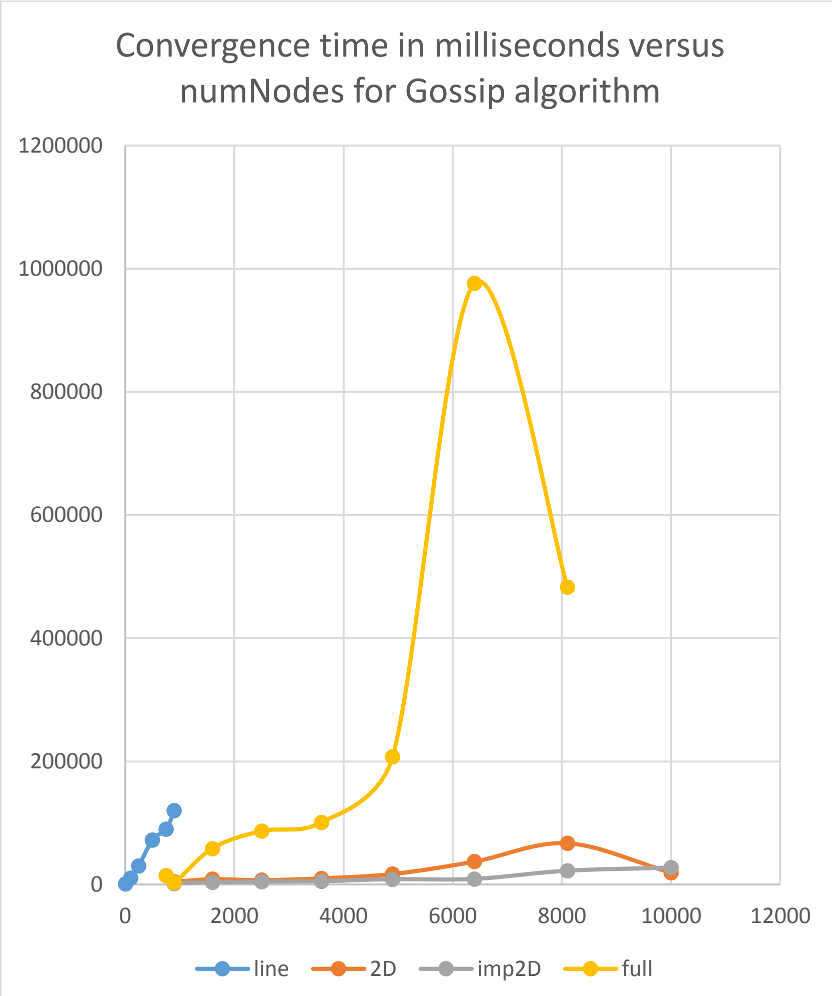
#termination rounds limit = observed value + (0.05*total nodes)

The following table consists of our observations for the number of rounds observed for every node to receive the message atleast once, the convergence time and the calculated termination rounds limit with buffer, for varying number of total nodes and for all four topologies.

buffer = 0.05

	line			2D			imp2D			full		
numNodes	#rounds	time (ms)	#rounds with buffer	#rounds	time (ms)	#rounds with buffer	#rounds	time (ms)	#rounds with buffer	#rounds	time (ms)	#rounds with buffer
10	16	969	16.5									
100	145	10275	150									
250	405	30452	417.5									
500	875	72316	900									
750	1220	89866	1257.5							18	14574	55.5
900	1777	120328	1822	51	4679	96	24	1813	69	18	3037	63
1600	OoM			61	8824	141	25	3844	105	18	58568	98
2500	OoM			67	7205	192	30	4629	155	25	86739	150
3600	OoM			85	10194	265	33	5439	213	25	101132	205
4900	OoM			115	17164	360	35	8645	280	25	207315	270
6400	OoM			132	37290	452	37	9093	357	30	976516	350
8100	OoM			145	67077	550	39	22510	444	30	483060	435
10000	OoM			160	19438	660	40	27443	540	OoM		

Following is the graph of convergence time for different topologies for varying number of nodes.



time (ms)

numNodes	line	2D	imp2D	full
10	969			
100	10275			
250	30452			
500	72316			
750	89866			14574
900	120328	4679	1813	3037
1600		8824	3844	58568
2500		7205	4629	86739
3600		10194	5439	101132
4900		17164	8645	207315
6400		37290	9093	976516
8100		67077	22510	483060
10000		19438	27443	

As seen, imperfect 2D grid performs well followed closely by the 2D grid. The line network and full network lie at extremeties when it comes to number of neighbours and hence both suffer in terms of performance. In line topology, the number of neighbours is too low and in full topology, the neighbours are so many that message propagation might become unstable and might lack direction.

Following is the graph of the number of rounds required for each node to receive the message at least once for different topologies for varying number of nodes.

From the observed results, the relationship between the number of nodes and number of rounds required for convergence are derived. However, these relations are specific to the test network (affected by test network and system conditions). Thus, it will be a good practice to derive these relations afresh for a new setup.

The derived relations show that the ideal termination limit is a function of the log (#nodes) which is expected.

Derived relations are –

Line: #termination limit = $0.0007x^2 + 1.2654x + 19.088$

2D: #termination limit = $47.987\ln(x) - 291.87$

imp2D: #termination limit = $7.2387\ln(x) - 26.54$

full: #termination limit = $5.4294\ln(x) - 19.182$

where x = number of nodes

The second graph shows the number of rounds required after taking into account a 5% buffer for each node to receive the message at least once for different topologies for varying number of nodes. This is to avoid premature termination of gossip propagation. Different relationships have been derived taking into account the buffer. They are-

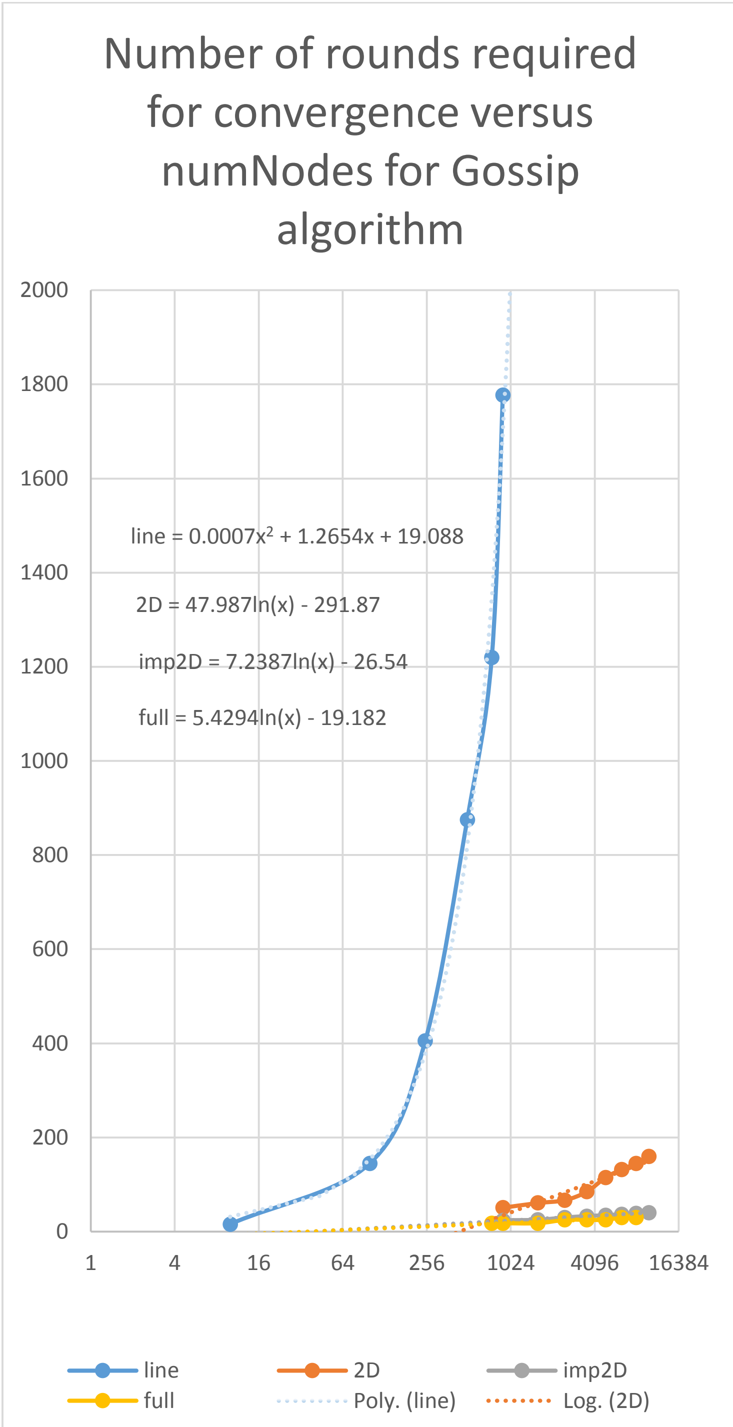
Line: #termination limit = $0.0007x^2 + 1.3154x + 19.088$

2D: #termination limit = $231.87\ln(x) - 1563.7$

imp2D: #termination limit = $191.12\ln(x) - 1298.3$

full: #termination limit = $149.42\ln(x) - 974.77$

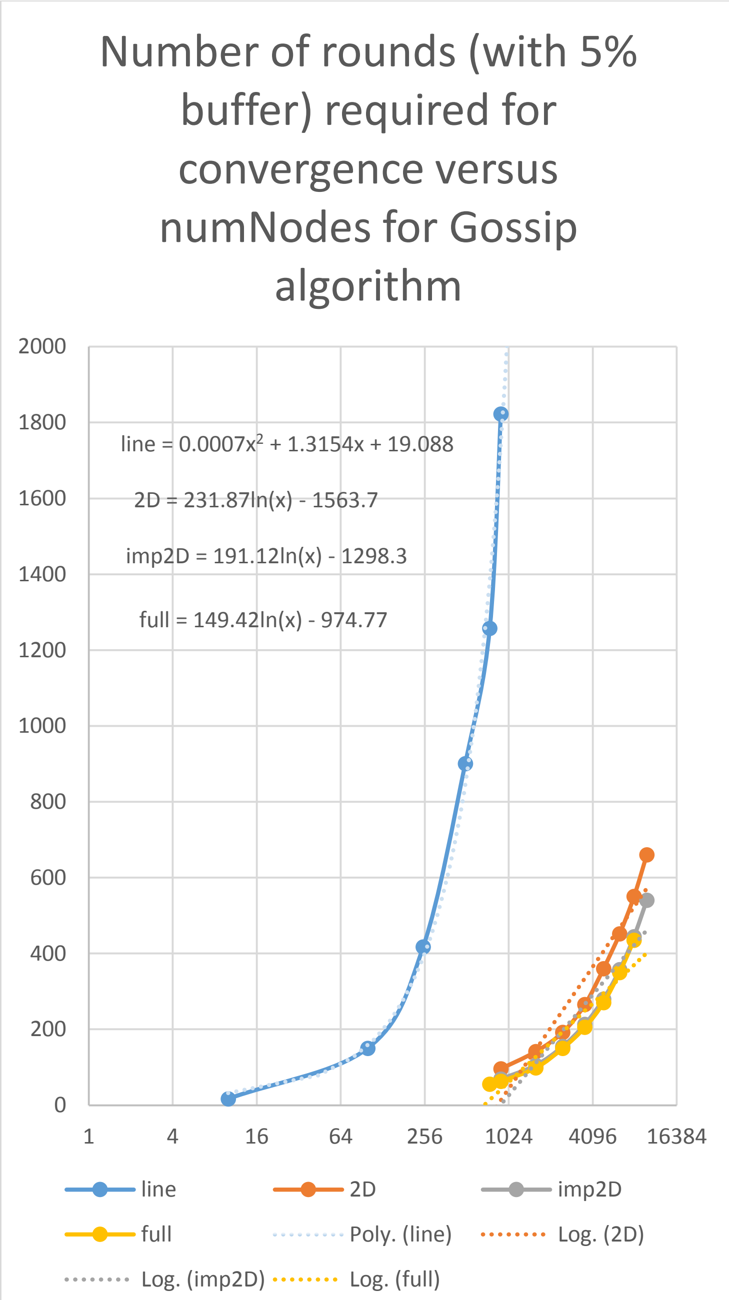
where x = number of nodes



#rounds

numNodes	line	2D	imp2D	Full
10	16			
100	145			
250	405			
500	875			
750	1220			18
900	1777	51	24	18
1600		61	25	18
2500		67	30	25
3600		85	33	25
4900		115	35	25
6400		132	37	30
8100		145	39	30
10000		160	40	

As seen, the number of rounds required for converging in case of line topology is very high. The random and possibly wider range of selection for the full topology and imperfect 2D topology sees convergence in fewer rounds.



#rounds

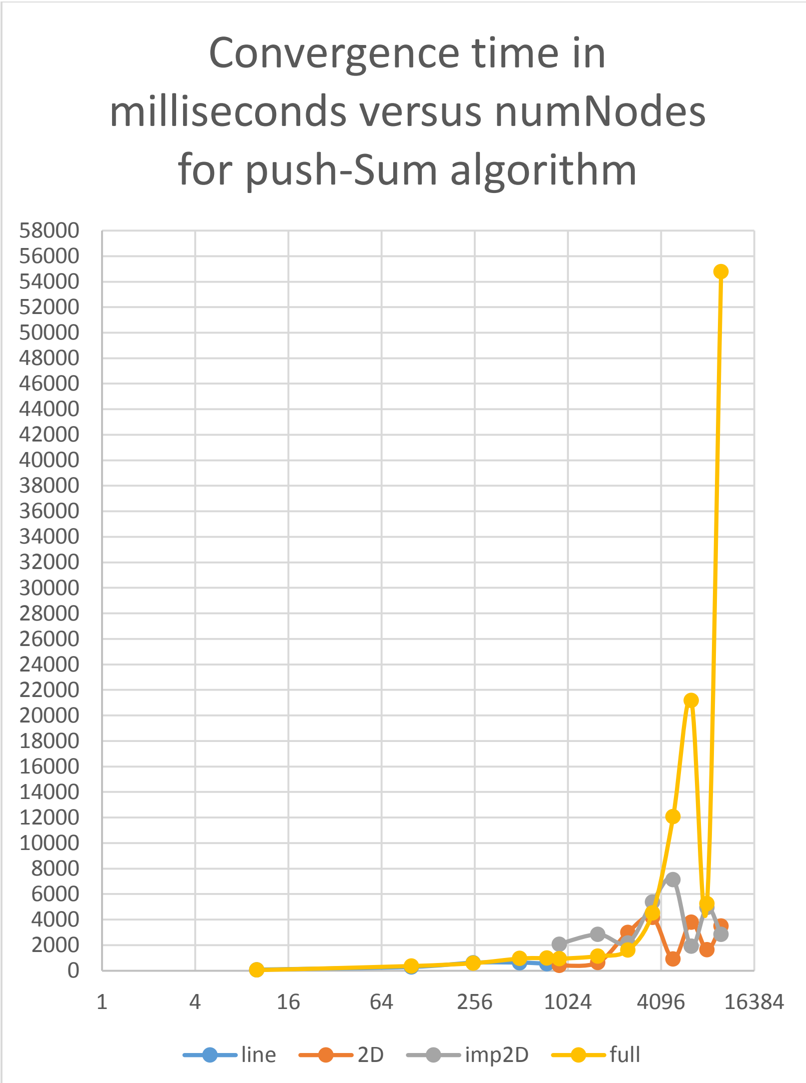
numNodes	line	2D	imp2D	full
10	16.5			
100	150			
250	417.5			
500	900			
750	1257.5			55.5
900	1822	96	69	63
1600		141	105	98
2500		192	155	150
3600		265	213	205
4900		360	280	270
6400		452	357	350
8100		550	444	435
10000		660	540	

Push-Sum Algorithm

For the push-sum algorithm, following data was collected for different topologies for varying number of nodes.

	line			2D			imp2D			full			
numNodes	time (ms)	Converged Ratio		time (ms)	Converged Ratio		time (ms)	Converged Ratio		time (ms)	Converged Ratio		Actual Average
10	70	3.89409129								52	4.826923077		4.5
100	289	51.57244512								373	48.4665797		49.5
250	621	115.3164353								592	125.0533691		124.5
500	626	136.529324								958	247.2301698		249.5
750	543	87.34585468								984	370.0690278		374.5
900				408	456.5395324		2082	444.2086081		945	450.1014634		449.5
1600				633	821.6557686		2844	787.1198146		1141	799.8796449		799.5
2500				2999	1182.995529		2162	1223.430669		1614	1243.915124		1249.5
3600				4175	2197.604593		5379	1779.649552		4527	1794.42334		1799.5
4900				919	2295.680723		7145	2429.34783		12090	2456.241017		2449.5
6400				3810	3846.001822		1924	3157.305269		21193	3214.35574		3199.5
8100				1649	4834.028757		4959	4025.42207		5263	4043.809453		4049.5
10000				3499	4483.669192		2850	5018.40448		54800	4980.562383		4999.5

Following page contains the graph obtained on plotting the convergence time for different topologies for varying number of total nodes.



time in milliseconds

numNodes	line	2D	imp2D	full
10	70			52
100	289			373
250	621			592
500	626			958
750	543			984
900		408	2082	945
1600		633	2844	1141
2500		2999	2162	1614
3600		4175	5379	4527
4900		919	7145	12090
6400		3810	1924	21193
8100		1649	4959	5263
10000		3499	2850	54800

For the push-sum algorithm, time taken to converge is found to be high for the full topology. This could be attributed to the chances of communication being localized and hence converging. For a small number of nodes, the results for full and line topology are similar.

In this case, a systematic approach (2D) seems to work better for most ranges of total nodes.