Distributed MIS Algorithms in Coordinated Graphs

Ionut-Daniel Chirvase 1[1], Alexandru Popa 2[1,2], Anisur Rahaman Molla 3
[1]Doctoral School of Computer Science, University of Bucharest, Romania
[2]Full Professor at University of Bucharest, Romania
[3]ISI Kolkata
chirvaseionutdaniel@gmail.com

Abstract

The best distributed algorithm for solving maximal independent set (MIS) problem for general graphs uses O(log *n*) rounds. The lower bound for any algorithm for maximal independent set in the local congest model is $O(\Delta + \log^* n)$ rounds.

In this paper, we want to improve the bounds for computing MIS using the widely used standard model (called the congest model) to study distributed algorithms.

We study the problem in the case of coordinated graphs, which are undirected graphs constructed by adding a new vertex (called coordinator) to graphs and edges from it to all the vertices. Initially each node has limited knowledge about the graph. The goal is to compute a MIS in the core subgraph G in the distributed congest model with minimum number of communication rounds. The coordinator vertex may assist to achieve a fast computation.

A fundamental constraint in distributed systems is the cost of sending messages to other nodes. We assume that the communication occurs in synchronous rounds, i.e., nodes run at the same processing speed and any message that is sent by some node *v* to its neighbors in some round *r* will be received at the end of *r*. To ensure scalability, we restrict the number of bits that are processed and sent per round by each node to be polylogarithmic in *n*, the network size. Note that the computation that is performed by the nodes locally is 'free', i.e., it does not affect the number of rounds.

Keywords: distributed; MIS; algorithms
Domain: Computer Science
Section: New (2020) thesis proposals

Motivation

Maximal independent set problem plays a very important role in distributed computing, being one of the very first problem in the case of studying distributed algorithms. Moreover, it still remained one of the most studied problem in the last decades due to its many applications in symmetry breaking [1].

Designing new distributed algorithms, which will prove their efficiency focusing on running time, to improve the bounds for computing MIS still represents a very good research topic.

Methodology of Research

In this paper we study the MIS problem in the case of coordinated graphs. Next, we give the definition of a coordinate graph. Given an undirected graph G = (V, E) with $|V| = n$ and $|E| = m$, a coordinated graph G' is constructed by adding a new vertex *x* to V (called coordinator) and edges from *x* to all the vertices in V. That is, $G' = (V \cup \{x\}, E \cup \{(x,v) : v \in V\})$. We call the graph G as core subgraph of G'.

Further, we give the definition of a maximal independent set. Given a graph G = (V, E), an independent set $M \subseteq V$ is a subset of the vertices such that $\forall\, x, y \in M, (x, y) \notin E$. M is maximal if there does not exists a vertex $v \in V - M$ such that $M \cup \{v\}$ is an independent set.

Given a coordinated graph G', the problem is to find a maximal independent set in the core subgraph G in the distributed congest model. In the end, each vertex in G outputs whether it's in the MIS or not.

The goal is to compute MIS with minimum number of communication rounds, where the coordinator vertex may assist to achieve a fast computation.

To do so, we design and then, implement and test some distributed algorithms.

Results and Comparison with State-of-the-art

As previously said, in the last decades the problem related to the maximal independent set was studied by many researchers [1, 2, 3, 7, 8, 9]. We refer to [6] for a thorough review of the state-of-the-art.

In this paper, we design an algorithm that computes the MIS using BFS search. To do so, we first design an algorithm that computes a spanning tree (T) from the root node in O(log n) rounds.

Secondly, we design an algorithm that computes BFS tree in O(log n) rounds. We apply the previous algorithm to compute the spanning tree of G. After every node sends their parent node to the coordinator, the last one chooses one node *v* as the root, assign *v* as level 0, then level 1 to the neighbors of *v* in the spanning tree T, then level 2 to the two-hop neighbors of *v* in T and so on. Note that the coordinator can assign the level locally as it has the complete structure of T. The coordinator sends the corresponding level number to all the nodes. Root node *v* starts from round 0. At every round each node *u* broadcasts its level to all neighbors in the original graph G. Then every node updates their parent node choosing the lowest level node in their neighbor in G (if such lower level neighbor exists). This way *u* chooses a parent node closer to the root than its earlier parent. The coordinator computes a new BFS tree using all the edges it received so far and then, it transmits the updated levels to the nodes of G.

Finally, capitalizing algorithms explained previously we design an algorithm, recursive and informally, that computes MIS in … rounds.

To present the algorithm, we make the following two observations. First, the MIS in a coordinated graph can be computed in O(Δ), where Δ is the maximum degree. Second, the MIS in any graph G (with or without a coordinator node) can be computed in O(D), where D is the diameter of G.

If the maximum degree in G is O(log n) we compute the MIS using the first observation and the algorithm stops. Then, we apply the previous algorithm to compute BFS tree in O(log n) rounds. If the depth D of the BFS tree is O(log n), we compute the MIS using the second observation and the algorithm stops. Then, since the sets of the vertices at odd levels and at even levels of the BFS tree, are independent, we first compute, recursively in parallel the MIS for each set of vertices at the even level, remove their neighbors, and then compute the MIS similarly for the remaining vertices at odd levels.

Conclusions

Studying the problem in the case of coordinated graphs and using the distributed congest model [4, 5] may help to achieve a fast computation which may also lead to improve the bounds for computing MIS.

Acknowledgements

The PhD student want to thank Alexandru Popa, the PhD student supervisor, for suggesting this problem and would also like to acknowledge helpful conversations with the other two authors.

References

1. M. Luby, A simple parallel algorithm on the maximal independent set problem, SIAM Journal on Computing 15(4) (1986), 1036-1053
2. N. Alon, L. Babai and A. Itai, A fast and simple randomized parallel algorithm for the maximal independent set problem, J. Algorithms 7 (1986), 567-583
3. R. Karp and A. Wigderson. A fast parallel algorithm for the maximal independent set problem. Journal of the ACM (JACM), 32(4):762-773, 1985
4. G. Pandurangan and M. Khan. Theory of communication networks. In Algorithms and Theory of Computation Handbook, Second Edition. CRC Press, 2009.
5. D. Peleg. Distributed computing: a locality-sensitive approach. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
6. L. Barenboim, M. Elkin, S. Pettie ad J. Schneider. The locality of distributed symmetry breaking. J. ACM, 63(3):20:1-20:45, 2016
7. M. Ghaffari. An improved distributed algorithm for maximal independent set. In Pro. Of ACM-SIAM Symp. On Disc. Alg. (SODA), 2016
8. G. Blelloch, J. Fineman and J. Shun. Greedy sequential maximal independent set and matching are parallel on average. In Proceedings of the twenty-fourth annual ACM Symposium on Parallelism in algorithms and architectures, p. 308-317. ACM, 2012
9. M. Goldberg and T. Spencer. A new parallel algorithm for the maximal independent set problem. SIAM Journal on Computing, 18(2):419-427, 1989