

A Creative Survey of Fully Dynamic Maximal Independent Set in Expected Poly-log Update Time

Authors: Liang, Tianxin, Li, Wei

Date: 2022-02-24T09:24:24+00:00

Abstract

This paper focus on the researches of Maximal Independent Set (MIS). Based on reading and analysis of several recent papers, we divide the MIS problems into several classifications. The first is the classification based on the research objects, including the solution and maintenance of MIS; the second is the classification based on research methods, including serial, parallel, deterministic and randomized algorithms; the third is experimental analysis, including worst time complexity and expected time complexity.

Full Text

A Creativity Survey of Fully Dynamic Maximal Independent Set in Expected Poly-log Update Time

TIANXIN LIANG

Department of Computer Science, University of Chinese Academy of Sciences
liangtianxin21@mails.ucas.ac.cn

WEI LI

Institute of Computing Technology, Chinese Academy of Sciences
liwei@ict.ac.cn

Abstract

This paper focuses on research related to Maximal Independent Set (MIS). Based on a comprehensive reading and analysis of several recent papers, we categorize MIS problems into several classifications. The first classification is based on research objects, encompassing both the solution and maintenance of MIS. The second classification is based on research methods, including serial, parallel, deterministic, and randomized algorithms. The third classification concerns

experimental analysis, covering worst-case time complexity and expected time complexity.

Keywords: maximal independent set, classification, algorithm, theoretical computer science, time complexity

1 Introduction

A maximal independent set (MIS) of a given graph $G = (V, E)$ is a subset M of vertices such that M contains no two neighboring vertices, and every vertex in V has a neighbor in M . The MIS problem is fundamental in computer science, bearing both theoretical and practical significance.

In multi-core environments, parallel threads are widely employed for resource scheduling, leader election, feature tracking, and other applications. The efficient computation of MIS has been studied for decades, with efficient parallel algorithms emerging as early as the 1980s. With the ever-increasing volume of information and user demands, the dynamic MIS problem has attracted widespread attention since 2010.

We select a recent paper on dynamic MIS algorithms (reference [1]) as our focal point. Based on our reading and analysis of several related references, we categorize the MIS problem into distinct classifications that reveal the research focus and innovation of each work. These classifications are organized by research objects, research methods, and experimental analysis.

Section 2 categorizes references based on research objects, including the solution and maintenance of MIS. Section 3 classifies them by research methods, encompassing serial, parallel, deterministic, and randomized algorithms. Section 4 organizes them by experimental analysis, covering worst-case and expected time complexity. The remainder of this paper is structured as follows: Section 2 presents the classification of research objects for MIS problems, Section 3 introduces the classification of research methods, Section 4 compares experimental analyses in related literature, Section 5 discusses future research opportunities, and Section 6 concludes the paper.

2 Classification of Research Objects

Table 1: Different Research Objects

Criteria	Theory Research	Algorithm Application
MIS finding	I. [2][12]	III. [8][11][14]
MIS maintaining	II. [1][3][4][5][7][9][10][13]	IV. [8]

As a fundamental model in computer theory, the MIS problem holds significance not only for theoretical research but also for practical applications. Once an MIS

has been computed, if the local environment changes, we wish to construct a new MIS as quickly as possible. This section employs two independent criteria to divide research objects into distinct types.

First, **Aspect** distinguishes between Theory Research and Algorithm Application. Theory Research aims to develop general algorithms for abstract mathematical models, focusing on algorithmic correctness and complexity. Algorithm Application involves constructing appropriate mathematical models for practical problems and transforming them into MIS problems, emphasizing the deployment of algorithms in real-world scenarios.

Second, **Problem Type** differentiates between MIS finding (Static MIS) and MIS maintaining (Dynamic MIS). MIS finding studies how to efficiently compute an MIS in an undirected graph. Since serial algorithms typically require $O(n)$ time complexity, parallel algorithms are often investigated. MIS maintaining examines how to efficiently compute a new MIS when node relationships change after an initial MIS has been obtained.

Based on these classification criteria, we present the taxonomy in Table 1. The meaning of each category is as follows:

Type I: Studies how to efficiently find a maximal independent set in a short time, given an undirected graph with almost no additional information. This is called “Static MIS.”

Type II: Studies how to efficiently find a new maximal independent set when the local environment is disturbed (such as through node or edge additions/deletions), given that a maximal independent set has already been found. This is called “Dynamic MIS.”

Type III: Studies the application of Static MIS algorithms in practical problems.

Type IV: Studies the application of Dynamic MIS algorithms in practical problems.

2.3 Explanation of Different Types

References [2] and [12] belong to Type I. Reference [2] computes MIS in parallel environments using $O(|E|d_{\max})$ processors. Reference [12] employs two strategies to solve the MIS problem: assigning identical copies of a simple algorithm to small local portions, or converting a Monte Carlo algorithm into a simple deterministic algorithm.

References [1][3][4][5][7][9][10][13] belong to Type II. Reference [1] is our target paper. Reference [7] studies the MIS problem in distributed settings and proposes the concept of dynamic MIS. Reference [3] presents the first algorithm for dynamically maintaining MIS in sublinear time, enabling recomputation of MIS in sublinear time for edge additions, deletions, and node modifications. Refer-

ences [9][10][4][13][5] successively improve the dynamic MIS algorithm, gradually reducing its time complexity.

References [8][11][14] belong to Type III. Reference [8] introduces applications of MIS in leader election, reference [14] discusses its use in resource allocation, and reference [11] covers applications in distributed graph algorithms.

Reference [8] also belongs to Type IV, addressing the impact of dynamically modifying MIS in leader election scenarios.

3 Classification of Research Methods

Table 2: Different Research Methods

Randomization	Serial	Parallel
Deterministic	I. [4][9][10][13]	III. [12]
Randomized	II. [1][4][5][6][9]	IV. [2]

To overcome the bottleneck of serial running time, parallel algorithms are often necessary to achieve acceleration through simultaneous multi-threaded execution. However, parallel algorithms differ significantly from serial algorithms in design philosophy. Beyond the complexity of individual threads, parallel algorithms must also consider execution order and synchronization between threads. Furthermore, some randomized algorithms exhibit superior robustness and generalization capabilities, though their performance analysis is more complex than that of deterministic algorithms.

This section employs two independent criteria to classify research methods. First, **Implementation** distinguishes between Serial and Parallel algorithms. A serial algorithm executes instructions sequentially on a single processor, while a parallel algorithm executes threads synchronously across multiple processors. Second, **Randomization** differentiates between Deterministic and Randomized algorithms. Deterministic algorithms execute identically for the same inputs, whereas randomized algorithms may exhibit different execution paths for identical inputs.

Based on these criteria, we present the classification in Table 2. The meaning of each category is as follows:

Type I: Serial deterministic algorithms that run on single-core processors.

Type II: Serial randomized algorithms that run on single-core processors.

Type III: Parallel deterministic algorithms that run on multi-core processors.

Type IV: Parallel randomized algorithms that run on multi-core processors.

3.3 Explanation of Different Types

References [3][9][10][13] belong to Type I, each presenting at least one deterministic algorithm suitable for serial environments.

References [1][4][5][6][9] belong to Type II, each presenting at least one randomized algorithm suitable for serial environments.

Reference [12] belongs to Type III, presenting a deterministic algorithm for parallel environments.

Reference [2] belongs to Type IV, presenting a randomized algorithm for parallel environments.

4 Review of Experimental Analysis

This section classifies evaluation metrics and system parameters as shown in Table 3. All experimental analyses are categorized according to these metrics and parameters. As evident from Table 3, most references compare worst-case time complexity and expected time complexity.

Table 3: Experiments with Different Metrics and Parameters

Metric	Worst-case time complexity	Expected time complexity
Static MIS	I. [12]	III. [2]
Dynamic MIS	II. [1][5][7][9]	IV. [1][3][4][5][6][9][10][13]

4.1 Metric of Evaluation

Worst-case time complexity refers to the maximum time complexity across all possible cases, defined as:

$$\text{Worst-case time complexity} = \max\{\text{running time complexity in a certain case}\}$$

Expected time complexity refers to the average time complexity across all cases, defined as:

$$\text{Expected time complexity} = \frac{\sum \text{running time complexity in a certain case}}{\text{Number of total cases}}$$

System Parameters

Static MIS represents the evaluation of MIS finding, which studies how to efficiently compute an MIS in an undirected graph. **Dynamic MIS** represents the evaluation of MIS maintaining, which studies how to find a new MIS when node relationships change after an initial MIS has been obtained.

Experimental Comparison

In reference [7], the author studies the problem of maintaining an MIS in distributed dynamic settings where the graph evolves over time, investigating whether algorithms can outperform recomputation.

In reference [3], the author presents an $\tilde{O}(\min\{\Delta, m^{3/4}\})$ amortized expected-time deterministic algorithm for maintaining MIS, representing the first sublinear-time algorithm for this problem.

In reference [9], the author improves the worst-case time complexity to $O(\min\{\Delta, m^{3/4}\})$ and presents a randomized algorithm with $O(\min\{\Delta, \sqrt{m} \log^{1.5} m\})$ expected time complexity.

In reference [10], the author independently presents an algorithm with $O(\min\{\Delta, m^{3/4}\})$ complexity.

In reference [4], the expected time complexity is further improved to $\tilde{O}(\sqrt{n})$ and $\tilde{O}(m^{1/2})$.

In reference [13], the author improves the expected time complexity to $O(\alpha^2 \cdot \log^2 n)$ on uniformly sparse graphs.

In reference [5], the author presents a polynomial-time algorithm with $O(\log^2 \Delta \cdot \log^2 n)$ expected time complexity and $O(\log^2 \Delta \cdot \log^4 n)$ worst-case time complexity.

In reference [2], the author presents a parallel algorithm for solving maximal independent set using $O(|E| \cdot d_{\max})$ processors with $O(\log n)$ expected time complexity.

In reference [12], a parallel algorithm for solving maximal independent set using $O(n^2 \cdot m)$ processors with $O(\log^2 n)$ worst-case time complexity is proposed.

5 Discussion and Suggestions

This paper examines the research methods and objects across various references, revealing that most studies focus on the algorithmic theory of dynamic MIS problems, particularly the gradual optimization of serial time complexity. However, research on optimizing other metrics and applying dynamic MIS remains limited. We therefore propose the following directions for future MIS research:

1. **Optimize constants and space complexity of dynamic MIS algorithms.** Most existing papers focus on time complexity optimization without addressing actual running time or space complexity.
2. **Integrate dynamic MIS algorithms into specific application environments.** With solid theoretical foundations established, future work should examine algorithm performance in practical applications.

3. **Develop parallel algorithms for dynamic MIS.** Given the prevalence of multi-core environments, efficient parallel algorithms may overcome the time complexity bottlenecks of serial approaches.

6 Conclusions

Our analysis reveals that the time complexity of dynamic MIS algorithms has improved continuously in recent years. However, studies on algorithmic applications in specific environments and other aspects remain scarce. This paper offers suggestions for potential future innovations.

The limitation of our work lies in not yet delving deeply into the algorithmic principles of these papers. Through further fine-grained analysis of the theorems and data structures employed by each algorithm, it may be possible to identify and combine their respective advantages, thereby discovering new innovations.

References

- [1] Chechik S, Zhang T. Fully dynamic maximal independent set in expected poly-log update time[C]//2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2019: 370-381.
- [2] Itai A, Babai L, Alon N, A fast and simple randomized parallel algorithm for the maximal independent set problem[J]. Journal of algorithms, 1986, 7(4): 567-583.
- [3] Sepehr Assadi, Krzysztof Onak, Baruch Schieber, and Shay Solomon. Fully dynamic maximal independent set with sublinear update time. In Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018, pages 815-826, 2018.
- [4] Schieber B, Onak K, Assadi S, et al. Fully dynamic maximal independent set with sublinear in n update time[C]//Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2019: 1919-1936.
- [5] Derakhshan M, Behnezhad S, Hajiaghayi M T, et al. Fully dynamic maximal independent set with polylogarithmic update time[C]//2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2019: 382-405.
- [6] Bernstein A, Henzinger M, Forster S. A deamortization approach for dynamic spanner and dynamic maximal matching[J]. ACM Transactions on Algorithms (TALG), 2021, 17(4): 1-51.
- [7] Haramaty E, Censor-Hillel K, Karnin Z. Optimal dynamic distributed MIS[C]//Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing. 2016: 217-226.
- [8] Kuhn F, Daum S, Gilbert S, et al. Leader election in shared spectrum radio networks[C]//Proceedings of the 2012 ACM symposium on Principles of

distributed computing. 2012: 215-224.

[9] Zhang H, Du Y. Improved algorithms for fully dynamic maximal independent set[J]. arXiv preprint arXiv:1804.08908, 2018.

[10] Gupta M, Khan S. Simple dynamic algorithms for maximal independent set and other problems[J]. arXiv preprint arXiv:1804.01823, 2018.

[11] Linial N. Distributive graph algorithms global solutions from local data[C]//28th Annual Symposium on Foundations of Computer Science (sfcs 1987). IEEE, 1987: 331-335.

[12] Luby M. A simple parallel algorithm for the maximal independent set problem[J]. SIAM journal on computing, 1986, 15(4): 1036-1053.

[13] Schieber B, Onak K, Solomon S, et al. Fully dynamic MIS in uniformly sparse graphs[J]. arXiv preprint arXiv:1808.10316, 2018.

[14] Yu D, Hua Q S, Wang Y, et al. Distributed $(\Delta + 1)$ -coloring in the physical model[J]. Theoretical Computer Science, 2014, 553: 37-56.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.