*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 26, Issue 6, Ser. 3 (Nov. – Dec. 2024), PP 25-35*
*www.iosrjournals.org*

# Efficient Approximation Algorithm For Hypergraph Coloring

Rahul Pitale[1], Kapil Tajane[1], Aditya Agre[1], Amay Chandravanshi[1], Siddhi Bajpai[1], Yash Athawale[1]
*Department Of Computer Engineering, Pimpri Chinchwad College Of Engineering, Pune, India*

***Abstract:***
*This research paper centrally focuses on hypergraphs, a subdomain in theoretical computer science, investigating its integration with diverse domains. We provide a survey of the algorithms that have been previously proposed in the literature, as well as propose a novel algorithm for hypergraph coloring as well, and compare the results obtained with the State Of The Art (SOTA) algorithm. The algorithm that we propose in this paper can provide a sufficiently accurate hyper- chromatic number of a hypergraph in a O($n^2$) time complexity.*
***Key Word:*** *Algorithms, Hypergraph, Coloring, Edge Coloring, Vertex Coloring*

---

Date of Submission: 11-12-2024 Date of Acceptance: 21-12-2024

---

## I. Introduction

Hypergraph coloring is a specialized area within theoretical computer science that deals with the coloring of hypergraphs, which are generalizations of graphs where edges can connect more than two vertices. In a hypergraph, an edge can connect any number of vertices, leading to a richer structure compared to traditional graphs. Hypergraph coloring involves assigning colors to the vertices of a hypergraph in such a way that no two vertices sharing the same hyperedge have the same color. The hypergraph coloring problem is an extension of the classic graph coloring problem to hypergraphs. In a hypergraph, edges can connect more than two vertices, allowing for more complex relationships between vertices. Formally, given a hypergraph H = (V, E), where V is the set of vertices and E is the set of hyperedges, a proper hypergraph coloring is an assignment of colors to vertices such that no hyperedge contains vertices of the same color. More formally, let k be the number of colors available for coloring. A proper hypergraph coloring is a function $f: V \to \{1, 2, . . ., k\}$ such that for every hyperedge E, there exist at least two distinct vertices v1, v2 such that f (v1) = f (v2). The goal of the hypergraph coloring problem is to find the minimum number of colors needed for a proper coloring, which is known as the chromatic number of the hypergraph. The study of hypergraph coloring is motivated by various real-world applications, such as scheduling problems, resource allocation, and data clustering. For example, in scheduling problems, hypergraph coloring can be used to model tasks and resources, where each hyperedge represents a set of tasks that must be executed simultaneously, and each vertex represents a resource that can perform those tasks. Efficient coloring algorithms for hypergraphs can help optimize resource allocation and improve the overall efficiency of scheduling. One of the fundamental problems in hypergraph coloring is determining the chromatic number of a hypergraph, which is the minimum number of colors required to color its vertices such that no two vertices sharing a hyperedge have the same color. Unlike traditional graph coloring, finding the chromatic number of a hypergraph is NP-hard, making it a challenging problem from a computational perspective. To address the complexity of hypergraph coloring, researchers have developed various approximation algorithms and heuristics that aim to find reasonably good colorings in polynomial time. These algorithms often rely on techniques such as greedy coloring, local search, and metaheuristic optimization to iteratively improve the quality of the coloring. While these algorithms may not always guarantee optimal solutions, they provide practical approaches for coloring large hypergraphs efficiently. Another area of research in hypergraph coloring is the study of structural properties of hypergraphs that can impact their coloring complexity. For example, researchers have investigated properties such as hypergraph connectivity, hypergraph degree, and hypergraph regularity, which can provide insights into the hardness of coloring problems and guide the development of more efficient algorithms. In recent years, there has been growing interest in applying hypergraph coloring techniques to emerging domains such as machine learning and data mining. Hypergraphs can be used to represent complex relationships and dependencies in data, and coloring algorithms can help uncover meaningful patterns and structures within large datasets. By leveraging the power of hypergraph coloring, re- searchers aim to develop innovative solutions for tasks such as community detection, clustering, and pattern recognition. We first provide a necessary background for understanding of this paper. In the 3rd section, we go over the past work that has been done so far in regards to this problem. In the 4th section, we propose a novel approach for hypergraph coloring. In the last section of the paper, we delve into Hypergraphs and its applications.

## II. Background

In the domain of TCS, a graph is represented as a mathematical set G = (V, E), with V denoting the set of vertices and E representing the set of edges. Hypergraphs, on the other hand, are non-linear data structures that extend and generalize the concept of graphs. A hypergraph is defined as a pair H = (V, E), where V is a finite set of elements known as vertices, and E comprises a family of subsets of V referred to as edges or hyperedges. Essentially, a hypergraph can be visualized as a grouping of vertices, in contrast to a simple line connecting two vertices as seen in traditional graphs.

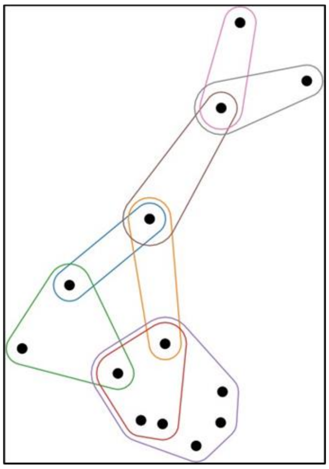

**Figure 1: Denotes a hypergraph**

A hypergraph, as a generalization of a graph, introduces hyperedges that can connect more than two vertices, unlike traditional graphs where edges connect pairs of vertices. Hyperedges in a hyper- graph can contain any number of vertices, expanding the structural complexity beyond the constraints of graphs. Hypergraph coloring addresses the fundamental problem of assigning colors to vertices in such a way that no hyperedge contains vertices of the same color. The minimum number of colors required for a valid coloring is termed the chromatic number of the hypergraph. Proper edge-coloring, a variant of hypergraph coloring, involves assigning colors to hypergraph edges such that no two edges of the same color share a vertex. In the context of hypergraph coloring, the concept of strong coloring emerges, allowing vertices to possess multiple colors, aiming to minimize the total number of colors utilized. The Lov´asz Local Lemma, a potent probabilistic technique, finds application in proving the existence of combinatorial objects, often employed in hypergraph coloring to establish bounds on the number of colorings and analyze hypergraph properties. Composition width serves as a parameter characterizing the complexity of hypergraphs, significantly influencing the difficulty of coloring problems and the performance of approximation algorithms. Ramsey numbers, representing the smallest values where specific combinatorial structures contain certain substructures, have implications for hypergraph coloring bounds. Nonrepetitive coloring, a coloring scheme ensuring that adjacent vertices (or edges) do not share the same color, is pertinent to both hypergraph coloring and graph theory. Fru- gal coloring, on the other hand, aims to minimize the total number of colors used, finding application in various hypergraph coloring scenarios. Moreover, in the realm of Boolean satisfiability problems, k-SAT involves logical formulas with k literals per clause. Hypergraph coloring techniques prove relevant for solving k-SAT instances, showcasing the interdisciplinary nature of hypergraph coloring and its applications in computational problem-solving.

## III. Literature Survey

Coloring a graph with the fewest colors possible is a well-known NP-hard problem, even when restricted to graphs that can be colored with a constant number of colors (k-colorable graphs for a constant k > 3). In 1991 A. Blum [27] presented a thesis addressing the approximation problem of coloring k-colorable graphs, with a focus on k = 3. It introduces an algorithm that efficiently colors 3-colorable graphs with improved bounds. The research extends to worst-case bounds for k-colorable graphs k > 3, and explores the coloring of random k-colorable graphs and semi-random graphs. Additionally, the thesis establishes lower bounds on the difficulty of approximate graph coloring and its implications for other computational problems.

A. Blum [22] in 1994, investigated the approximation problem of coloring k-colorable graphs with the minimum number of additional colors in polynomial time. The previous best upper bound for polynomial-time coloring of n-vertex 3-colorable graphs was O(n) colors, as established by Berger and Rompel et al, improving upon a bound of O(n) colors by Wigderson. This paper introduces an algo- rithm that can color any 3-colorable graph with O(n3/8)polylog(n)) colors, surpassing the ”O(n(1/2- o(1)))” barrier. The algorithm proposed here is based on examining second-order neighborhoods of vertices, a departure from previous approaches that only considered immediate neighborhoods of ver- tices. Furthermore, the results are extended to enhance the worst-case bounds for coloring k-colorable graphs for constant k > 3. Avi Wigderson [24] in 1983 introduced a new graph coloring algorithm with a performance guarantee of O(n(loglogn)2/(logn)2), surpassing the previous best-known guarantee of O(n/logn) for graphs on n vertices.

Graph coloring is a fundamental problem with applications in production scheduling and timetable construction. The graph coloring problem is NP-complete, making it challenging to find a polynomial- time algorithm that guarantees optimal coloring. They presented algorithms (A, B, C) with improved performance guarantees and practical implementations, addressing the gap between NP-hardness and existing polynomial-time guarantees. The hybrid algorithm (E) combines Algorithm C with a previous one, achieving a guarantee of O(n(loglogn)2/log3n). Halld´orsson and Magnu´s M. [25] enhanced the previously known best performance guarantee by employing an approximate algorithm for the inde- pendent set problem. The achieved performance guarantee is now expressed as O(n(loglogn)2/log3n). In the paper Approximate hypergraph coloring the authors Kelsen, Mahajan, and Ramesh et al [26] in the year 2006 developed approximation algorithms specifically designed for coloring 2-colorable hypergraphs. The initial outcome is an algorithm capable of coloring any 2-colorable hypergraph with n vertices and dimension d, utilizing O(n(1 1/d)log(1 1/d)n) colors. Notably, this algorithm marks the first instance of achieving a sublinear number of colors within polynomial time. The approach is rooted in a novel technique for reducing degrees in a hypergraph, which holds independent significance. For the particular scenario of hypergraphs with a dimension of three, the authors enhanced the previous result by introducing an algorithm that employs only O(n(2/9)log(17/8)n) colors. This achievement relies significantly on the utilization of semidefinite programming. The paper Approximate Coloring of Uniform Hypergraphs presented by Krivelevich, Michael Sudakov, Benny[19] in 1998 examines the algorithmic challenge of coloring r-uniform hypergraphs.

Since determining the exact chromatic number of a hypergraph is known to be NP-hard, we explore approximate solutions. Through a straightforward construction and leveraging established findings on graph coloring hardness, we demonstrate that, for any r3, it is computationally infeasible to approximate the chromatic number of r-uniform hypergraphs with n vertices within a factor of n(1 ) for any

> 0, unless NP is a subset of ZPP. On a positive note, we introduce an approximation algorithm for coloring r-uniform hypergraphs with n vertices, achieving a performance ratio of O(n(loglogn)2/(logn)2). Additionally, we outline an algorithm for coloring 3-uniform 2-colorable hypergraphs with n vertices using O˜(n(9/41)) colors, thereby surpassing the prior results of Chen and Frieze as well as Kelsen, Mahajan, and Ramesh.

We now discuss a few well established primitives to hypergraphs. In the initial findings, a partial Steiner system characterized by parameters (t, k, n) is a hypergraph with n vertices and k-uniformity, where any collection of t vertices is found in only one edge, while a full Steiner system with the same parameters ensures that each set of t vertices is precisely contained in a single edge. It’s important to note that a Steiner system with parameters (t, k, n) consists of n choose t times k choose t edges, implying k choose t times n choose t. The Existence conjecture for designs, often called the Steiner system Existence conjecture, suggests that, with only a few exceptions, these divisibility conditions are adequate to guarantee the existence of a Steiner system with parameters (t, k, n). In 1963, Erdos and Hanani raised the question of an approximate version of this conjecture, which Rodl confirmed in 1985, introducing the well-known ’nibble’ method.

**Theorem 3.1**: For all $k > t \geq 1$ and $\epsilon > 0$, there cannot exist an n such that the following holds. For all n n0, a partial Steiner system exists with parameters (t, k, n) and at least (1) C(n, t)/C(k, t) edges.[2]

The proof of the Existence conjecture by Keevash, along with alternative combinatorial proofs, demonstrates the intricate interplay between algebraic and combinatorial methods in extremal combinatorics. The connection between partial Steiner systems and perfect matchings in hypergraphs provides a powerful tool for understanding the existence of certain combinatorial structures. The generalizations and relaxations of conditions by Frankl, R¨odl, and Pippenger highlight the robustness and applicability of these results across a broader class of hypergraphs, emphasizing the flexibility and depth of the underlying mathematical techniques. Overall, these findings contribute to a deeper understanding of structural properties in combinatorics and offer insights into the existence of certain combinatorial configurations under varying conditions.

**Theorem 3.2:** For all k, $\epsilon > 0$, a $\delta > 0$ exists such that the following holds. If H is an n-vertex k-uniform D-regular hypergraph with codegree at most δD, then there is a matching in H covering all but at most n vertices.[2] The observation deduced from the theorem highlights the robust nature of large, k-uniform D-regular hypergraphs. For any positive integers k and (ϵ, the theorem establishes the existence of a positive constant such that if a hypergraph H is n-vertex, k-uniform, D-regular, and exhibits limited codegree (at most D), then there exists a matching in H that covers nearly all vertices, leaving at most ϵ n uncovered. This implies a high degree of structure and regularity in hypergraphs, ensuring the prevalence of near-perfect matchings despite variations in codegree, contributing to a better understanding of their combinatorial properties.

**Theorem 3.3:** For every k, $\epsilon > 0$, there exists > 0 such that the following holds. If H is a k-uniform D-regular hypergraph of codegree at most D, then [3] $\chi'(H)(1 + \epsilon)D$. The relationship between hypergraph properties is highlighted through the inequality (H) H /$\chi 0$(H), where, for a D regular and k-uniform hypergraph H, the conditions |H| = D|V (H)| hold.

Pippenger and Spencer's proof of a result akin to Theorem 3.3 involves the random selection of nearly perfect matchings using the nibble process. Through the iterative choice of D such matchings in a semi-random manner, they demonstrate that the remaining hypergraph exhibits a small maximum degree, allowing for a proper edge-coloring with at most D colors in a greedy fashion. This proof is further strengthened by Kahn's observation that the generality of Theorem 3.3 extends to k-bounded hypergraphs with a maximum degree at most D, as he establishes that such hypergraphs can be embedded in nearly D-regular k-uniform hypergraphs with the same or larger chromatic index. This progression of results culminates in Kahn's extension of the Pippenger Spencer theorem to list coloring, broadening the applicability of these findings to various hypergraph structures and coloring scenarios.

**Theorem 3.4:** For every k, $\epsilon > 0$, there exists $\delta > 0$ such that the following holds. If H is a k-bounded hypergraph of maximum degree at most D and codegree at most δD, then
$\chi' l(H)(1 + \epsilon)D$

The observation derived from the theorem shows the remarkable chromatic property of k-uniform D-regular hypergraphs. For any positive integers k and ϵ, the theorem establishes the existence of a positive constant δ. It asserts that if a hypergraph H is k-uniform, D-regular, and has a codegree

not exceeding δD, then its chromatic index $\chi'(H)$ is bounded by $(1 + \epsilon)D$. In simpler terms, this implies that despite potential irregularities in the hypergraph's codegree, its chromatic index can be nearly as low as the regularity parameter D increased by a small factor. This insight contributes to understanding the chromatic behavior of hypergraphs under certain structural conditions.

**Conjecture 3.1:** For every k, $\epsilon > 0$, there exists K such that the following holds. If H is a k-bounded multi-hypergraph, then [4]
$\chi' l(H) \max(1 + \epsilon)\chi' f\,(H), K$

The conjecture regarding the list chromatic index of hypergraphs remains wide open, especially in its weaker version where the list chromatic index is replaced by the chromatic index, with the exception of the known case when k = 2. The case for 2-bounded hypergraphs, corresponding to graphs with edge- multiplicity 1, is established through Vizing's theorem for the chromatic index and Theorem 3.4 for the list chromatic index. Seymour, employing Edmonds' Matching Polytope theorem, demonstrated that every multigraph G satisfies $\chi_0 f$ (G) = max{Δ(G), Γ(G)}, confirming Conjecture 3.5 for k = 2. Kahn extended this result to list coloring, asymptotically confirming the conjecture for k = 2. In the context of the ordinary chromatic index, Kahn's work asymptotically confirmed the Goldberg-Seymour conjecture, originally proposed by Goldberg and Seymour in the 1970s, which posits that every multigraph G satisfies
$\chi 0(G) \leq \max\{\Delta(G) + 1, d\Gamma(G)\}$ d (G) = max 2|E(H)||V (H)| − 1: H ⊆ G, |V (H)| ≥ 3.

Recently, Chen, Jing, and Zang provided a comprehensive proof of the Goldberg-Seymour conjecture, eliminating reliance on probabilistic arguments. Additionally, the text introduces a conjecture by Alon and Kim regarding t-simple hypergraphs, defined by having every two distinct edges with at most t common vertices, where the special case of 1- simple corresponds to linear hypergraphs.

**Conjecture 3.2:** For every $k \geq t \geq 1$ and $\epsilon > 0$, there exists D0 such that the following holds. For every D ≥ (D0, if H is a )k-uniform, t-simple multi-hypergraph with maximum degree at most D, k-uniform, t-simple multi-hypergraphs, there's a limit to the number of colors needed to paint the edges. For any positive values of k, t, and ϵ, it claims that if the maximum number of connections any point has (known as the maximum degree) is more than a certain amount D0, then you can color the edges using a number of colors related to t and ϵ. This indicates that as things get more connected beyond a certain point, predicting the number of colors needed becomes simpler.

If we consider a hypergraph H with k-uniformity and D-regularity on n vertices. Pippenger's theorem indicates that if the co-degree of H is small compared to D, there exists a matching covering nearly all but a few vertices, although the proof lacks a specific estimate for the error term. Additionally, certain theorems suggest that a small codegree implies the chromatic index of H is close to

D. Advancements, particularly with improved analysis and variations of the nibble method, aim to refine these error terms, making them more precise for various applications. Grable's work shows that a limited codegree leads to a matching covering almost all but a specific number of vertices. In 1997, Alon, Kim, and Spencer enhanced this bound for linear hypergraphs by showing the following.

**Theorem 3.5**: Let k ≥ 3. Let H be a k-uniform D-regular n-vertex linear hypergraph. Then H has a matching containing all but at most O(nD− 1/logck D) vertices, where c = 0 for k > 3 and c3 = 3 . [6]

The text discusses conjectures and results related to the simple random greedy algorithm used in computer simulations for generating matchings in hypergraphs. Based on simulations, Alon, Kim, and Spencer conjectured that this algorithm should yield a matching containing all but at most

1

O(nD k−1 log

O(1/D) vertices. However, existing results, including those by Spencer, R¨odl, Thoma, Wormald, and Bennett and Bohman, have only shown that the random greedy algorithm produces a matching covering nearly all but o(n) vertices, with efforts to refine the error term. Kostochka and R¨odl extended a particular theorem to hypergraphs with small codegrees. Vu further extended this by removing certain assumptions on the codegree. Kang, Ku¨hn, Methuku, and Osthus recently improved upon these results for hypergraphs with small codegree, specifically in the case of linear hypergraphs. The refined theorem states that a linear hypergraph with certain properties has a matching containing all but at most O (n - 1 / k−1 logc D) vertices for some constant c > 0, demonstrating progress in understanding matchings in hypergraphs with varying degrees of codegree complexity.

**Theorem 3.6:** Let k > 3 and let 0 < γ, μ < 1 and 0 < η <k−3(k−1)(k3 −2k2 −k+4). Then there exists n0 = n0(k, γ, η, μ) such that the following holds for n n0 and D exp(logμ n). If H is a k-uniform D-regular linear hypergraph on n vertices, then H contains a matching covering all but at most nD− k−1 −η vertices. [7]

The text describes an approach involving the R¨odl nibble process, which not only constructs a substantial matching in hypergraphs but also generates well-distributed 'augmenting stars.' These augmenting stars play a crucial role in significantly enhancing the matching produced by the R¨odl nibble process. Shifting focus to improvements on the chromatic index of hypergraphs, Molloy and Reed made notable progress in 2000 by refining the error term in Theorem 3.4. Specifically, for linear hypergraphs, their result is characterized by a sharpened error term, contributing to a more precise understanding of the chromatic index in this context.

**Theorem 3.7:** If H is a k-uniform linear hypergraph with maximum degree at most D, then

χ′(H) ≤ D + O D1− 1 log4 D . [8]

index for k-uniform hypergraphs. Molloy and Reed's work in 2000 improved upon a result by H˚astad, H˚aggkvist, and Janssen, offering the best-known general bound for the List Edge Coloring conjecture and a refined bound for the ordinary chromatic index. Their more general result states that any k- uniform hypergraph H with a maximum degree at most D and co-degree at most C has a list chromatic index at most D + O D klogD /4, which also provides the best-known bound for the ordinary chromatic index χ0(H). Recently, Kang, Ku¨hn, Methuku, and Osthus further improved this bound, specifically for linear hypergraphs, demonstrating ongoing progress in understanding chromatic indices for hypergraphs with varying degrees of uniformity and complexity.

**Conjecture 3.3**: Every Steiner triple system with n vertices has a matching of size at least n−4 . [9] Recently, a breakthrough by Keevash, Pokrovskiy, Sudakov, and Yepremyan combined the nibble method with the robust expansion properties of edge-colored pseudorandom graphs to show that every Steiner triple system has a matching covering nearly all but at most O log n vertices.

This result addresses a related problem, specifically the well-known conjecture by Ryser, Brualdi, and Stein, asserting that every n n Latin square should have a transversal of order n 1 and, for odd n, a full transversal. The best-known bound for this Latin square problem, demonstrated by the same authors, is that every n × n Latin square contains a transversal of order n − O log n.

**Conjecture 3.4:** If H is a Steiner triple system with n > 7 vertices, then χ′(H) ≤ n−1 + 3 and moreover, if n ≡ 3 (mod 6), then χ′(H) ≤ n−1 + 2. [10]

Given that an n-vertex Steiner triple system is n−1 -regular, it's evident that χ0(H) is at least n−1 , and this holds true only if H can be broken down into perfect matchings. Therefore, when n ≡ 1 (mod 6), χ0(H) is at least n+1 . In fact, constructions of Steiner triple systems with n vertices illustrate that Conjecture 3.4, if correct, is precisely accurate. Similarly, for Latin squares, a conjecture was independently proposed by Cavenagh and Kuhl in 2015 and by Besharati, Goddyn, Mahmoodian, and Mortezaeefarbeen in 2016.

**Conjecture 3.5:** Let L be an n n Latin square. If HL is the corresponding 3-uniform 3- partite hypergraph, then χ′(HL) n + 2 and moreover, if n is odd, then χ′(HL) n + 1. [2] The text highlights the relationships between several conjectures in combinatorics. Conjecture 3.4 implies Conjecture 3.3, and Conjecture 3.5 implies the well-known Ryser-Brualdi-Stein conjecture. Every n- vertex Steiner triple system has a chromatic index at most n +O n 2 − 1 , and correspondingly, every hypergraph associated with an n × n Latin square has a chromatic index at most n + O ($n^2$) .

These results currently represent the best-known bounds for these problems, offering insights into the chromatic properties of Steiner triple systems and hypergraphs derived from Latin squares in the realm of combinatorics.

**Theorem 3.6:** There exists an absolute constant c > 0 such that the following holds. If G is an n-vertex triangle-free graph of average degree at most d, then α(G) ≥ c n log d. The findings presented in Theorem 3.6 and its hypergraph counterpart by Koml´os, Pintz, Spencer, and Szemer´edi have spurred extensive research spanning four decades. These results, with unexpected applications in number theory and geometry, have become pivotal in combinatorics. Enhancing and extending Theorem 3.6 remains a crucial challenge, given its profound connections to Ramsey theory, random graphs, and algorithmic studies. The ongoing exploration of these theorems reflects their significant impact and the depth of their implications in various mathematical domains, marking them as central pursuits in the realm of combinatorial research.

**Theorem 3.7:** For every ϵ > 0, there exists Δ0 such that the following holds for ev(ery Δ ≥ Δ0.Shearer's bound for regular graphs presents a major open problem in improving the leading constant or determining its optimality. Frieze and Luczak's result for random Δ-regular graphs, stating that they have chromatic number 1 ± o(1) Δ with high probability, raises the open question of whether a polynomial-time algorithm exists to almost surely find a proper vertex-coloring with at most (1 ϵ) Δ colors for some ϵ > 0. The possibility of such an algorithm is linked to the problem of coloring triangle-free graphs of maximum degree at most Δ. Kim and Johansson's proofs, using a nibble approach inspired by Kahn's proof of Theorem 3.4, have been simplified by Molloy and Reed, with further simplifications by Bernshteyn. However, Bernshteyn's proof is non-constructive, while Molloy's 'entropy compression' method provides an efficient randomized algorithm for proper coloring, matching the 'algorithmic barrier' for coloring random graphs. These proofs rely on a 'coupon collector'-type approach, and it is believed that a similar result holds for Kr-free graphs for every fixed r, although with a potentially worse leading constant.

Conjecture 3.6: For every r N , there exists a constant cr such that the following holds. If G is a Kr-free graph with maximum degree at most Δ, then χl(G) cr Δ . [33] The obtained limit on the independence number poses a significant unresolved challenge, initially proposed by Ajtai, Erd˝os, Koml´os, and Szemer´edi. Remarkably, even for r = 4, the problem remains unsolved. Similarly, the conjecture on the chromatic number, suggested by Alon, Krivelevich, and Sudakov, is yet to be proven. Johansson's contributions to this domain include proving that for any fixed r, a Kr-free graph with a maximum degree of at most Δ has a list chromatic number of O Δ log log Δ . These findings, initially unpublished, gained new proof through Molloy and Bonamy, Kelly, Nelson, and Postle. Alon, Krivelevich, and Sudakov further extended Johansson's results to 'locally sparse graphs,' introducing complexities based on the neighborhood of vertices. This progression was generalized to list coloring by Vu. Davies, Kang, Pirot, and Sereni later enhanced this result, demonstrating its validity with a leading constant of 1 + o(1) as the parameter f approaches infinity, thereby expanding the scope of the theorem.

**Theorem 3.9:** For every ϵ > 0, there exists Δ0 such that the following holds for every Δ Δ0.

If G is a graph of maximum degree at most Δ such that the neighborhood of any vertex spans at most Δ2/f edges for f ≤ Δ2 + 1, then χl(G) ≤ (1 + ϵ) ( Δ√ . [14] The text discusses the utilization of the nibble method in various results by Kim, Johansson, and Vu. Davies, Kang, Pirot, and Sereni presented a generalized approach called the 'local occupancy method,' which extends the nibble method results and introduces optimization problems related to the 'hard-core model'—a family of probability distributions over the independent sets of a graph. Their method, inspired by Molloy and Bernshteyn's work, relies on the Lopsided Local Lemma, and it connects to previous results bounding the average size of independent sets. The main outcome of Davies, Kang, Pirot, and Sereni is proven using the Lopsided Local Lemma or entropy compression, offering additional algorithmic coloring results. All the results in this subsection provide chromatic number bounds with a local sparsity condition, restricting the chromatic number away from Δ. The text highlights the classic result by Brooks, stating that equality χ(G) = Δ(G) + 1 holds only for complete graphs or odd cycles. Even with a relaxed local sparsity condition, the chromatic number can be bounded away from Δ, as demonstrated in the presented result.

**Theorem 3.10**: There exists K such that the following holds. If G is a bipartite graph of maxi- mum degree at most Δ, then χl(G) K log Δ. [15] The prevailing conjecture, highlighted by Theorem, remains a challenging problem with the most recognized boundary. Interestingly, this boundary can be more directly demonstrated using the 'coupon collector' approach discussed earlier. Alon, Cambie, and Kang employed this method to establish a more robust outcome for list coloring bipartite graphs, particularly when each vertex in one part possesses a list of colors of the conjectured size. Alon and Krivelevich proposed an even stronger bound, χ′(G) (1 + o(1)) log2 Δ, which could be optimal for complete bipartite graphs. Saxton and Thomason subsequently enhanced our understanding, demon-strating that every graph with a minimum degree of at least d necessitates a list chromatic number of at least (o(1)) log2 d, surpassing a previous result by Alon. These endeavors contribute to unrav- eling the intricacies of list coloring and refining conjectures in graph theory. Theorems in hypergraph theory, notably by Keevash and others, lay the groundwork for understanding hypergraph structures. Transitioning to hypergraph coloring, these theorems become crucial. They provide a foundation for addressing challenges in efficiently assigning colors to hypergraph vertices, with broader implications for computational efficiency and algorithmic design.

**Theorem 3.11**: For all k, there exist constants c and Δ0 > 0 such that the following statement holds for every Δ . If H is a k-uniform hypergraph with maximum degree at most Frieze and Mubayi's analysis of a nibble procedure, inspired by Johansson's proof, led to the proof of a result regarding linear hypergraphs, extending it from graphs. The Molloy conjecture states that for k = 3, the result holds for c = 2 + o(1). This is based on the 'coupon collector' heuristic.has a girth of at least five. Frieze and Mubayi extended the result to include linear hypergraphs for k 3 by applying it to vertex-disjoint induced subgraphs. Where the girth is at least four. Cooper and Mubayi further generalized this for k = 3 by replacing the girth condition with the absence of triangles in the hypergraph. Frieze and Mubayi conjectured a generalization of Conjecture 3.3 for k-uniform hypergraphs. This conjecture was later disproved by Cooper and Mubayi for all k, providing insights into the complexities of hypergraph coloring.

## IV. Proposed Algorithm

This paper aims to present an algorithm to tackle hypergraph colouring. Given that this is an NP Complete problem, it is obvious that the worst case complexity of any such algorithm would be exponential. However, the worst cases for colouring hyper-graphs arise when there are too many vertices that can be coloured with the same colour and choices need to be made so as to find the least number of colours required (chromatic number). Brute force methods, tend to check each combination one by one without prioritising any vertex over another. This section presents an algorithm that prioritises colouring vertices with higher degrees first. For instance, if a particular colour can be repeated for two vertices, brute force strategy suggests checking both cases one by one and completing rest of the colouring. Both the outcomes are analysed and then the better one is chosen. This has very high computational complexity. The presented algorithm sacrifices optimum colouring for colouring the hyper-graph in reasonable time. The algorithm begins by sorting the given hypergraph vertices in descending order of degrees. A colour counter is set, indicating the number of colours used yet. We start off with the number of colours as zero. We colour the first uncoloured vertex (according to sorted degree) with the colour '1'. We maintain a list of vertices with the same colour. Thereafter, we look for the next most connected vertex (in order) that is compatible with our current vertices with this colour that we have stored in the list mentioned above. Compatibility here can be defined as follows: let
VertexA = [10101]
VertexB = [01100]

Here, both vertices are a part of the third edge. Hence, a colour used for A cannot be used for B. However, If the vertices were of the following nature:
VertexA = [10101]
VertexB = [01000]

None of the edges that contain A, also contain B. Therefore, the same colours may be used for both. Mathematically, this can be checked as, Summation (Ea*Ba) If this value is zero for a pair of vertices, then and only then can they be considered compatible. Now, once the algorithm comes across such a vertex that is compatible with all the vertices already assigned a particular colour, that colour is readily assigned to that vertex. This vertex is now added to the list maintaining the record of all the vertices with the same colour. Now the algorithm moves forward to check the next vertex. In case it reaches the end of the hypergraph, and colouring has not been completed, a new colour '2' will be taken up. The list of vertices with the same colour will be reset to an empty list. The process is so repeated till all the vertices are coloured. Since, each colour used up can iterate at most once over the hypergraph matrix, the traversal will always have complexity of the order O(colours). However, we know that the number of colours needed to colour a hypergraph will always be less than or equal to the number of total vertices. Hence, this traversal will take complexity of the order O(n). Since we sort the hypergraph based on degrees in the beginning, the calculation of degrees requires a O(n2) complexity. Hence, the overall time complexity is of the order O(n2). In a few cases, the colouring returned by the algorithm may not be the optimal solution. For instance, hypergraph may have chromatic number as 10, while the algorithm might return 11. In such cases, the solution returned is still a proper solution to the hypergraph colouring problem, however, not the optimal solution. For such cases, the following error function has been defined: Xo denotes the chromatic number of a hypergraph. X' denotes the number of colours used for colouring by the algorithm.

The following is the pseudocode for the algorithm:
Proper_coloring(hypergraph_object):
# Function to do complete proper coloring for a given hypergraph

if hypergraph_object.ongoing_color == False:
# If no ongoing color, assign a new color to an uncolored vertex for vertex in hypergraph_object.vertices: if vertex is uncolored:
# Assign a new color to this vertex hypergraph_ object. colors_ used += 1 hypergraph_ object. current_ color = hypergraph_ object. colors_ used hypergraph_ object. current_ color_ vertices = [] hypergraph_ object. repeatable_ list_ clear()
hypergraph_ object. assign_ color (vertex, hypergraph_ object. current_ color) hypergraph_ object. latest_ colored_ vertex = vertex hypergraph_ object. current_ color_ vertices. Append (vertex) hypergraph_ object. ongoing_ color = True
break

if hypergraph_ object. ongoing_ color == True: Proper_ coloring (hypergraph_ object)

else:
# Ongoing color is true, look for more vertices to color rep_list = hypergraph_object.repeatable_list()
if len(rep_list) == 0:
# No more repetitions possible, restart the coloring process Proper_coloring(hypergraph_object)

elif len(rep_list) == 1:
# Only one vertex to color, color it and continue hypergraph_object.assign_color(rep_list[0], hypergraph_object.current_color) Proper_coloring(hypergraph_object)

else:

# Multiple vertices to choose from, find the best one to color comparison_array = []

for vertex in rep_list:
hypergraph_object.assign_color(vertex, hypergraph_object.current_color)

leftover_hypergraph = hypergraph_object.uncolored_sub_hypergraph() colored_hypergraph = hypergraph_object.colored_sub_hypergraph()

hypergraph_object_new = colored_hypergraph + Proper_coloring(leftover_hypergraph) comparison_array.append(hypergraph_object_new)

final_hypergraph = comparison_array[0] for hypergraph in comparison_array:
if hypergraph.return_chromatic_number() < final_hypergraph.return_chromatic_number(): final_hypergraph = hypergraph

return final_hypergraph

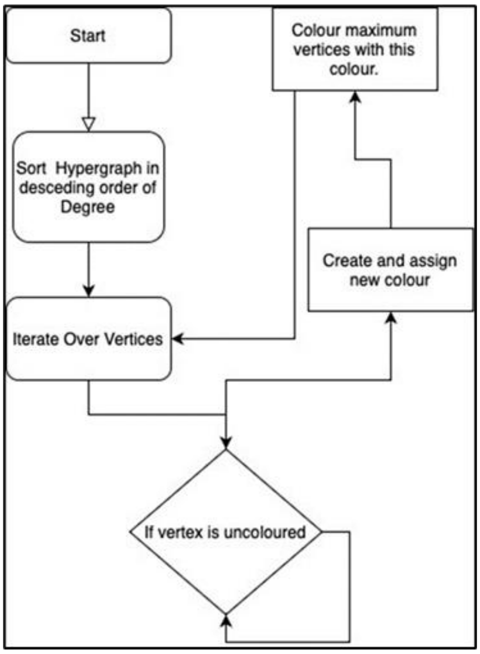


**Figure 2: Flowchart for the algorithm**

## V. Results

We tested the algorithm proposed in this paper on the following synthetic datasets, and obtained the following results:

For all 4x3 hypergraphs: Number of hypergraphs tested: 24∗3 = 4096

| Total Instances | Correct Instances | Mean | Accuracy | Relative Error |
|---|---|---|---|---|
| 4096 | 1738 | 0.32 | 57.56 | 0.212 |

For all 4x4 hypergraphs: Number of hypergraphs tested: 24∗4 = 65536

| Total Instances | Correct Instances | Mean | Accuracy | Relative Error |
|---|---|---|---|---|
| 65536 | 21354 | 0.76 | 32.58 | 0.384 |

For all 5x4 hypergraphs: Number of hypergraphs tested: 25∗4 = 1, 048, 576

| Total Instances | Correct Instances | Mean | Accuracy | Relative Error |
|---|---|---|---|---|
| 1048576 | 158754 | 1.25 | 15.14 | 0.934 |

The decrease in accuracy and increase in relative error as the hypergraph size grows indicate that the model may not scale well with the complexity of the data. The increase in mean correctness alongside decreasing accuracy suggests a disconnect between the model's confidence in its predictions and its actual performance. The rising relative error points to increasing uncertainty and prediction error as hypergraph size grows. The decreasing accuracy coupled with the increasing mean error and relative error indicates that there is a scope for further improvement of the algorithm. Following testing algorithm was used, that uses Backtracking strategy.

```
def is valid_ coloring(hypergraph, colors, hyperedge, color): for i in range(len(hypergraph[hyperedge])):
if hypergraph[hyperedge][i] ==  1: for j in range(len(hypergraph)):
if j != hyperedge and hypergraph[j][i] == 1 and colors[j] == color: return False
return True

def backtrack_coloring(hypergraph, colors, hyperedge,  max_colors): if hyperedge == len(hypergraph):
return True

for color in range(1,  max_colors  +  1):
if is_valid_coloring(hypergraph, colors, hyperedge, color): colors[hyperedge] = color
if backtrack_coloring(hypergraph, colors, hyperedge + 1, max_colors): return True
colors[hyperedge] = 0 return False
```

def find_chromatic_index(hypergraph): num_hyperedges = len(hypergraph) max_colors = num_hyperedges colors = [0] * num_hyperedges

for c in range(1, max_colors + 1):
if backtrack_coloring(hypergraph, colors, 0, c): return c, colors

return max_colors, colors

## VI. Conclusion

In this paper, we discuss Graphs and Hypergraphs. The research primarily focuses on the complexities of graph theory, including structural properties, optimization challenges, and the subtle intricacies of hypergraph coloring. We present a thorough survey of existing methods, highlighting their strengths and limitations in practical scenarios. Furthermore, we propose a novel method to solve the hypergraph coloring problem with a time complexity of O(n2), offering a unique approach that distinguishes itself from traditional algorithms. This proposed method has the potential to improve efficiency and accuracy in solving complex hypergraph coloring issues, with possible implications for fields such as network theory, combinatorics, and computational optimization.

## References

[1] Piotr Formanowicz And Krzysztof Tana´S, ”A Survey Of Graph Coloring - Its Types, Methods And Applications,” Foundations Of Computing And Decision Sciences, Vol.37, No.3, 2012, Pp.223-238. Https://Doi.Org/10.2478/V10209-011-0012-Y.
[2] V. R¨Odl And L. Thoma, ”Asymptotic Packing And The Random Greedy Algorithm,” Random Struc- Tures Algorithms, Vol. 8, 1996, Pp. 161–177.
[3] N. Pippenger And J. Spencer, ”Asymptotic Behavior Of The Chromatic Index For Hypergraphs,” J. Combin. Theory Ser. A, Vol. 51, 1989, Pp. 24–42.
[4] ”Asymptotically Good List-Colorings,” J. Combin. Theory Ser. A, Vol. 73, 1996, Pp. 1–59.
[5] N. Alon And J. H. Kim, ”On The Degree, Size, And Chromatic Index Of A Uniform Hypergraph,” J. Combin. Theory Ser. A, Vol. 77, 1997, Pp. 165–170.
[6] N. Alon, J.-H. Kim, And J. Spencer, ”Nearly Perfect Matchings In Regular Simple Hypergraphs,” Israel J. Math., Vol. 100, 1997, Pp. 171–187.
[7] D. Kang, D. Ku¨Hn, A. Methuku, And D. Osthus, ”New Bounds On The Size Of Nearly Perfect Matchings In Almost Regular Hypergraphs,” Arxiv Preprint Arxiv:2010.04183, 2020.
[8] M. Molloy And B. Reed, ”Near-Optimal List Colorings,” Random Structures Algorithms, Vol. 17, 2000, Pp. 376–402.
[9] A. E. Brouwer, ”On The Size Of A Maximum Transversal In A Steiner Triple System,” Can. J. Math., Vol. 33, 1981, Pp. 1202–1204.
[10] M. Meszka, R. Nedela, And A. Rosa, ”Circulants And The Chromatic Index Of Steiner Triple Sys- Tems,” Math. Slovaca, Vol. 56, 2006, Pp. 371–378.
[11] M. Ajtai, J. Koml´Os, And E. Szemer´Edi, ”A Note On Ramsey Numbers,” J. Combin. Theory Ser. A, Vol. 29, 1980, Pp. 354–360.
[12] ”A Dense Infinite Sidon Sequence,” European J. Combin., Vol. 2, 1981, Pp. 1–11.
[13] ”The List Chromatic Number Of Graphs With Small Clique Number,” J. Combin. Theory Ser. B, Vol. 134, 2019, Pp. 264–284.
[14] ”Graph Structure Via Local Occupancy,” Arxiv Preprint Arxiv:2003.14361, 2020.
[15] N. Alon And M. Krivelevich, ”The Choice Number Of Random Bipartite Graphs,” Ann. Comb., Vol. 2, 1998, Pp. 291–297.
[16] ”Coloring Simple Hypergraphs,” J. Combin. Theory Ser. B, Vol. 103, 2013, Pp. 767–794.
[17] D.-Y. Kang, T. Kelly, D. Ku¨Hn, A. Methuku, And D. Osthus, ”Graph And Hypergraph Colouring Via Nibble Methods: A Survey,” Arxiv Preprint Arxiv:2106.13733, 2021.
[18] M. Krivelevich And B. Sudakov, ”Approximate Coloring Of Uniform Hypergraphs (Extended Ab- Stract),” 1998, Pp. 477–489. Doi: 10.1007/3-540-68530-8 40.
[19] D. N. Thakkar And K. N. Kalariya, ”Strong Vertex Covering In Hypergraphs.”
[20] B. Moln´Ar, ”Applications Of Hypergraphs In Informatics: A Survey And Opportunities For Research,” Annales Universitatis Scientiarum Budapestinensis De Rolando E¨Otv¨Os Nominatae Sectio Compu- Tatorica, Vol. 42, 2014, Pp. 261–282.
[21] A. Blum, ”New Approximation Algorithms For Graph Coloring,” J. Acm, Vol. 41, No. 3, 1994, Pp. 470–516. Doi: 10.1145/176584.176586.
[22] ”N. Musliu And M. Schwengerer, ”Algorithm Selection For The Graph Coloring Problem,” 2013, Pp. 389–403. Doi: 10.1007/978-3-642-44973-442.”
[23] A. Wigderson, ”Improving The Performance Guarantee For Approximate Graph Coloring,” J. Acm, Vol. 30, No. 4, 1983, Pp. 729–735. Doi: 10.1145/2157.2158.
[24] M. M. Halld´Orsson, ”A Still Better Performance Guarantee For Approximate Graph Coloring,” Infor- Mation Processing Letters, Vol. 45, No. 1, 1993, Pp. 19–23. Doi: 10.1016/0020-0190(93)90246-6.
[25] P. Kelsen, S. Mahajan, And R. Hariharan, ”Approximate Hypergraph Coloring,” Nordic Journal Of Computing, Vol. 3, 2006, Pp. 41–52. Doi: 10.1007/3-540-61422-2119.
[26] A. Blum, ”Algorithms For Approximate Graph Coloring,” Technical Report, Massachusetts Institute Of Technology, 1991.
[27] M. Henrique Lemos, M. Prates, P. Avelar, And L. Lamb, ”Graph Coloring Meets Deep Learn- Ing: Effective Graph Neural Network Models For Combinatorial Problems,” Arxiv:1903.04598v2 [Cs.Lg], 2019.
[28] M. Krivelevich, R. Nathaniel, And B. Sudakov, ”Approximating Coloring And Maximum Indepen- Dent Sets In 3-Uniform Hypergraphs,” Journal Of Algorithms, Vol. 41, No. 1, 2001, Pp. 99–113. Doi: 10.1007/3-540-68530-840.
[29] D. Du Et Al., ”Geometric Hypergraph Learning For Visual Tracking,” Ieee Transactions On Cy- Bernetics, Vol. 47, 2016, Pp. 4182-4195.

[30] Y. Zhou Et Al., "Reinforcement Learning Based Local Search For Grouping Problems: A Case Study On Graph Coloring," Expert Syst. Appl., Vol. 64, 2016, Pp. 412–422.
[31] Y. Feng, H. You, Z. Zhang, R. Ji, And Y. Gao, "Hypergraph Neural Networks," Aaai Conference On Artificial Intelligence, 2018.
[32] B. Berger And J. Rompel, "A Better Performance Guarantee For Approximate Graph Coloring," Algorithmica, Vol. 5, 1990, Pp. 459–466. Doi: 10.1007/Bf01840398.
[33] V. Faber, "Linear Hypergraph Edge Coloring."
[34] D. K. Thakkar And K. N. Kalariya, "Vertex Covering And Stability In Hypergraphs," Issn 1819- 4966, Vol. 11, No. 1, 2016, Pp. 61-69.
[35] M. Krul And L. Thoma, "2-Colorability Of R-Uniform Hypergraphs."
[36] D. Bauer, "A Survey Of Minimum Rank Problems," Linear Algebra And Its Applications, Vol. 448, 2014, Pp. 470–491.
[37] R. Diestel, "Graph Theory," Fourth Edition, Springer, 2010.
[38] M. R. Garey And D. S. Johnson, "Computers And Intractability: A Guide To The Theory Of Np-Completeness," W. H. Freeman, 1979.
[39] L. Lov´Asz, "On The Shannon Capacity Of A Graph," Ieee Transactions On Information Theory, Vol. 25, No. 1, 1979, Pp. 1-7.
[40] D. B. West, "Introduction To Graph Theory," Prentice Hall, 2000.
[41] A. Zelikovsky, "A Practical Approach To Np-Completeness," Cambridge University Press, 2018.
[42] M. Rescigno And E. A. Rundensteiner, "A Comparison Of Graph Coloring Algorithms," In "Pro- Ceedings Of The 1998 Acm Symposium On Applied Computing," 1998, Pp. 158–164.
[43] J. Schmidt And A. Siegel, "A Regression Model For Predicting Edge Coloring Algorithm Performance," Journal Of Graph Algorithms And Applications, Vol. 23, No. 5, 2019, Pp. 695–715.
[44] S. Even, "Graph Algorithms," Computer Science Press, 1979.
[45] R. Tarjan, "Depth-First Search And Linear Graph Algorithms," Siam Journal On Computing, Vol. 1, No. 2, 1972, Pp. 146-160.
[46] H. H. Hopcroft And R. M. Karp, "An N 2 Algorithm For Maximum Matchings In Bipartite Graphs," Siam Journal On Computing, Vol. 2, No. 4, 1973, Pp. 225-231.