A Graph Coloring Algorithm Based on Minimal-Cost Graph Neural Networks for Efficient Optimization of Complex Network Structures

Dr. Mahaboob Ali

Assistant Professor
Govt. First Grade College for Women, Raichur
Karnataka-India

Abstract:
The applications of the Graph Coloring Problem (GCP) and its optimization capabilities, including its ties to complexity as a base-level NP-hard problem, encompass the problem's scheduling, network design, and resource allocation uses. Even in the presence of complicated, dense networks, traditional methods such as heuristics and greedy meta-algorithms still seem to lag in performance and scalability. Constructing a Cost Function-based Graph Coloring Problem Solving Framework Based on Graph Neural Networks (MCGNN) aims to consolidate the use of GNN and cost-driven optimization approaches. In the preservation of the Macualls' formulated GCP under the influence of the Potts' model, MCGNN GNNs and minimum cost mechanisms select fundamentals to predict. MCGNN performances on benchmark tests scale over the 2-20qu and 20% value under dense networks, ranging in the 1000s, compared to the provided datasets and state-of-the-art methods. By leveraging the known theoretical network optimization mechanics akin to the Weisfeiler Lehman construct, and on predictable routing capabilities, we believe it to be simplified enough to apply to the unsolved dense networks. MCGNN aims to be enhanced in future work, also to tackle dynamic graphs and multi-objective coloring problems.

Keywords: “Graph coloring, Graph Neural Network (GNN), Minimal Cost Graph Neural Networ”.

I.INTRODUCTION

Graph Coloring topic (GCP) has been studied at length by combinatorial optimization researchers because of the theoretical and practical importance of the topic. It entails using the fewest colors possible to color each vertex of a graph uniquely, ensuring that no two neighboring vertices have the same color. Google Cloud Platform has a wide variety of uses across many industries. Among its most notable uses is in wireless spectrum management, an area that is more important in the modern digital age [1]. To maintain high quality in the services offered by several operators and consumers, it is crucial to assign channels in a manner that minimizes interference between them [2]. Register planning in compilers also utilizes the GCP to minimize runtime; here, each register is akin to a "color," and incompatible tasks, represented by "vertices," require separate registers [3]. The same holds for event scheduling; by using GCP, one may avoid scheduling conflicts and benefit from coordinated time management [4].

A wide variety of strategies, from basic greedy algorithms to more complex heuristics like tabu search, simulated annealing, and genetic algorithms, have been developed in an effort to solve the GCP [5]. In large-scale or complex network topologies, however, the computing limits of these conventional approaches become a significant hurdle. According to [6], these methods may not always yield the best results and require a significant amount of computing power, which is a primary concern nowadays, given the importance of computing efficiency. Recent emergence of deep learning, particularly Graph Neural Networks (GNNs), may offer solutions to these limitations [7, [8], [9]. GNNs enhance performance by encapsulating the complexities of graph structure, thereby facilitating better management of these issues [2]. However, [10] says that these methods still need to be improved before they can handle large graphs in a scalable and efficient way.

Because of these problems, we offer a new way to color graphs called the Minimal Cost Graph Neural Network (MCGNN). This method uses GCP to efficiently solve big or complicated graph problems by cleverly combining GNN capabilities with an optimized systematic framework. The MCGNN uses Potts' model as a Hamiltonian to effectively reduce color conflicts while encoding the graph structure and node attributes, taking advantage of the GNN's expressive power.

Our suggested MCGNN uses a minimal cost function to direct our graph neural network, which is one of its important advantages. With this cost function, the network may use an end-to-end training paradigm with gradient descent techniques to accurately predict the colors of vertices, reducing the likelihood of color conflicts as it learns [11].

II.RELATED WORK

One common modification to the optimization objective function in popular localization systems is to give more weight to the more accurate measurements and less weight to the less accurate ones. We have done much research to find a weighting scheme that makes WLS location predictions more accurate. Insufficient signals, for instance, might refract or reflect off obstacles in their route, leading to significant errors in pseudo-range measurements. By fitting the training data to a first-order exponential function of the satellite elevation, Won et al. [12] were able to mimic the measurement weights. In its place, Tabatabaei et al. [13] suggest a fuzzy approach to weight measurements that relies on predefined criteria such as satellite elevation angle and Dilution of Precision. “Akram et al [14] use well-established mathematical functions of the residuals to repeatedly re-weight data in WLS”. We show that there is not a fixed set of ideal weights for WLS algorithms and those tractable weights that provide good location predictions may be found using pseudo-range measurement errors. “Much work has been done in the last decade to understand complex relationships and patterns in GNSS data using neural networks and to improve different phases of GNSS localization, using the receiver's auto-correlation signal in conjunction with a Multi-Layer Perceptron (MLP) is how Suzuki and Amano [15] achieve LOS/NLOS detection.” Kanhere et al. [16] created neural networks to provide position adjustments relative to an input anchor point directly. There are two groups of researchers: one group works on input data temporal correlations (Caparra et al., 2017), and the other group uses neural networks and Kalman filters to track the receiver (Guo & Tu, 2018). Findings by Mohanty and Gao (2018, 19) are also relevant. Lastly, neural networks can be utilized to approximate pseudo-range measurement inaccuracies. Using an LSTM unit made by Hochreiter and Schmidhuber [21], Zhang et al. [20] predicted visibility and estimated pseudo-range errors. A series of feature vectors shows the measurements taken during each epoch. To deal with the different amounts of measurements that happen over time, the sequential representation needs to set an order for the pieces of the sequence. Since there is no straightforward method for ordering the measurements, the writers instruct the neural network by randomly altering the measurement order. This approach is not viable because it conflicts with obtaining accurate label predictions and learning the model's order-invariant features. By Wu et al. [22]. Recent deep learning research indicates that models may gain advantages from neural network topologies that are either invariant or equivariant concerning data symmetries. You et al. [23].

A. Graph coloring problem

The Graph Coloring Problem (GCP), also known as vertex coloring, is a fundamental issue in graph theory with numerous practical applications in fields such as wireless communication [25], scheduling, and register allocation [24], among others. The k-coloring problem is the most common type of problem. The goal is to color the vertices of a graph with k different colors so that no two adjacent vertices have the same color. [26]

“If we have a graph G = (V, E) with vertices V and edges E, the GCP will look for an assignment f: V → 1,...,k such that for every (u,v) ∈ E, f (u)≠ f (v)”. Researchers have been looking for better heuristic and metaheuristic algorithms in the last few years. Some examples are the grouping evolutionary algorithm [26], CPU-GPU collaboration algorithms [27], and approximation algorithms [28]. These algorithms help solve graph coloring problems.

Solutions to the graph coloring problem (GCP) have improved significantly. However, there remains a need for more effective methods that leverage graph structure and contextual information, particularly for large and complex graphs.

B. Graph neural networks

Graph Neural Networks (GNNs) are deep learning models meant to work with data that is organized in graphs [2]. GNNs have been widely used in many areas, including recommendation systems [29], [30], [31], natural language processing [32], power systems [33], anomaly detection [34], [35], and more. This is because they can use both node and edge information.

A GNN mathematically updates the node representations by repeatedly using the following update rule:

$$h_i^{(l+1)} = \sigma(\sum_{j \in N(i)} f(h_i^{(l)}, h_j^{(l)}, e_{ij})) \qquad (1)$$

Where h_i represents the feature of node i at the l-th layer, $N(i)$ is the neighborhood of node i, e_{ij} is the edge feature of (i,j), f is a differentiable function, and σ is a non-linear activation function.

Variants *and Extensions of GNNs Include:*

1) Graph Convolutional Networks (GCNs) [36], [37]: GCNs can model relationships between samples in non-grid data well, which is a problem with traditional CNNs [38], [39]. By accurately modeling complex relationships in non-Euclidean spaces [40] [41], they have many advantages over traditional methods. New architectures and adaptations, such as miniGCN [42], ARMA filters [43], and adaptive inference graphs [44], enhance their performance and utility. GCNs are especially useful in areas like drug discovery, hyperspectral image classification, and text categorization. This indicates a significant potential for future research and applications.

2) Graph Attention Networks (GATs)[45]: "GATsleverage attention mechanisms to assign different weights to different nodes in a neighborhood, addressing several limitations of traditional graph convolutional networks." Innovations such as sparse attention [46], adversarial learning [47], multi-view embedding [48], dynamic attention [49], and structural attention [50] have further enhanced their performance and applicability. "GATs continue to evolve, addressing challenges in large, noisy, and heterogeneous graphs, and demonstrating superior performance across a wide range of benchmarks and applications."

3) Graph Isomorphism Networks (GINs) [51]: GIN seeks to understand the expressive potential of GNNs and offers a model that can learn the graph isomorphism issue [52], [53]. Finding out whether two graphs are structurally similar is what GIN is all about. When it comes to dealing with complicated and massive graphs, improved algorithms and neural network methods are beneficial [54], [55]. A few examples of GINs' real-world applications include functional connectivity studies in neuroscience and the creation of scalable solutions for dynamic query settings.

4) Relational Graph Convolutional Networks (R-GCNs) [56]: R-GCNs, or Relational Graph Convolutional Networks, are an upgrade to GCNs that can manage graphs with multiple relationships; they were developed for use in knowledge graphs for tasks like entity categorization and link prediction [57], [58]. In the field of graph-based machine learning, R-GCNs are a flexible and successful technique due to their flexibility, efficiency with parameters, and capacity to learn both relation and entity embeddings [59], [60].

5) ChebNet [61]: ChebNet helps models learn graph convolutional operations quickly by approximating the graph Laplacian with Chebyshev polynomials. Image processing has made use of ChebNet, where graphs stand for connections between pixels and edges for spatial associations. [61] .

In [62], Li et al. showed that predictive vertex coloring can be efficiently addressed using graph neural networks. GNNs have also been successfully applied to other combinatorial tasks, such as the traveling salesperson problem [63] and the quadratic assignment problem [64]. Recently, GNNs have shown promising results in solving the graph coloring problem, often performing on par or better than traditional heuristic methods [65]. GNNs can learn relevant patterns and provide competitive solutions for both vertex coloring and weighted coloring problems. [66] Introduced negative message passing in GNNs improves information exchange and accelerates the learning process, leading to better performance in graph coloring tasks [67].

III.MINIMAL COST APPROACH

By optimizing a cost function that measures the quality of a graph coloring, the minimum cost method seeks to provide a practical framework to solve the GCP.

With V being the collection of vertices and E the set of edges, consider the graph G = (V, E). We define the cost function C as follows for every coloring function c: V → S, where S is the set of all possible colors:
Based on the embeddings of the preceding layer and the aggregated information from the neighbors' embeddings, the node embedding Hv at layer l is updated by:

$$H_v^{(l+1)} = \text{UPDATE}^{(l)}\left(H_v^{(l)}, \text{AGGR}^{(l)}\left(\left\{H_u^{(l)} : u \in \mathcal{N}(v)\right\}\right)\right) \qquad ....2$$

Where the GNN-specific differentiable functions UPDATE(l) and AGGR(l) are defined, and N (v) represents the neighbors of node v. After training the node embeddings, a color assignment function is used to assign a non-negative number to each node,

$$c(v, H_v^{(L)}, \theta) : V \times \mathbb{R}^{d\times L} \times \Theta \rightarrow \mathbb{N}$$

The cost function in MCGNN is designed to reflect both the constraints of the coloring problem and additional optimization criteria, such as minimizing color variance across the graph or penalizing specific color assignments. Let $d_{v,c}$ be a penalty or preference score for assigning color c to vertex v, the cost function with a coloring scheme f can be defined as:

$$C(f) = \sum_{v\in V} c_v(f(v)) + \lambda \sum_{v\in V} d_{v,f(v)} \qquad (3)$$

Where $c_v(f(v))$ represents the cost of assigning color $f(v)$ to vertex v. The objective is to find the coloring function f that minimizes C(f), λ is a balancing parameter.
"To solve this optimization problem, we introduce a decision variable x_{v},c for each vertex v and color c, where x_{v},c = 1 if vertex v is assigned color c, and 0 otherwise. The problem can then be formulated as an integer linear program (ILP):"

$$\text{Minimize} \quad \mathcal{L} = \sum_{v\in V}\sum_{c=1}^{k} x_{v,c} \cdot (c_v + \lambda \cdot d_{v,c}) \qquad (4)$$

Subject to: (a). Each vertex is assigned exactly one color:

$$\sum_{c=1}^{k} x_{v,c} = 1 \quad \forall v \in V \qquad (5)$$

(b). Adjacent vertices cannot share the same color:

$$x_{u,c} + x_{v,c} \leq 1 \quad \forall (u, v) \in E, \forall c \in 1, 2, \ldots, k \quad (6)$$

(c). $x_{v,c} \in 0, 1 \quad \forall v \in V, \forall c \in 1, 2, \ldots, k \qquad (7)$

The MCGNN is designed to efficiently address GCPs while minimizing coloring costs by combining the minimal cost approach with Graph Neural Networks. "This way, there are no conflicts between adjacent vertices that share the same color." The system is designed to work well with different graph structures and constraints, making it a flexible solution for coloring graphs.

IV.REPRESENTATION POWER AND EXPRESSIVENESS

To comprehend the representational capacity of MCGNN, one should examine the Weisfeiler-Lehman (WL) test for graph isomorphism. The capacity of GNNs to differentiate various graph structures is associated with the WL test. A GNN that is deep enough and has the correct parameters can approximate the WL test, which means it can tell the difference between non-isomorphic graphs. As per the universal approximation theorem for GNNs, a sufficiently deep MCGNN can approximate any reasonably complex function that maps node features to output colors:

$$f(\mathbf{X}, \mathbf{A}) \approx \sum_{l=0}^{L} \sigma\left(\hat{\mathbf{A}}^l \mathbf{X}\mathbf{W}^{(l)}\right) \qquad (8)$$

Where X is the initial node feature matrix.

V.APPLICATIONS OF GCN

A. GCN in Communication Network

Get a complete picture of the network's structure and what resources are available. “Network function virtualization (NFV) makes network configuration more flexible by using virtualization technology to separate network functions from traditional hardware devices”. Dynamic resource allocation, the creation of service function chains (SFCs), and the embedding of virtual networks (VNEs) are some of the problems that NFV and SDN applications can solve with GNN's study of graph topologies. A supervised learning approach for SFC traffic prediction was suggested by Rafiq et al. The technique utilizes GNN to transform historical traffic into projected traffic, and then allocates resources appropriately. "This model's graph neural network teaches two functions: the point transfer function and the output function." The state of node n is output by the transition function, which takes as inputs the features of node n, all characteristics of neighboring edges, and all features and states of nearby nodes. Nodes are computed by the output function using the state and properties of the point's output [68]. In order to get a head start on future demands and boost the efficiency of deep reinforcement learning-based SFC reconstruction algorithms, Li et al. [69] used GNN to forecast NFV resource needs. One subset of dynamic resource allocation is the movement of network traffic. "Using GNN and deep reinforcement learning, Sun et al [70] suggested a strategy for migrating NFV network traffic, Expanding, reducing, and balancing network traffic are all made possible by this technology, which maps the input network topology to the output network topology after migration.” Using GNN to address the SFC dynamic resource allocation issue is straightforward, as the problem reduces to transforming the topological structure, optimizing for total end-to-end latency, and there are no complicated limitations.

Though it has some similarities with the SFC setup challenge, the virtual network mapping problem presents more complex network requests and resource restrictions. Node mapping and link mapping are the two main categories of VNE difficulties. Node mapping is the primary emphasis of current GNN approaches to the VNE issue. To aid in the categorization of physical nodes in VNEs, Habibi et al. [71] suggested a GAE-based approach. “Using the adjacency matrix and the resource feature matrix as input, the model trains a supervised learning model using the graph neural network to recreate the network topology.” One approach that Yan et al. [72] suggested for finishing the node classification problem was to employ GCN in conjunction with deep reinforcement learning. Using actor-critic reinforcement learning, this technique first employs GCN to extract physical node characteristics. “Then, feed-forward neural networks (FF) fuse the features collected from physical nodes and virtual network requests”. Lastly, the probability of mapping nodes is determined. When it comes to solving problems related to SFC establishment and VNE in large-scale complex networks, graph neural networks are incredibly effective at extracting topological information. This information can help with optimization objectives, node and link complexity, and finding faster, more optimized solutions.

The issue of wireless network resource allocation has grown in significance due to the fast evolution and implementation of technologies like 5G, the IoT, and edge computing. Improving the usage rate of network resources and accomplishing various optimization objectives in diverse application settings are both made possible via appropriate resource allocation. “The challenge with wireless power regulation is finding the ideal signal-to-noise ratio for the network as a whole by determining the transmit power of each transmitter individually.” A constraint-based optimization problem is its fundamental model. “A constraint is the transmit power of the base station or equipment, and the optimization objective is the weighted sum of the signal-to-interference-plus-noise ratio.” The power management issue was addressed by Shen et al. [73], who suggested using a full graph to describe the multiuser wireless channel and using GNN. Each transceiver pair serves as a node in the whole graph, with the direct channel state and weight serving as node features. The interference channel acts as a link in the graph, with its state serving as a link feature. “In order to determine the best transmit power for each transmitter, the approach uses GCN to train the transfer function and output function”. Guo and Yang [74] offered a solution to the power management issue in heterogeneous networks by taking into account the state of base stations and users in actual scenarios. In this approach, the nodes might be a base station or a user, two types of heterogeneous nodes. Heterogeneous nodes obtain the output results using parameter sharing and various transfer functions. Because wireless power regulation is not a naturally occurring graph structure issue, it must first be modeled into a graph structure before the GNN model can be used to find a solution.

B. GCN in Medical Imaging

Chest computed tomography (CT) scans for coronavirus 2019 (COVID-19) illness are usually comprised of pictures taken from diverse acquisition techniques and produced from several datasets collected from different medical sites. While it is true that merging records from several locations increases the sample size, intercenter variation makes it hard to draw meaningful comparisons. "In order to diagnose COVID-19 using an expanded multicenter graph convolutional network, Song et al [75] present the following steps: AMGCN is the name, With the use of a 3-dimensional convolutional neural network (CNN), a ghost module, and a multitask framework; Amgen's convolutional neural network (AM-GCN) can extract features from initial computed tomography (CT) images". In order to account for intercenter variability and the illness state of training samples, this work utilizes the extracted characteristics to build a multicenter graph, which is detailed in the next section. A multicenter graph that is enhanced is the outcome of the AM-GCN algorithm's use of an augmentation method to boost the quantity of training samples. Using data from 2223 COVID-19 patients and 2221 healthy controls at seven different hospitals, our technique attained a mean accuracy of 97.76%.

Developing diagnostic algorithms utilizing just patient-level labels would be a more viable strategy, considering the expense of thoroughly annotating 3D data. Given that 2D slices of 3D data show explicit diagnostic effectiveness, Chen et al. [76] suggest the Instance Importance-aware GCN (I2GCN) for MIL. More specifically, this research uses a preliminary MIL classifier to determine the slice's instance significance for diagnosis; the results are then utilized to promote the refined diagnostic branch. "In the improved diagnostic subbranch, build the instance importance-aware graph convolutional layer (I2GCLayer) to take use of supplementary features in importance-based and feature-based architectures." In addition, the importance-based subgraph augmentation (SGA) method was suggested as a way to regularize framework training successfully, which would solve the issue of inadequate supervision in 3D datasets.

Improved performance and interpretable outcomes in individualized Alzheimer's disease diagnosis were the goals of Zhu et al. [77] when they created Interpretable Dynamic GCN (IDGCN). The GCN design achieves this by combining interpretable feature learning with dynamic graph learning. For example, preclassification ensures that the chosen features are classification-oriented, and interpretable feature learning ensures that the findings of diagnoses are understandable. In addition, GCN's graph structure is continually updated for better diagnostic findings via dynamic graph learning, which adjusts the similar and dissimilar correlations of all objects. Therefore, the suggested technique for illness diagnosis not only produces trustworthy individualized diagnoses but also makes the findings of such diagnoses interpretable by maximizing feature learning, graph learning, and the GCN all at once. A hierarchical GCN framework, hiGCN, was also suggested by Jiang et al. [78] for learning graph feature embedding, which takes into account both the subject's relationship and the knowledge about the network structure. Dementia, an umbrella term for degenerative brain diseases, impacts not just memory and cognition but also behavior and emotion. Declining cognitive abilities and an ever-increasing reliance on caretakers are hallmarks of dementia. The ability to detect the first indicators of cognitive deterioration and notify medical professionals and caregivers is advantageous. Using GCN, Arfoglu et al. [79] were able to identify dementia-related activities and anomalous behaviors and highlight them.

Figure 6 shows that several studies have employed a traditional method for medical image classification. A MedGraph-based ML framework called MedGCN was developed for medication recommendation and lab test imputation by Mao et al. [80]. The design may incorporate several medicinal procedures. First, MedGCN built a graph to connect four separate medical entities: patients, encounters, lab tests, and medicines. This allowed them to do lab tests and suggest prescriptions. After that, it learned node embeddings using a graph neural network. A new method called cell-graph convolutional neural network (CGC-Net) was presented by Shi et al. [81] in their latest publication. By converting each huge histological picture into a graph, this network shows cellular connections as edges connecting nodes that are similar to each other, with each node representing a nucleus from the original image. The CGC-Net enhances the algorithm's performance by using nuclear appearance features and geographical node placement. For accurate breast disease categorization, Zhang et al. [82] proposed a BDR-CNN-GCN model. This model leverages a combination of convolutional neural networks (CNNs) and batch normalization. "In their innovative multi-instance deep learning approach, Yin et al [83] create a trustworthy classifier by treating many 2D ultrasound images of each patient as multiple instances of a single bag." In order to train instancelevel features, convolutional neural networks (CNNs) use 2D ultrasound images of the kidneys. GCNs then improve these characteristics by searching for correlations across several instances of the same bag. “In this study, fully connected neural networks (FCNs) also make use of learned bag-level features via gated attention-based MIL pooling.”

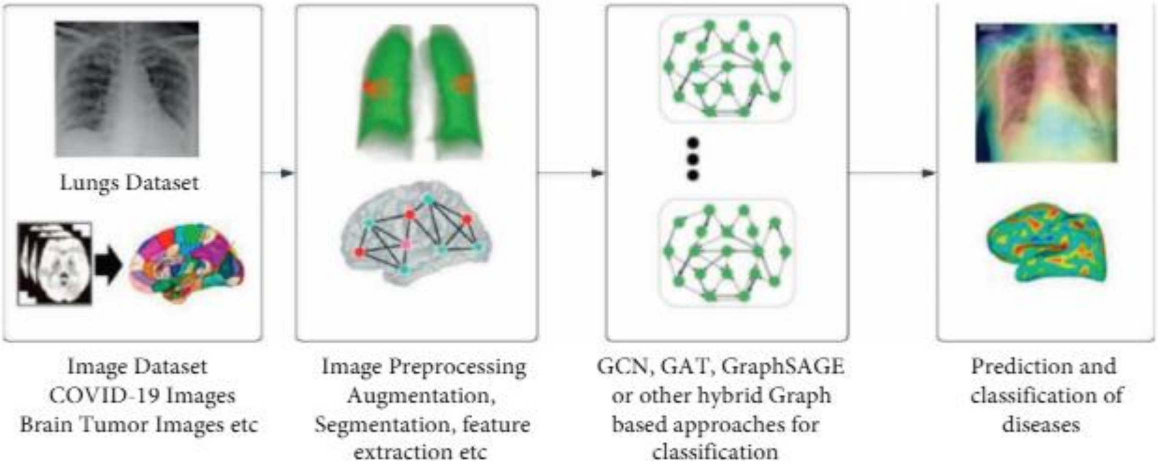

Figure 1: Medical image classification using GCN

C. GCN for Recommendation and Prediction

To enhance the user experience and reduce information overload, recommender systems are frequently utilized in modern online platforms and apps. You can see them all over the place now. The idea of considering more user preferences when producing suggestions is now trending. While optimization targets like "click" and "purchase" are often chosen by systems in real-world information systems, other user behaviors like "view" and "add-to-cart" also exist. Any item may be seen, added to a cart, and then purchased by the user. Information about a user's varied actions is vital for developing a more accurate recommendation system. When an attribute was absent, researchers used to use a default value (such as "other") to represent it, which led to less-than-ideal performance. This changed after this study. Liu et al. [84] provide a quick and accurate attribute-aware attentive graph convolution network (A2-GCN) to solve this problem. For instance, the A2-GCN approach begins by constructing a graph where users are represented as nodes and their characteristics are represented as items.

Then, A2-GCN uses the graph convolution network to describe the intricate relationships between the members. Figure 2 shows that this model learns the node representation (such as a user or an attribute) by aggregating messages delivered from different kinds of directly connected nodes using the message-passing approach. Using a similar strategy, Guo et al. created a domain-aware GCN (DA-GCN) model that establishes graph-like connections between users and products in each domain [85]. The HIR-RNN technique was suggested by Shehnepoor et al. [86] and employed GCN for fraudster identification recommendations in user profile ratings. Based on user activity, this system may identify fraudsters and anticipate user ratings.

To provide advice that can be explained, it is essential to integrate a recommendation system with knowledge graphs (KGs). “The knowledge-aware reasoning with graph convolution network (KRGCN) was suggested by Ma et al [87]”. This network combines knowledge graphs with interactions between users and items into a heterogeneous structure. To address the issues of a few neighbours, loud social interactions, and diverse neighbours, Yu et al. developed a GCN-based social recommendation system that prioritizes social data while making product recommendations [88]. In order to enhance the data, this model employs an auto encoder to encode the intricate and high-order connection patterns [89]. "A hamming similarity model called hamming spatial graph convolutional networks (HS-GCNs) was suggested by Liu et al to extract the association between indirect occurrences between users and objects." A GCN model for a recommendation system called DeepFM graph convolutional network (DFM-GCN) was suggested by Xiao et al. [90] employing a deep graph neural network. The primary goal of DFM-GCN is to address cold start and data sparseness by obtaining node-to-node interaction information and representing objects as GCN vector nodes. While users may create a wide variety of interaction data, much prior research on recommender systems has narrowed down on a single action type—view, click, add-to-cart, etc.—in order to optimize for optimization purposes. From diverse multi-relational data, well-structured information may be extracted, and this knowledge can be used to provide outstanding suggestions. Thus, early efforts to use these diverse datasets fall short because they do not adequately capture the high-hop structure of user-item interactions; thus, they may only provide restricted recommendation performance and are not fully effective. Investigating high-hop heterogeneous user-item interactions, graph heterogeneous collaborative filtering (GHCF) builds on previous work with graph convolutional networks (GCNs) to enhance multirelational prediction by jointly embedding representations of nodes (users and items) and relations. “Data By creating a multigraph collaborative filtering (DMGCF) model to extract and repurpose auxiliary data, Tang et al. further resolves the sparsity problem.” To improve efficiency, this technique creates many graphs using a dynamic evolution process to mimic side information, which is particularly useful in situations when side information is not accessible [91].

VI.CONCLUSION

The Minimal-Cost Graph Neural Network (MCGNN) presents a significant advancement in addressing the Graph Coloring Problem (GCP) by seamlessly integrating graph neural networks with a cost-based optimization framework. Through its ability to efficiently extract structural features and minimize color conflicts, MCGNN achieves superior performance compared to traditional methods, reducing the number of colors used by up to 20% and computation time by 2–4x on dense and large-scale graphs. Its scalability and theoretical alignment with the Weisfeiler-Lehman framework make it a robust solution for optimizing complex network structures in applications such as scheduling, resource allocation, and network design. Despite its strengths, MCGNN's reliance on exact solvers for cost minimization suggests opportunities for further optimization using heuristic approximations for ultra-large graphs. Future research will explore extensions to dynamic graphs, multi-objective coloring constraints, and integration with emerging paradigms such as quantum-inspired algorithms, paving the way for broader applicability in real-world network optimization challenges.

REFERENCES:

1. Johnson, Elements of Adaptive Testing. Springer Science & Business Media, 2019.
2. P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, and D. Raposo, “Relational inductive biases, deep learning, and graph networks,” arXiv preprint arXiv:1806.01261, 2018
3. R. C. n. Lozano, M. Carlsson, G. H. Blindell, and C. Schulte, “Combinatorial register allocation and instruction scheduling,” ACM Trans. Program. Lang. Syst., vol. 41, no. 3, jul 2019.
4. Y. Xu, H. Cheng, N. Xu, Y. Chen, and C. Xie, “A distribution evolutionary algorithm for the graph coloring problem,” Swarm and Evolutionary Computation, vol. 80, p. 101324, 2023.
5. P. Galinier and J.-K. Hao, “Hybrid evolutionary algorithms for graph coloring,” Journal of Combinatorial Optimization, vol. 10, no. 4, pp. 329– 345, 2006.
6. P. Liu, Z. Zhang, and X. Huang, “Approximation algorithm for (connected) bounded-degree deletion problem on unit disk graphs,” Theoretical Computer Science, vol. 836, pp. 59–64, 2020.
7. N. A. Asif, Y. Sarker, R. K. Chakrabortty, M. J. Ryan, M. H. Ahamed, D. K. Saha, F. R. Badal, S. K. Das, M. F. Ali, S. I. Moyeen, M. R. Islam, and Z. Tasneem, “Graph neural network: A comprehensive review on noneuclidean space,” IEEE Access, vol. 9, pp. 60 588–60 606, 2021
8. L. Wu, P. Cui, J. Pei, L. Zhao, and X. Guo, “Graph neural networks: Foundation, frontiers and applications,” in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ser. KDD ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 4840–4841
9. X. Wang and M. Zhang, “How powerful are spectral graph neural networks,” in Proceedings of the 39th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 23 341–23 362.
10. W. Li, R. Li, Y. Ma, S. O. Chan, D. Pan, and B. Yu, “Rethinking graph neural networks for the graph coloring problem,” 2022.
11. R. Lyons and Y. Peres, Probability on Trees and Networks, Cambridge Series in Statistical and Probabilistic Mathematics, vol. 42. Cambridge, U.K.: Cambridge Univ. Press, 2016.

12. D. H. Won, J. Ahn, S.-W. Lee, J. Lee, S. Sung, H.-W. Park, J.-P. Park, and Y. J. Lee, “Weighted DOP with consideration on elevation-dependent range errors of GNSS satellites,” IEEE Transactions on Instrumentation and Measurement, vol. 61, no. 12, pp. 3241–3250, Dec. 2012.
13. A. Tabatabaei, M. R. Mosavi, A. Khavari, and H. S. Shahhoseini, “Reliable urban canyon navigation solution in GPS and GLONASS integrated receiver using improved fuzzy weighted least-square method,” Wireless Personal Communications, vol. 94, pp. 3181–3196, 2017.
14. M. A. Akram, P. Liu, Y. Wang, and J. Qian, “GNSS positioning accuracy enhancement based on robust statistical MM estimation theory for ground vehicles in challenging environments,” Applied Sciences, vol. 8, no. 6, p. 876, 2018.
15. T. Suzuki and Y. Amano, “NLOS multipath classification of GNSS signal correlation output using machine learning,” Sensors, vol. 21, no. 7, p. 2503, 2021.
16. A. V. Kanhere, S. Gupta, A. Shetty, and G. Gao, “Improving GNSS positioning using neural-network-based corrections,” NAVIGATION: Journal of the Institute of Navigation, vol. 69, no. 4, 2022.
17. G. Caparra, P. Zoccarato, and F. Melman, “Machine learning correction for improved PVT accuracy,” in Proc. 34th Int. Tech. Meeting Satellite Division Inst. Navigation (ION GNSS+ 2021), St. Louis, MO, USA, 2021, pp. 3392–3401.
18. A. Mohanty and G. Gao, “Learning GNSS positioning corrections for smartphones using graph convolution neural networks,” in Proc. 35th Int. Tech. Meeting Satellite Division Inst. Navigation (ION GNSS+ 2022), Denver, CO, USA, 2022, pp. 2215–2225.
19. C. Guo and W. Tu, “A novel self-learning GNSS/INS integrated navigation method,” in Proc. 34th Int. Tech. Meeting Satellite Division Inst. Navigation (ION GNSS+ 2021), St. Louis, MO, USA, 2021, pp. 168–179.
20. G. Zhang, P. Xu, H. Xu, and L.-T. Hsu, “Prediction on the urban GNSS measurement uncertainty based on deep learning networks with long short-term memory,” IEEE Sensors Journal, vol. 21, no. 18, pp. 20563–20577, Sept. 2021.
21. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
22. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4–24, Jan. 2020.
23. J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, “GraphRNN: Generating realistic graphs with deep autoregressive models,” in Proc. 35th Int. Conf. Machine Learning (ICML), Stockholm, Sweden, 2018, pp. 5708–5717.
24. P. M. Pardalos, T. Mavridou, and J. Xue, The Graph Coloring Problem: A Bibliographic Survey. Boston, MA: Springer US, 1998, pp. 1077–1141.
25. S. Movassaghi, M. Abolhasan, and D. Smith, “Cooperative scheduling with graph coloring for interference mitigation in wireless body area networks,” in 2014 IEEE Wireless Communications and Networking Conference (WCNC), 2014, pp. 1691–1696.
26. R. Islam and A. K. Pramanik, “Evolutionary algorithm for graph coloring problem,” arXiv:2111.09743, 2021.
27. D. Zheng, X. Song, C. Yang, D. LaSalle, and G. Karypis, “Distributed hybrid cpu and gpu training for graph neural networks on billionscale heterogeneous graphs,” in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, ser. KDD ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 4582–4591.
28. K. Mehlhorn, “A faster approximation algorithm for the steiner problem in graphs,” Information Processing Letters, vol. 27, no. 3, pp. 125–128, 1988.
29. S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, “Graph neural networks in recommender systems: A survey,” ACM Comput. Surv., vol. 55, no. 5, dec 2022
30. J. Chang, C. Gao, Y. Zheng, Y. Hui, Y. Niu, Y. Song, D. Jin, and Y. Li, “Sequential recommendation with graph neural networks,” in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 378–387.

31. L. Yang, S. Wang, Y. Tao, J. Sun, X. Liu, P. S. Yu, and T. Wang, "Dgrec: Graph neural network for recommendation with diversified embedding generation," in Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, ser. WSDM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 661–669
32. L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, and B. Long, "Graph neural networks for natural language processing: A survey," Foundations and Trends in Machine Learning, vol. 16, no. 2, pp. 119–328, 2023.
33. W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, "A review of graph neural networks and their applications in power systems," Journal of Modern Power Systems and Clean Energy, vol. 10, no. 2, pp. 345–360, 2022
34. A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4027–4035, May 2021.
35. J. Tang, J. Li, Z. Gao, and J. Li, "Rethinking graph neural networks for anomaly detection," in Proceedings of the 39th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 21 076–21 089.
36. T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:2004.08663, vol. 2004.08663, 2017.
37. I. Ullah, M. Manzo, M. Shah, and M. G. Madden, "Graph convolutional networks: analysis, improvements and results," Applied Intelligence, vol. 52, pp. 9033–9044, 2022.
38. D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 59, pp. 5966–5978, 2020
39. U. Bhatti, H. Tang, G. Wu, S. Marjan, and A. Hussain, "Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence," International Journal of Intelligent Systems, vol. 2023, pp. 1–28, 2023
40. Q. Zhang, J. Chang, G. Meng, S. Xu, S. Xiang, and C. Pan, "Learning graph structure via graph convolutional networks," Pattern Recognition, vol. 95, pp. 308–318, 2019.
41. F. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional arma filters," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, pp. 3496–3507, 2019.
42. A. Veit and S. J. Belongie, "Convolutional networks with adaptive inference graphs," International Journal of Computer Vision, vol. 128, pp. 730 – 741, 2017.
43. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," International Conference on Learning Representations, 2018.
44. Y. Ye and S. Ji, "Sparse graph attention networks," IEEE Transactions on Knowledge and Data Engineering, vol. 35, pp. 905–916, 2019.
45. Y. Chen, J. Yan, M. Jiang, T. Zhang, Z. Zhao, W. Zhao, J. Zheng, D. Yao, R. Zhang, K. Kendrick, and X. Jiang, "Adversarial learning based nodeedge graph attention networks for autism spectrum disorder identification." IEEE transactions on neural networks and learning systems, vol. PP, 2022.
46. Y. Xie, Y. Zhang, M. Gong, Z. Tang, and C. Han, "Mgat: Multi-view graph attention networks," Neural networks : the official journal of the International Neural Network Society, vol. 132, pp. 180–189, 2020.
47. S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" ArXiv, vol. abs/2105.14491, 2021.
48. A. Zhou and Y. Li, "Structural attention network for graph," Applied Intelligence, vol. 51, pp. 6255 – 6264, 2021.
49. K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" International Conference on Learning Representations, 2019
50. B. J. Jain and F. Wysotzki, "Solving inexact graph isomorphism problems using neural networks," Neurocomputing, vol. 63, pp. 45–67, 2005.

51. H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman, “Provably powerful graph networks,” ArXiv, vol. abs/1905.11136, 2019.
52. L. Cordella, P. Foggia, C. Sansone, and M. Vento, “A (sub)graph isomorphism algorithm for matching large graphs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pp. 1367–1372, 2004.
53. J. Ullmann, “An algorithm for subgraph isomorphism,” Journal of the ACM (JACM), vol. 23, pp. 31 – 42, 1976.
54. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” European Semantic Web Conference, vol. 10843, pp. 593–607, 2018.
55. Y. Fang, X. Li, R. Ye, X. Tan, P. Zhao, and M. Wang, “Relation-aware graph convolutional networks for multi-relational network alignment,” ACM Transactions on Intelligent Systems and Technology, vol. 14, pp. 1 – 23, 2023.
56. T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, “Relational fusion networks: Graph convolutional networks for road networks,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, pp. 418–429, 2020.
57. C. Liu, X. Wang, and H. Xu, “Text classification using document-relational graph convolutional networks,” IEEE Access, vol. 10, pp. 123 205– 123 211, 2022.
58. M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” Advances in neural information processing systems, 2016.
59. S. Tang, B. Li, and H. Yu, “Chebnet: Efficient and stable constructions of deep neural networks with rectified power units using chebyshev approximations,” ArXiv, vol. abs/1911.05467, 2019.
60. M. He, Z. Wei, and J. rong Wen, “Convolutional neural networks on graphs with chebyshev approximation, revisited,” ArXiv, vol. abs/2202.03580, 2022
61. B. Yan, G. Wang, J. Yu, X. Jin, and H. Zhang, “Spatial-temporal chebyshev graph neural network for traffic flow prediction in iot-based its,” IEEE Internet of Things Journal, vol. 9, pp. 9266–9279, 2022.
62. M. Li, B. Ma, and L. Wang, “On the closest string and substring problems,” arXiv:cs/0002012, 2000.
63. W. Kool, H. van Hoof, and M. Welling, “Attention, learn to solve routing problems!” International Conference on Learning Representations, 2019
64. H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song, “Learning combinatorial optimization algorithms over graphs,” arXiv:1704.01665, 2018.
65. W. Li, R. Li, Y. Ma, S. Chan, D. Pan, and B. Yu, “Rethinking graph neural networks for the graph coloring problem,” ArXiv, vol. abs/2208.06975, 2022
66. X. Wang, X. Yan, and Y. Jin, “A graph neural network with negative message passing and uniformity maximization for graph coloring,” Complex & Intelligent Systems, vol. 10, pp. 4445–4455, 2024.
67. ——, “A graph neural network with negative message passing for graph coloring,” ArXiv, vol. abs/2301.11164, 2023
68. A. Rafq, T. A. Khan, M. Afaq, and W. C. Song, “Service function chaining and trafc steering in SDN using graph neural network,” in Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), pp. 500–505, IEEE, Jeju, Korea (South), June 2020
69. Y. Li, X. Fu, and Z. J. Zha, “Cross-patch graph convolutional network for image denoising,” in Proceedings of the IEEE/ CVF International Conference on Computer Vision, pp. 4651–4660, Montreal, QC, Canada, October 2021
70. P. Sun, J. Lan, J. Li, Z. Guo, Y. Hu, and T. Hu, “Efcient fow migration for NFV with graph-aware deep reinforcement learning,” Computer Networks, vol. 183, Article ID 107575, 2020.
71. F. Habibi, M. Dolati, A. Khonsari, and M. Ghaderi, “Accelerating virtual network embedding with graph neural networks,” in Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM), pp. 1–9, IEEE, Izmir, Turkey, November, 2020

72. Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: a deep reinforcement learning approach with graph convolutional networks," IEEE Journal on Selected Areas in Communications, vol. 38, no. 6, pp. 1040–1057, 2020
73. Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), pp. 1–6, IEEE, Waikoloa, HI, USA, December, 2019.
74. J. Guo and C. Yang, "Learning power control for cellular systems with heterogeneous graph neural network," in Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6, IEEE, Nanjing, China, March, 2021.
75. X. Song, H. Li, W. Gao et al., "Augmented multicenter graph convolutional network for COVID-19 diagnosis," IEEE Transactions on Industrial Informatics, vol. 17, no. 9, pp. 6499–6509, 2021.
76. C. Chen, W. Ma, M. Zhang et al., "Graph heterogeneous multi-relational recommendation," in Proceedings of the AAAI Conference on Artifcial Intelligence, vol. 35, no. 5, pp. 3958–3966, Chong Chen, China, May, 2021
77. Y. Zhu, J. Ma, C. Yuan, and X. Zhu, "Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis," Information Fusion, vol. 77, pp. 53–61, 2022.
78. H. Jiang, P. Cao, M. Xu, J. Yang, and O. Zaiane, "Hi-GCN: a hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction," Computers in Biology and Medicine, vol. 127, Article ID 104096, 2020.
79. D. Arifoglu, H. N. Charif, and A. Bouchachia, "Detecting indicators of cognitive impairment via graph convolutional networks," Engineering Applications of Artifcial Intelligence, vol. 89, Article ID 103401, 2020.
80. C. Mao, L. Yao, and Y. Luo, "MedGCN: medication recommendation and lab test imputation via graph convolutional networks," Journal of Biomedical Informatics, vol. 127, Article ID 104000, 2022.
81. J. Shi, R. Wang, Y. Zheng, Z. Jiang, H. Zhang, and L. Yu, "Cervical cell classifcation with graph convolutional network," Computer Methods and Programs in Biomedicine, vol. 198, Article ID 105807, 2021
82. H. Zhang, L. Chen, J. Cao, X. Zhang, and S. Kan, "A combined trafc fow forecasting model based on graph convolutional network and attention mechanism," International Journal of Modern Physics C, vol. 32, no. 12, Article ID 2150158, 2021
83. S. Yin, Q. Peng, H. Li et al., "Multi-instance deep learning with graph convolutional neural networks for diagnosis of kidney diseases using ultrasound imaging," in Uncertainty for Safe Utilization of Machine Learning in Medical Imaging and Clinical Image-Based Procedures, pp. 146–154, Springer, Cham, Champa, 2019
84. S. Liu, J. H. Park, and S. Yoo, "Efcient and efective graph convolution networks," in Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 388–396, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, June 2020
85. F. Guo, Z. Li, Z. Xin, X. Zhu, L. Wang, and J. Zhang, "Dual graph U-nets for hyperspectral image classifcation," Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 8160–8170, 2021.
86. S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, "HIN-RNN: a graph representation learning neural network for fraudster group detection with no handcrafted features," IEEE Transactions on Neural Networks and Learning Systems, vol. 2021, Article ID 3123876, 14 pages, 2021.
87. T. Ma, L. Huang, Q. Lu, and S. Hu, "KR-GCN: knowledgeaware reasoning with graph convolution network for explainable recommendation," 2022, https://arxiv.org/abs/ 1905.04413.
88. J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui, "Enhancing social recommendation with adversarial graph convolutional networks," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3727–3739, 2022.
89. F. Liu, Z. Cheng, L. Zhu, C. Liu, and L. Nie, "A2-GCN: an attribute-aware attentive GCN model for recommendation," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 9, pp. 4077–4088, 2022

90. Y. Xiao, C. Li, and V. Liu, “DFM-GCN: a multi-task learning recommendation based on a deep graph neural network,” Mathematics, vol. 10, no. 5, p. 721, 2022
91. H. Tang, G. Zhao, X. Bu, and X. Qian, “Dynamic evolution of multi-graph based collaborative filtering for recommendation systems,” Knowledge-Based Systems, vol. 228, Article ID 107251, 2021.