Article

Not peer-reviewed version

Graph Neural Network Based Learning for Facility Location Optimization

Carmen Cvancara *

Posted Date: 5 October 2024

doi: 10.20944/preprints202410.0329.v1

Keywords: Deep Learning; Graph Neural Network; Facility Location; Mixed Integer Programming

Article

Graph Neural Network Based Learning for Facility Location Optimization

Carmen Cvancara

Affiliation 1; carmen.cvancara@outlook.com

Abstract: This paper investigates the application of Graph Neural Networks (GNNs) in solving the facility location decision problem. By adapting the learning objectives and structure of GNNs, this research bridges the gap between traditional optimization approaches and modern machine learning techniques. The proposed method demonstrates improved decision-making capabilities, providing promising results for facility location optimization. Additionally, potential future research directions are outlined, highlighting areas where GNNs can further enhance decision-making processes in complex supply chain networks. Hybrid GNN and Mixed Integer Programming solutions have been proposed.

Keywords: deep learning; graph neural network; facility location and mixed integer programming

1. Introduction

Facility location optimization is a critical problem in supply chain logistics, affecting transportation costs, service quality, and overall efficiency. Traditionally, various methods like mixed-integer programming, heuristics, and metaheuristics have been applied to solve such location-allocation problems. However, with the advent of machine learning and, more specifically, Graph Neural Networks (GNNs), a new dimension has emerged in tackling this problem. GNNs have proven effective in solving combinatorial optimization problems, particularly those that can be modeled as graphs, making them suitable for warehouse location optimization. Traditional approaches to warehouse location problem needs some new directions, and here we bring in GNN from a different angle.

2. Literature Review

In supply chain networks, warehouses and distribution hubs can be represented as nodes, and transportation routes can be modeled as edges between nodes. The problem of selecting the optimal warehouse locations in a supply chain can be viewed as a combinatorial optimization problem, where we aim to minimize costs while maximizing the system's efficiency. The complex nature of this problem stems from the many factors involved, such as transportation costs, service levels, customer demand, and geographical constraints. Traditional methods often struggle to scale efficiently with the increasing complexity of supply chain networks. In contrast, GNNs, which can effectively model dependencies between nodes (warehouses) and edges (routes), have shown promise in handling these complexities by learning high-dimensional features of graph-structured data.

One of the primary advantages of using GNNs in this context is their ability to capture spatial and topological relationships between different locations in a supply chain. Graphs naturally represent these relationships, with nodes representing physical locations (such as potential warehouse sites or distribution hubs) and edges representing transportation routes [1]. GNNs excel at embedding this graph structure into a latent space, which can then be used to optimize the location of warehouses in terms of minimizing transportation costs and balancing the flow of goods across the supply chain. GNN has proven to have the ability to handle complex data presented by Wang et al. [2] in sports analytics. By training a GNN on historical data, the model can learn to predict the optimal locations based on factors such as transportation costs, customer proximity, and distribution patterns.

Furthermore, the graph-based structure allows for the incorporation of dynamic elements, such as varying demand patterns or fluctuating transportation costs. These dynamic factors are challenging to model using traditional methods but can be integrated into a GNN framework. For example, dynamic Graph Neural Networks (DGNNs) can handle time-varying information by using temporal graph structures. This feature makes GNNs suitable for real-time decision-making in supply chain networks, where warehouse and hub locations may need to be adapted due to shifting demand or disruptions in transportation networks.

Another important aspect to consider is scalability. Supply chains often encompass a large number of nodes and edges, especially in global logistics networks. Traditional optimization techniques, such as mixed-integer programming or even heuristic approaches, may struggle to scale efficiently with increasing network size and complexity. GNNs, on the other hand, are designed to scale with the size of the graph. By leveraging message-passing mechanisms and neural network layers, GNNs can process large graphs efficiently while retaining valuable information about node and edge relationships.

Several recent studies have demonstrated the potential of GNNs in location optimization problems. For example, [3] explored the use of GNNs for solving facility location problems by modeling supply chain networks as graphs and using the GNN to predict optimal facility placements. Their results indicated significant improvements in computational efficiency and solution quality compared to traditional methods. Additionally, other research has focused on integrating GNNs with reinforcement learning approaches to optimize warehouse locations dynamically based on real-time data [4]. There are quite a number of solutions using machine learning to determine distribution network hub location, as shown in [5], [6] and [7]. [8] model it as a multi-criteria optimization problem for machine learning.

3. Methodology

3.1. GNN Learning Objectives and Optimization

The learning objective and optimization process in Graph Neural Networks (GNNs) for warehouse and distribution hub location optimization is the core of how these models are trained and used to find optimal solutions. In this context, GNNs are employed to predict and optimize the placement of warehouses or hubs in supply chains, aiming to minimize costs, maximize service efficiency, or balance other operational constraints.

In a typical warehouse or distribution hub location optimization problem, the goal is to find the best locations for a set of facilities (e.g., warehouses or hubs) while minimizing a specific objective function [9]. Common objectives include:

- Minimizing transportation costs (e.g., shipping from warehouses to customers or suppliers to warehouses).
- Maximizing service levels (e.g., reducing delivery times by locating warehouses closer to high-demand areas).
- Balancing inventory holding costs and operational costs of facilities.
- Minimizing the environmental impact (e.g., reducing carbon emissions by optimizing transportation routes).

These objectives are often subject to various constraints, such as:

- The maximum number of warehouses that can be opened.
- Capacity constraints for each warehouse.
- Demand fulfillment requirements for each region or customer.

This optimization problem is typically combinatorial in nature, meaning that there are many possible configurations of warehouse locations, and the challenge is to select the one that minimizes (or maximizes) the objective function.

In a supervised learning setup, GNNs are trained on historical data where optimal or near-optimal warehouse locations are already known. The data might include:

- Node features: Each node in the graph could represent a location (e.g., a potential warehouse site, customer, or supplier) and could include features like geographic coordinates, demand, costs, capacities, and proximity to other locations.
- Edge features: Edges represent transportation routes or relationships between nodes (e.g., distance between locations or transportation costs) and could include information like transit times, road conditions, or route capacities.

The GNN learns a mapping from this graph-structured input data to a target output, which could be a binary classification of whether a particular location should be selected as a warehouse or a regression task to predict the overall cost or efficiency of a particular configuration. The loss function in this case quantifies how far the predicted warehouse locations are from the optimal solution, and it might involve minimizing transportation costs or maximizing service efficiency. The most common loss functions used in this context are:

- Mean Squared Error (MSE): If the GNN is performing a regression task, such as predicting the overall transportation cost or service time for a given configuration of warehouses, the loss function could be the MSE between the predicted cost and the true cost (see Equation (1)).

  $$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2 \tag{1}$$

  where y_i is the true cost, y_i' is the predicted cost, and N is the number of samples in the dataset.
- Binary Cross-Entropy Loss: If the GNN is solving a binary classification problem (e.g., deciding whether to open a warehouse at a specific location), the loss function might be binary cross-entropy, which penalizes incorrect predictions of whether a location is suitable for a warehouse (see Equation (2)).

  $$L_{BCE} = \frac{1}{N}\sum_{i=1}^{N}(log(y_i') + (1 - y_i)log(1 - y_i')) \tag{2}$$

  where y_i is the true label (1 if a warehouse should be placed, 0 otherwise), and y_i' is the predicted probability of placing a warehouse at that location.

During training, the GNN updates its internal parameters using backpropagation and stochastic gradient descent (or one of its variants) to minimize this loss function, learning to predict more accurate warehouse locations or costs.

3.1.1. Optimization Objective

In many real-world applications, the GNN is tasked with minimizing an overall objective function that encapsulates the costs associated with the supply chain network [8]. The most common objectives in warehouse location optimization problems include:

- Transportation Cost Minimization: This objective seeks to reduce the total cost of transporting goods between suppliers, warehouses, and customers. A GNN can be trained to predict which warehouse locations minimize this cost by learning from historical transportation data and the structure of the supply chain network (Equation (3)).

  $$\min \sum_{i \in warehouses} \sum_{j \in customers} c_{ij}x_{ij} \tag{3}$$

  where c_{ij} represents the transportation cost between warehouse i and customer j, and x_{ij} is a binary variable indicating whether warehouse i serves customer j.
- Service Level Maximization: Another common objective is to maximize the service level, which might involve minimizing delivery times or ensuring that all customers are within a certain

distance of a warehouse. GNNs can be trained to predict warehouse locations that maximize service level by learning patterns in customer demand and geographic proximity (see Equation (4)).

$$\max \sum_{i \in customers} S_i \tag{4}$$

Where S_i represents the service level (e.g., percentage of orders delivered within a certain time) for customer i.

- Multi-Objective Optimization: In more complex scenarios, a GNN might optimize for multiple objectives simultaneously, such as balancing transportation costs with service levels or minimizing carbon emissions while maintaining profitability. In such cases, a multi-objective loss function is used, which combines several objectives into a single loss function with appropriate weights (see Equation (5)).

$$L = \alpha \times Cost + \beta \times ServiceLevel - \gamma \times Sustainability \tag{5}$$

Here α, β, and γ are weight factors that determine the relative importance of each objective (cost, service level, and sustainability).

3.1.2. Generalization to New Configurations

Once the GNN has been trained on historical data, the model can generalize to new configurations and networks. For example, if a company is expanding into new regions or if demand patterns change, the GNN can be fed new input data reflecting these changes, and it can predict the optimal locations for new warehouses. This generalization capability is one of the key advantages of GNNs over traditional optimization techniques, which often need to be solved from scratch for each new problem instance [10].

In this sense, the GNN learns a universal representation of the supply chain network, capturing general principles about how costs and service levels depend on warehouse locations and transportation routes. This allows the GNN to provide solutions to a wide variety of different optimization problems without needing to be re-engineered for each specific case.

3.1.3. Integration with Traditional Optimization Techniques

In some cases, GNNs are not used in isolation but are integrated with traditional optimization techniques. One common approach is to use the GNN to generate candidate solutions or a set of potential warehouse locations, which are then fed into a more traditional optimization algorithm (such as mixed-integer programming or a heuristic solver) to fine-tune the solution. For example, the GNN might predict a set of promising warehouse locations based on the graph structure and then a linear programming solver could be used to optimize the allocation of goods between these warehouses and customers, subject to additional operational constraints. This hybrid approach combines the scalability and learning capabilities of GNNs with the rigorous guarantees provided by traditional optimization methods.

3.1.4. Reinforcement Learning for Dynamic Optimization

In some cases, GNNs are integrated with reinforcement learning (RL) to handle more dynamic scenarios, where the objective is not just to find a static set of warehouse locations but to adapt them over time as demand patterns, transportation costs, or other factors change. Here, the GNN serves as a policy network that learns to take actions (such as opening or closing a warehouse) based on the current state of the supply chain.

The RL agent receives a reward based on the objective function (e.g., a reward for reducing transportation costs or improving service levels), and it uses the GNN to make decisions about how to modify the warehouse locations in response to changing conditions. Over time, the GNN learns to optimize the network dynamically, adapting to shifts in demand, disruptions in transportation, or changes in supply.

3.2. Message Passing Method for GNN in Warehouse Location

In the context of GNNs, the goal is to learn node representations (embeddings) that capture the relevant characteristics of each location (node) while considering the connectivity patterns (edges) in the graph. These learned embeddings can then be used for optimization tasks such as selecting the best locations for warehouses or minimizing overall transportation costs.

At the core of GNNs is the message-passing mechanism, which allows nodes in a graph to aggregate information from their neighboring nodes iteratively. In the case of location optimization, this mechanism helps each warehouse or distribution hub node to gather information from surrounding locations, such as transportation costs, proximity to demand centers, or the availability of other facilities. This process enables the network to understand how changes in one part of the supply chain impact other parts, capturing dependencies that would be difficult to model using conventional methods.

For example, a GNN could help a warehouse node "understand" the flow of goods and costs from its neighboring nodes (such as suppliers or customer demand points) by aggregating the features from those neighbors. As the message-passing process continues through multiple layers, nodes are able to gather a more global perspective of the entire network, enabling them to make more informed decisions about their role in the overall supply chain.

The concept of message passing is central to how GNNs operate. During this process, a node (such as a warehouse) collects information from its neighboring nodes (such as suppliers, distribution hubs, or customer locations). This information aggregation allows the node to "understand" not only its own features (e.g., location, capacity, or operational costs) but also the features and relationships of nearby nodes.

In GNNs, the message-passing mechanism works in multiple layers, where each layer updates the node's representation based on the information it receives from its neighbors. In the case of warehouse location optimization, the process can be described as follows: In the Initial Node Representation stage, at the beginning, each node (warehouse) has an initial set of features that describe it [11]. For a warehouse node, these features might include: 1. Geographic location (e.g., coordinates or region), 2. Operational costs (e.g., fixed and variable costs), 3. Capacity constraints (e.g., how much inventory the warehouse can handle), 4. Proximity to demand centers (e.g., distances to customers). Neighboring nodes, like suppliers or customers, will have their own sets of features such as supply levels, demand, transportation costs, etc.

First Layer of Message Passing: In the first layer, the warehouse node aggregates features from its directly connected neighbors (e.g., customers, suppliers, or other warehouses). For instance, if a customer node is connected to the warehouse, it might pass its demand level and distance to the warehouse. If a supplier node is connected, it might pass its supply capacity and transportation cost to the warehouse. The aggregation function used in GNNs could be as simple as taking the sum, average, or a weighted combination of the features from neighboring nodes [12]. The output is an updated representation of the warehouse that incorporates not only its own features but also information about the supply and demand patterns from nearby nodes.

Now we come to the Subsequent Layers of Message Passing stage: As the message-passing process continues across multiple layers, each node starts receiving information from nodes that are further away in the graph. This means that the warehouse node no longer just has information about its immediate neighbors, but also about nodes that are two or more steps away (such as suppliers or other customers further down the chain).

For example, if a warehouse is connected to a supplier, and that supplier is connected to multiple other warehouses or manufacturing plants, the warehouse can start to understand the larger supply network. It could "learn" that there is another nearby warehouse that might be better positioned to serve certain demand points or that the cost of goods from a distant supplier might be higher due to transportation issues further up the supply chain.

If we take a look from the learning a global perspective: As the warehouse aggregates more information from the broader network, it builds a more global understanding of the entire supply chain. This global view helps the node "see" not just its own role but also how it fits into the larger logistics system. For example, the warehouse can use this global knowledge to determine:

- Which routes or connections are more cost-effective for distributing goods. How demand is distributed across the network, helping it decide which regions to prioritize.
- How opening or closing the warehouse will affect not only its direct customers but also other nodes in the supply chain (like neighboring warehouses or suppliers).

By using these learned embeddings, GNNs can optimize decisions, such as determining the best warehouse locations, adjusting inventory levels, or minimizing transportation costs [13].

A good analogy is to think of the warehouse as a manager in a logistics company. Initially, the manager only knows the basic details about their own warehouse: the number of staff, the inventory levels, and the cost of operation. After communicating with nearby warehouses and customers (message passing), the manager learns about the demand and supply situation in neighboring areas. As communication continues (more layers), the manager starts to understand the global logistics network, including other warehouses, distant suppliers, and the overarching demand patterns.

Armed with this broader knowledge, the manager can make better decisions about how to allocate resources, where to place new warehouses, or how to improve the transportation network to meet customer demand while minimizing costs. This is essentially what GNNs do in an automated, data-driven way.

4. Hybrid Solution: Integrating GNNs with Traditional Operations Research Methods

While GNNs offer significant advantages in capturing complex, data-driven relationships within supply chain networks, they are not always a complete replacement for traditional operations research (OR) methods, such as mixed-integer programming (MIP), linear programming (LP), and heuristic algorithms like genetic algorithms or simulated annealing. Instead, a hybrid approach that integrates GNNs with traditional OR techniques can combine the strengths of both, yielding more efficient, scalable, and accurate solutions for the distribution centre location problem.

Traditional OR methods excel in handling structured optimization problems where constraints and objectives are clearly defined, such as the classical facility location problem (FLP) or the capacitated distribution centre location problem (CDCLP). These methods provide guarantees of optimality (in the case of exact algorithms like MIP) or near-optimal solutions (in the case of heuristics), making them highly reliable for decision-making. However, they often struggle with scalability and flexibility in dynamic, large-scale networks, especially when there are complex interdependencies and uncertainties in the data.

On the other hand, GNNs shine in learning patterns from data, modeling large, interconnected networks, and generalizing across different scenarios. They can capture the relationships between nodes (distribution centres, customers, suppliers) and learn from real-world datasets, incorporating both local and global features through message passing. However, GNNs do not inherently account for the hard constraints typically required in OR formulations, such as capacity limits, budget constraints, or specific service level requirements.

A hybrid approach combines these strengths, allowing GNNs to generate high-quality, data-driven solutions while leveraging OR techniques to ensure that all constraints are satisfied and the solution is optimized further. The hybrid approach typically follows a two-stage process:

4.1. Stage 1: GNN-Based Prediction and Candidate Generation

In the first stage, a GNN is used to analyze the supply chain graph and predict potential solutions. This step focuses on learning from historical data and capturing the spatial and relational dependencies in the network. The GNN can be trained to generate:

Candidate distribution centre locations: The GNN can predict a set of locations that are likely to minimize costs or maximize service levels based on the structure of the network and historical demand patterns.

Node embeddings

The GNN outputs high-dimensional representations of each node (e.g., potential distribution centre locations, customer demand points, or suppliers). These embeddings encapsulate both local features (e.g., costs, distances) and global information (e.g., supply chain structure), which can be used to inform downstream optimization.

4.2. Stage 2: OR-Based Refinement and Optimization

Once the GNN has generated a set of candidate solutions or potential distribution centre locations, the second stage involves refining and optimizing these solutions using traditional OR methods. This stage typically includes:

Constraint Handling: OR techniques, such as MIP or LP, are applied to ensure that the final solution adheres to critical constraints, such as capacity limits at distribution centres, budget constraints for opening new facilities, and service level agreements with customers. These constraints may not be fully incorporated into the GNN training process, so OR methods provide a way to ensure feasibility.

For example, if the GNN suggests several potential locations for new distribution centres, an OR model can evaluate these candidates while ensuring that the total number of centres does not exceed a given budget or that no single centre is overloaded.

Cost and Service Optimization

OR methods can fine-tune the GNN-generated solutions by further optimizing the objective function (e.g., minimizing transportation and operational costs). For instance, a MIP model could optimize the allocation of customer demand to the predicted distribution centre locations, ensuring that transportation routes are chosen to minimize costs while satisfying all demand.

Multi-Objective Optimization

In cases where there are multiple objectives (e.g., balancing cost minimization with service level maximization), traditional OR techniques like goal programming or Pareto optimization can help balance trade-offs, ensuring that the solution meets the business's priorities.

4.3. Hybrid Example: GNN + MIP for Distribution Centre Location

Consider a hybrid solution for the distribution centre location problem where a company wants to minimize total transportation and operational costs while meeting specific demand fulfillment and capacity constraints:

- GNN Candidate Generation: The GNN is first trained on historical data, learning from past distribution centre locations, transportation costs, and customer demand patterns. The model outputs a set of potential locations that are likely to reduce costs based on the graph structure, transportation routes, and demand distribution.
- MIP Optimization: Once the GNN has predicted the candidate locations, a mixed-integer programming (MIP) model is used to refine these candidates by solving a constrained optimization problem. The MIP ensures that the chosen locations meet the capacity and service level constraints while further minimizing the overall cost function.

This combination allows the GNN to handle the complexity of learning from large-scale, interconnected supply chain data, while the MIP ensures that the solution is both feasible and optimized according to traditional supply chain metrics. The hybrid solution of a MIP and GNN can also be found in [14], [9] and [15]. In the survey for solving mixed integer programming via machine learning conducted by [16], the author have summarized some systemic ways to handle MIP with machine learning.

4.4. Benefits of Hybrid Solutions

The hybrid GNN-OR approach offers several key benefits:

- Scalability: GNNs excel at handling large, complex networks with numerous nodes and edges, providing a scalable way to generate candidate solutions. By using OR methods for refinement, the approach ensures that the solution remains feasible and optimized, even in large supply chain networks.
- Adaptability: GNNs can learn from dynamic, real-time data, making the hybrid approach adaptable to changing conditions such as fluctuations in demand, supply disruptions, or changes in transportation costs. OR methods can then fine-tune the solution based on the latest data while ensuring optimality.
- Efficiency: By using GNNs to narrow down the search space and generate high-quality initial solutions, the OR methods can focus on optimizing a smaller, more manageable set of candidates, reducing computation time and improving efficiency.
- Improved Decision-Making: The hybrid approach allows companies to make more informed decisions by incorporating both data-driven insights from GNNs and rigorous optimization from OR methods. This combination helps businesses strike the right balance between minimizing costs, maximizing service levels, and adhering to operational constraints.

As hybrid solutions involving GNNs and OR techniques continue to evolve, several research directions are worth exploring:

- Dynamic optimization: The integration of GNNs with real-time OR methods, such as rolling-horizon optimization or reinforcement learning, could enable more dynamic and adaptive distribution centre location decisions, allowing companies to respond quickly to supply chain disruptions or demand shifts.
- Uncertainty modeling: Traditional OR methods can incorporate uncertainty through stochastic programming or robust optimization. A promising direction is to combine GNNs with these uncertainty-aware OR methods to handle variability in supply chain data, such as unpredictable demand patterns or transportation delays.
- Multi-objective optimization: As supply chains become more complex, multi-objective hybrid solutions will become increasingly important. Future work could explore ways to better integrate GNNs with OR-based multi-objective optimization techniques, balancing trade-offs between conflicting objectives like cost, sustainability, and customer service.

4.5. Informed Decision-Making and Other Application in Location Optimization

The final result of this message-passing process is that each warehouse node has a richer, multidimensional representation that encapsulates its own features along with the aggregated information from the entire supply chain network. These embeddings can be used to: 1. select optimal warehouse locations: The learned representations can help predict which locations minimize costs or maximize service levels. 2. Adjust transportation routes: Understanding how different nodes are connected allows for optimizing transportation paths to reduce costs and delivery times. 3. Respond to dynamic changes: As the network evolves (e.g., demand patterns shift or transportation costs change), the GNN can continue to learn and adapt the warehouse locations and routes based on the latest data.

doi:10.20944/preprints202410.0329.v1

Let's say a logistics company is trying to decide where to open a new warehouse. The GNN first represents the existing supply chain network as a graph, with nodes representing existing warehouses, customers, and suppliers, and edges representing the transportation routes between them. Through message passing, the model allows each warehouse node to gather information from the surrounding nodes. As a result, the GNN might learn that opening a new warehouse in a particular region would reduce overall transportation costs because that location is closer to several high-demand customers and well-connected to suppliers. The decision is based on a holistic view of the entire supply chain rather than isolated, local data.

In summary, the message-passing mechanism in GNNs enables each warehouse node to "understand" not only its direct connections but also how it fits into the broader supply chain. By aggregating information from multiple layers of neighbors, the GNN can optimize decisions such as warehouse location, transportation routes, and inventory management in a way that balances local efficiencies with global supply chain dynamics.

5. Future Research Direction

Despite the promise of GNNs, challenges remain in applying them to warehouse and distribution hub location optimization. One challenge is the need for large amounts of high-quality data to train the models effectively. In many real-world supply chain networks, historical data on transportation routes, costs, and demand patterns may be incomplete or noisy. This can lead to suboptimal GNN performance unless proper data preprocessing and augmentation techniques are employed. Furthermore, GNNs are often seen as "black box" models, making their predictions difficult to interpret. We have proposed ways to avoid such suboptimal solutions. In a field like logistics, where decision-makers need to understand the rationale behind location choices, this lack of interpretability can be a limitation. However, ongoing research into explainable AI (XAI) techniques for GNNs offers a potential solution to this issue by providing insights into how the model makes decisions.

Moreover, integrating GNNs with other machine learning techniques and traditional optimization approaches may offer a more robust solution. For instance, hybrid models that combine GNNs with linear programming or heuristic-based solvers can potentially leverage the strengths of both approaches. Such hybrid frameworks could provide a more comprehensive solution to the warehouse location problem, combining the GNN's ability to model complex relationships with the interpretability and reliability of traditional optimization techniques.

6. Conclusions

Incorporating Graph Neural Networks into traditional operations research frameworks presents a powerful hybrid solution for solving distribution centre location optimization problems. GNNs provide a scalable and flexible method for learning from complex, data-rich supply chain networks, while OR techniques ensure that the final solution is both feasible and optimized according to business constraints. This hybrid approach not only improves the quality of decision-making in static distribution centre location problems but also opens up new possibilities for dynamic and real-time optimization, allowing supply chains to become more adaptive and resilient in an increasingly uncertain world.

References

1. Hussain, M.; Hill, R. Custom lightweight convolutional neural network architecture for automated detection of damaged pallet racking in warehousing & distribution centers. *IEEE Access* **2023**, *11*, 58879–58889.
2. Wang, Z.; Zhu, Y.; Li, Z.; Wang, Z.; Qin, H.; Liu, X. Graph neural network recommendation system for football formation. *Applied Science and Biotechnology Journal for Advanced Research* **2024**, *3*, 33–39.
3. Wu, J.; Zhang, L.; Wang, H. Graph neural networks for facility location optimization. *IEEE Transactions on Systems, Man, and Cybernetics* **2020**, *50*, 4012–4023.

4. Li, W.; Zhao, M.; Liu, Y. Reinforcement learning and GNNs for dynamic warehouse location optimization. *Journal of Artificial Intelligence Research* **2021**, *70*, 255–275.
5. Bacanli, S.S.; Cimen, F.; Elgeldawi, E.; Turgut, D. Placement of package delivery center for UAVs with machine learning. 2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021, pp. 1–6.
6. Wei, J.Y.; Wang, C.H. A novel approach—fuzzy ANP for distribution center location. 2009 International Conference on Machine Learning and Cybernetics. IEEE, 2009, Vol. 1, pp. 537–542.
7. Mirshekali, H.; Dashti, R.; Keshavarz, A.; Shaker, H.R. Machine learning-based fault location for smart distribution networks equipped with micro-PMU. *Sensors* **2022**, *22*, 945.
8. Wei, R.; Yin, H.; Jia, J.; Benson, A.R.; Li, P. Understanding non-linearity in graph neural networks from the bayesian-inference perspective. *Advances in Neural Information Processing Systems* **2022**, *35*, 34024–34038.
9. Wu, Y.; Song, W.; Cao, Z.; Zhang, J.; Gupta, A.; Lin, M. Graph learning assisted multi-objective integer programming. *Advances in Neural Information Processing Systems* **2022**, *35*, 17774–17787.
10. Kim, H.G.; Park, S.; Lange, S.; Lee, D.; Heo, D.; Choi, H.; Yoo, J.H.; Hong, J.W.K. Graph neural network-based virtual network function deployment optimization. *International Journal of Network Management* **2021**, *31*, e2164.
11. Vignac, C.; Loukas, A.; Frossard, P. Building powerful and equivariant graph neural networks with structural message-passing. *Advances in neural information processing systems* **2020**, *33*, 14143–14155.
12. Balcilar, M.; Héroux, P.; Gauzere, B.; Vasseur, P.; Adam, S.; Honeine, P. Breaking the limits of message passing graph neural networks. International Conference on Machine Learning. PMLR, 2021, pp. 599–608.
13. Zhong, Z.; Li, C.T.; Pang, J. Hierarchical message-passing graph neural networks. *Data Mining and Knowledge Discovery* **2023**, *37*, 381–408.
14. Khalil, E.B.; Morris, C.; Lodi, A. Mip-gnn: A data-driven framework for guiding combinatorial solvers. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, Vol. 36, pp. 10219–10227.
15. Cantürk, F.; Varol, T.; Aydoğan, R.; Özener, O.Ö. Scalable Primal Heuristics Using Graph Neural Networks for Combinatorial Optimization. *Journal of Artificial Intelligence Research* **2024**, *80*, 327–376.
16. Zhang, J.; Liu, C.; Li, X.; Zhen, H.L.; Yuan, M.; Li, Y.; Yan, J. A survey for solving mixed integer programming via machine learning. *Neurocomputing* **2023**, *519*, 205–217.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.