

Conference Paper Title*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Abstract—Our goal is to develop a model for predicting optimal itineraries based on selectable criteria, which may include minimal carbon footprint, health considerations, and potentially a combination of these factors. The criteria selection is a difficult task as we want relevant ones and ones that are available in public datasets as we will not collect any new data here. Some criteria that we could consider are: minimal carbon footprint, air quality exposure, physical activity. We want our solution to add something new to the existing solutions: balancing multiple criteria and maybe allows weighting them. Graph neural networks (GNNs) are a promising approach for this problem, as they can be dynamic. If we manage to produce a solution to this problem, it could be used in various applications, such as enhancing existing apps to offer more personalized routing options. However we expect some challenges and limitations, such as data availability, computational complexity, and the need for real-time processing. The GitHub repository for this document can be found at <https://github.com/>

Identify applicable funding agency here. If none, delete this.

Deldrel/StageING4.

CONTENTS

I	Key algorithms and Methods	2
II	Multi-Objective Optimization	2
II-A	Genetic Algorithms	2
II-B	Pareto Optimization	2
III	Graph Neural Networks	2
III-A	Overview of Graph Neural Networks	2
III-B	Types of GNN Architectures	2
III-C	Applications	3
IV	Primary Studies	3
IV-A	First Study	3
IV-B	Second Study	4
IV-C	Third Study	4

I. KEY ALGORITHMS AND METHODS

Pathfinding algorithm are widely used in various applications, network routing, video games, etc. Below is a comparison table of some of the most popular algorithms:

Algorithm	Best for	Time Complexity	Space Complexity	Negative Weights?
Dijkstra's	Shortest paths from a single source in a graph	$O(V^2)$ or $O(V \log V)$	$O(V)$	Yes
A*	Pathfinding with heuristics to speed up the search	$O(E)$	$O(V)$	Yes
Bellman-Ford	Graphs with negative weight edges	$O(VE)$	$O(V)$	Yes
Floyd-Warshall	Shortest paths between all pairs of nodes	$O(V^3)$	$O(V^2)$	Yes

TABLE I: Comparison of Route Optimization Algorithms

II. MULTI-OBJECTIVE OPTIMIZATION

The key algorithms and methods are not sufficient to solve our problem. These algorithms can optimize multiple criteria simultaneously, which is very close to what we want.

A. Genetic Algorithms

These algorithms mimic natural selection, with crossovers and mutations. They are good for finding the global optimum of a function. However, they are not good for real-time applications.

B. Pareto Optimization

Pareto optimization is a multi-objective optimization technique that aims to find the best trade-offs between multiple objectives. It is based on the Pareto principle, which states that for any given problem, some solutions are better than others. This solution is good for our problem as we want to balance multiple criteria. However it is not dynamic, which is a limitation.

Despite their efficiency, the multi-objective optimization algorithms are not good options for us. We want scalable and dynamic solutions, which is why we are considering Graph Neural Networks (GNNs). This section is largely based on the work of [?]. It contains lots of references that we can use to deepen our understanding of the topic.

Overview of Graph Neural Networks
Graph Neural Networks (GNNs) are a class of deep learning models designed to handle graphs as input data. Traditional neural networks assume that the input data is in a Euclidean space, which is not the case for graphs. They also treat inputs as independent of each other, which is not true for graph data. These differences allows GNNs to capture the structure of the graph and the relationships between nodes. They are powerfull tools for tasks such as node classification, link prediction, and graph classification.

B. Types of GNN Architectures

The paper [?] describes four types of GNN architectures where we can classify most other GNNs:

- **Recurrent Graph Neural Networks (RecGNNs)**: They aim to learn node representations with recurrent architectures. They assume nodes are constantly exchanging information with their neighbors.
- **Convolutional Graph Neural Networks (ConvGNNs)**: They generalize the convolution from euclidean space to graph data. The idea is to aggregate information from neighbors to update the node representation.
- **Graph Autoencoders (GAEs)**: They aim to learn a low-dimensional representation of the graph. They can reconstruct the graph from this representation.
- **Spatial-Temporal Graph Neural Networks (STGNNs)**: They are designed for spatio-temporal data. They can learn hidden patterns from spatial and temporal data.

Feature	RecGNNs	ConvGNNs	GAEs	STGNNs
Core Idea	Iterative processing of node information to reach stable states	Generalize convolution operations to graph data	Encode graph structure into temporal data	Integrate both spatial and temporal information
Use Case	Complex relational reasoning, dynamic graph updates	Node classification, community detection	Graph reconstruction, link prediction	Dynamic prediction, traffic forecasting
Advantages	Captures dynamic changes in graph structure	Efficient in learning node representations	Useful for multiple learning tasks	Handles multiple objectives without sacrificing performance on any single objective.
Limitations	High computational cost for large graphs	Limited by neighborhood information	Requires large datasets	Complex model architecture

TABLE II: Comparison of Graph Neural Network Architectures

From these architectures, we can assume that STGNNs are the most relevant for our problem as our data will likely be space and time dependent.

C. Applications

- **Node Classification:** assign a label to each node in the graph.
- **Graph Classification:** assign a label to the entire graph.
- **Link Prediction:** predict the existence of a link between two nodes.
- **Computer Vision:** use GNNs for image segmentation, object detection, etc.
- **Natural Language Processing:** use GNNs for text classification, sentiment analysis, etc.
- **Recommender Systems:** use GNNs to recommend products, movies, etc.
- **Drug Discovery:** use GNNs to predict molecular properties.
- **Traffic Prediction:** use GNNs to predict traffic flow, congestion, etc.

The paper [?] refers to three other papers that used STGNNs [? ? ?] for traffic problems.

IV. PRIMARY STUDIES

A. First Study

A Graph Pointer Network-Based Multi-Objective Deep Reinforcement Learning Algorithm for Solving the Travelling Salesman Problem [?].

- **Context:** This research is based on multi-objective optimization problems and uses the travelling salesman problem as a case study.

With new methods to process sequential and temporal data, notably with GNNs, the authors propose a new algorithm to solve the TSP.

- **Problem:** The paper addresses the need for more sophisticated algorithms that can effectively manage multiple objectives without sacrificing performance on any single objective.
- **Data:** The authors generated their data using the random generator from PyTorch. The training set has 500,000 samples and the validation set has 1,000 samples. Note: we could use the same method if we can properly model our problem and the data we need.
- **Results:** The authors used a similar comparison test than other paper they cited. We can see in the image below that their model (MODGRL) outperforms the other models.

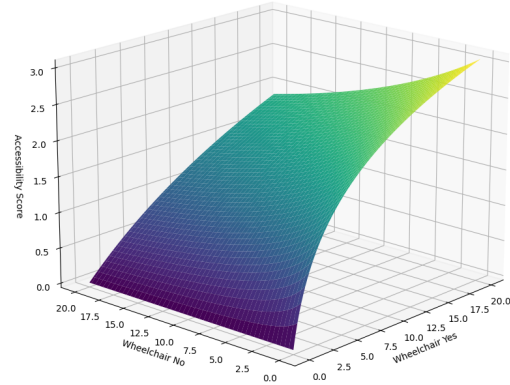


Fig. 1: Comparison of MODGRL with other models

- **Future Work:** The paper emphasizes the potential for further improvement and application of such models to even more complex and varied multi-objective optimization problems. The research suggests that leveraging the advancements in graph neural networks and deep reinforcement learning could lead to significant breakthroughs in various real-world applications requiring multi-objective decision-making.

B. Second Study

Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case [?].

- **Context:** This research is about routing optimization in dynamic and varied network topologies. It integrates Deep Reinforcement Learning (DRL) and Graph Neural Networks (GNNs) to develop a model that can generalize to unseen network.
- **Problem:** DLR solutions struggle to work on unseen topologies. The integration of a GNN allows better adaptation without retraining the model.
- **Data:** They author used synthetic topologies generated with NetworkX in python and real-world topologies from the Internet Topology Zoo [?]. Note: not the first time we see random generated datasets, we could use the same method.
- **Results:** The combination of DRL+GNN outperformed the DRL-only model on unseen topologies. Their solution handled dynamic routing in optical networks with various conditions without retraining.
- **Future Work:** The paper suggests that the model could be further improved on more applications. They say that having a model that could generalize even more could transform the way we design and manage networks.

C. Third Study

Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting [?]. This is one of the papers mentioned in [?], in section III.

- **Context:** This research is about traffic forecasting on road networks, for smart transportation systems. Note: this could be useful if we want to make traffic routing predictions.
- **Problem:** Forecasting traffic is a challenging task due to the complexity of road networks and the dynamic nature of traffic. Traditional methods struggle with long-term predictions because of the spatial and temporal dependencies.

- **Data:** This is why this paper is interesting for us. They used 2 datasets from PyTorch Geometric Temporal [?].

- METR-LA: traffic speed data from loop detectors (207) in Los Angeles.
- PEMS-BAY: traffic speed data from loop detectors (325) in the Bay Area.

- **Results:** Their model is classified as a STGNN by paper [?]. It outperformed traditional and some deep learning models on horizons of 15 minutes, 30 minutes, and 60 minutes.
- **Future Work:** The paper suggests that the model could be used on other spatio-temporal tasks than traffic forecasting. they also mention that it could be adapted to dynamic network structures.

V. ETUDE APPROFONDIE

a cause de la durée du stage, notre objectif est d'améliorer une solution existante en ajoutant des critères supplémentaires. nous nous sommes donc concentrés sur la 3eme étude. malheureusement leur code source n'est pas bien documenté et nous n'avons pas pu le reproduire. nous avons perdu du temps à essayer de comprendre leur code et à le reproduire. mais nous avons tout de même appris de leur solutions et ces connaissances nous seront peut etre utile à l'avenir.