Optimizing UAV-Based Delivery Networks via Graph Convolutional Networks

Van Chau Huy, Hung Do Hoang Duy

Abstract—**In this report we consider a problem that to find maximize satisfaction of user, with the minimun flying cost of series of unnamed aerial vehicles (UAV). A Graph Concolutional Neural networks which is designed to capture the rapid changes of the environment is used to solve this problem.**

I. Introduction

This report focuses on an optimization problem in which multiple UAVs are deployed from distributed warehouses to deliver essential supplies—such as medicine, medical equipment, and daily necessities—to user clusters with varying demands. Each cluster comprises a group of users requiring a specific type of item, and each item is associated with a weight and an importance score. The objective is twofold: first, to determine which items each UAV should carry to maximize user satisfaction—measured by coverage and importance of delivered goods—while minimizing travel distance; and second, to minimize the number of UAVs required under a constraint on average delivery time per cluster.

Traditional approaches to solving such problems often rely on heuristic methods or combinatorial optimization, which struggle with scalability and generalization. To address this challenge, we propose using a Graph Neural Network (GNN)-based supervised learning framework. By representing the entire system as a graph where UAVs and user clusters are nodes and edges capture spatial and demand-based relationships, the GNN learns to encode structural dependencies and make near-optimal assignment decisions efficiently.

II. GCN Power and Problems that could solve using GCN

III. System Formulation

A. UAV Maximize Capacity Formulation

For each UAV m, let $H_m \subseteq \{1, \ldots, N\}$ be the set of item-types it carries. We require

$$\sum_{n \in H_m} w_n \le w_m,$$

where w_n is the weight of one unit of item n, and w_m is UAV m's maximum payload.

B. Minimize Path Weight Formulation

Let $C_m \subseteq \{1, \ldots, C\}$ be the clusters served by UAV m, and $\mathrm{dist}(m,i)$ the Euclidean distance from UAV m's depot to cluster i. We seek to minimize total flight cost

$$\min \sum_{m=1}^{M} \sum_{i \in C_m} \mathrm{dist}(m,i).$$

C. Maximize User Satisfaction Formulation

Each cluster i has C_i users requesting item n_i, with importance score u_{n_i}. Introduce binary $\delta_{mi} = 1$ if UAV m serves cluster i. Then

$$\max \sum_{m=1}^{M} \sum_{i=1}^{C} \delta_{mi} \left(C_i \cdot u_{n_i}\right).$$

Optionally combine with a distance penalty:

$$\max \sum_{m,i} \delta_{mi} \left(C_i u_{n_i}\right) \; - \; \lambda \sum_{m,i} \delta_{mi} \, \mathrm{dist}(m,i).$$

IV. System Model

A. Parameters

- N: numbers of items, each items weight w_n each items importance (bias) u_n.
- C: number of clusters, each cluster i has C_i people and demands item t_i.
- M: number of UAVs, UAV m located at position (x_m, y_m, z_m) and has maximum capacity w_m.

B. Optimizing the solution with GCN

To solve the maximization of user satisfaction (with distance penalty) in a data-driven fashion, we model the assignment problem as a graph learning task and apply a Graph Convolutional Network (GCN).

a) 1) Graph Construction: Define a directed graph $G = (V, E)$ with

$$V = \{u_1, \ldots, u_M\} \,\cup\, \{c_1, \ldots, c_C\},$$

where each UAV u_m and each cluster c_i are nodes. We create an edge $(u_m, c_i) \in E$ for every potential service link.

Node features. Each UAV node u_m is endowed with

$$\mathbf{x}_{u_m} = \bigl[x_m,\, y_m,\, z_m,\, w_m,\, 0,\, 0\bigr]^\top \in R^6,$$

and each cluster node c_i with

$$\mathbf{x}_{c_i} = \bigl[x_i,\, y_i,\, 0,\, 0,\, C_i,\, n_i\bigr]^\top \in R^6,$$

where w_m is UAV payload, C_i user count, and n_i item index.

Edge features. For each (u_m, c_i), let

$$d_{mi} = \| \, (x_m, y_m, z_m) - (x_i, y_i, 0) \| \,, \quad t_{mi} = \frac{d_{mi}}{v}, \quad \omega_{mi} = C_i \, w_{n_i}, \quad u_m$$

and set $\mathbf{e}_{mi} = [\, d_{mi},\, t_{mi},\, \omega_{mi},\, u_{mi} \,]^\top \in R^4$.

b) 2) GCN Architecture: We employ a two-layer GCN followed by a linear read-out on UAV nodes:

$$\mathbf{H}^{(1)} = \sigma\big(\hat{A}\,\mathbf{X}\,W^{(1)}\big), \quad \mathbf{H}^{(2)} = \sigma\big(\hat{A}\,\mathbf{H}^{(1)}\,W^{(2)}\big),$$

$$\mathbf{Z} = \mathbf{H}^{(2)}\,W^{(o)}, \qquad \widehat{Y}_m = \mathrm{sigmoid}\big(\mathbf{Z}_{u_m}\big) \in R^{C}.$$

Here: - $\mathbf{X} \in R^{(M+C)\times 6}$ stacks all node features. - $\hat{A}$ is the symmetrically normalized adjacency (including self-loops). - $W^{(1)}, W^{(2)} \in R^{6\times H}$ and $W^{(o)} \in R^{H\times C}$ are learnable, with hidden size H. - σ is Sigmoid activation. - We only read out the first M rows $\mathbf{Z}_{u_1}, \ldots, \mathbf{Z}_{u_M}$.

c) 3) Learning and Inference: We frame the problem as a multi-class classification over C clusters for each UAV. Let $y_m \in \{1, \ldots, C\}$ be the ground-truth best cluster for UAV m. We minimize the cross-entropy loss

$$\mathcal{L} \;=\; -\sum_{m=1}^{M} \log\big[\widehat{Y}_m\,[y_m]\big].$$

We train with Adam (learning rate 10^{-3}), batch size 1 (full graph), for 200 epochs.

At inference, each UAV m is assigned to cluster $\arg\max_i \widehat{Y}_m[i]$, producing a near-optimal assignment that maximizes $C_i\,u_{n_i}$ while implicitly accounting for distance through the learned embeddings.

This GCN framework thus learns from data to solve the maximization problem in a single forward pass, offering scalability and adaptability beyond hand-crafted heuristics.

V. Performance Evaluation

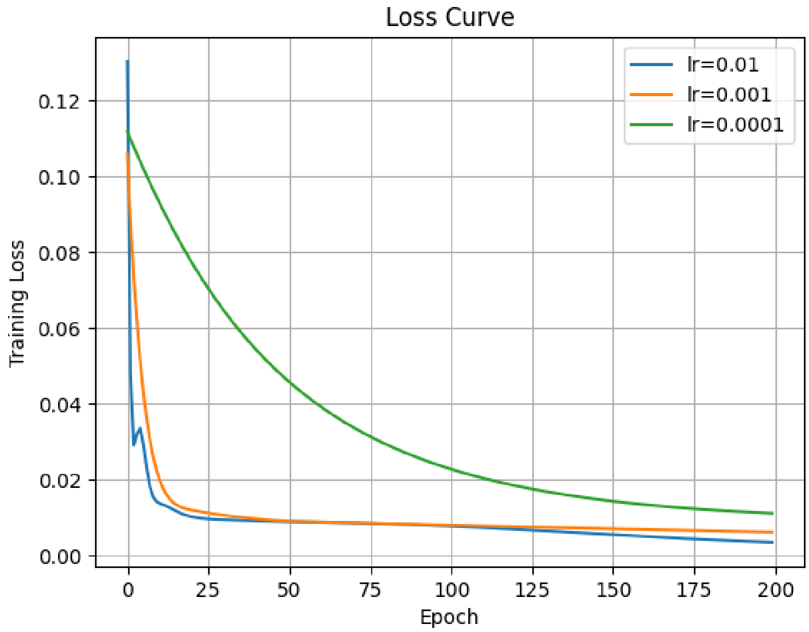

Fig. 1: Training Loss Visualization

Change the activation function of GCN model from relu into sigmoid makes model works better. While learning rate equal 1e-3 is still the best.Sigmoid provides smooth, differentiable curves that may help small or shallow networks learn more stable gradient updates than ReLu.

VI. Conclusion

We have proposed that the optimization of UAV-Based Delivery Networks with Graph Convolutional Neural Network works effectively.

VII. References

A. Throughput-Oriented in 5G networks using GCN - https://arxiv.org/pdf/2505.04894

B. Semi-Supervised Learning using GCN - https://www.alphaxiv.org/abs/1609.02907

Recommendation using GCN - https://www.alphaxiv.org/abs/2503.03313

C. Non-Euclidean Computer Vision - https://arxiv.org/abs/2407.21525

D. Natural Language Processing - https://arxiv.org/abs/2209.15003