Spectral GNNs vs GATs
When attention is (not all you need) the "only" option

Jorge Luiz Franco

Undergraduate Program in Computer Science

Advisor: Prof. Filipe Alves Neto Verri, Ph.D.

Universidade de São Paulo - ICMC

June 17, 2024

Summary

Summary

Motivation

- Graph machine learning can be tracked backwards to the problem of 'learning' on data that is inherently a graph [4, 2] or can be modeled as a graph [6, 7]
- Variety of tasks: node/edge classification, link prediction, graph partitioning, which rely on learning representations from graph-structured data.
- Techniques were most developed by complex networks researchers.
- **Last decade:** significant shift towards the merging of three main communities: graph signal processing, deep learning and complex nets.

GraphML paradigms Overview

- Three main learning paradigms: supervised, unsupervised, and semi-supervised learning
- We are interested on the (semi-)supervised learning paradigm, which encompasses a variety of techniques designed to leverage learning to (partially-)labeled data [5, 1]
- Focus in the subset of graph elements (nodes,edges,graph structure) prediction(classification/regression) methods.
- Consider the division of the field into `traditional` graph learning: topological graph measures into tabular ML algorithms [3, 4], and `deep` graph learning

Introduction to Last-Mile Delivery Drones

Figure: Drones Congestion in a high-traffic Last Mile Delivery context.
Source: [?]

Introduction to Last-Mile Delivery Drones

Last Mile Delivery Drones (LMDD)

- Heterogeneous research area:
  - Combining drones and trucks.
  - Linear integer modeling.
  - Fuzzy logic for uncertainties.
  - Multi-objective optimization.
  - Exclusive drone-based solutions.
- **Complex Systems Decentralized Approach**:
  - Tradable permit model for multi-agent airspace use [?].

Related Work and Centralized Control

- **Necessity of Air Traffic Management**:
  - Most centralized models don't address collision avoidance [**?**].
  - Ensuring optimal path planning and efficient airspace control.
- **Centralized Control and UTM**:
  - **Centralized Control**:
    - Federal Aviation Administration (FAA) and NASA's Unmanned Aircraft System Traffic Management (UTM) [**?**].
    - Ensures organized, legislative-backed airspace control.
  - **Decentralized Models**:
    - Novel but complex in scalability and regulatory compliance.

Proposed Approach

- **Aispace Control and MAPF Approach**:
  - Multi-Agent Path Finding (MAPF) is a solution for addressing spatial characteristics and collision avoidance.
- **Proposed Strategy**:
  - Employing MAPF strategy for Last Mile Delivery Drone problem.
  - Three approaches: MILP, heuristic and hybrid.
  - Use of prioritized planning [?] and conflict-based search [?] to manage computational complexity in the heuristic.
  - Comparing the MILP with the heuristic.
  - Qualitative comparison between centralized and decentralized approaches.

Summary

LMDD as MAPF

- Tasks for each agent (drone) described by tuples:

$$task_i :- \{(x_{\text{start}_i}, y_{\text{start}_i}), (x_{\text{goal}_i}, y_{\text{goal}_i})\}, \forall \text{ drone } d_i$$

- Grid bounds: $x \leq X, y \leq Y$.
- Goal: find paths P_{d_i} making the path length as short as possible.

Path Planning and Constraints

- Drones limited to four principal movements: upward $(x, y+1)$, downward $(x, y-1)$, rightward $(x+1, y)$, and leftward $(x-1, y)$.
- Adjacency: (x_1, y_1) is adjacent to $(x_2, y_2) \iff |x_2 - x_1| + |y_2 - y_1| = 1$.
- New decision variable: t_{begin} (arrival time).

LMDD Goals

- Given time T, minimize the sum of distances of all drones.
- Allow drones to wait in cells and choose entry time.
- Non-weighted distances make the problem easier than standard MAPF.

Network Flow Problem

- MAPF is equivalent to multi-commodity minimum cost maximum flow problem [?].
- Visualized as Network Flow problem.
- The number of drones is a max flow in the time-expanded net

MAPF as Network Flow Problem

- Transformation into a time-expanded network [?].

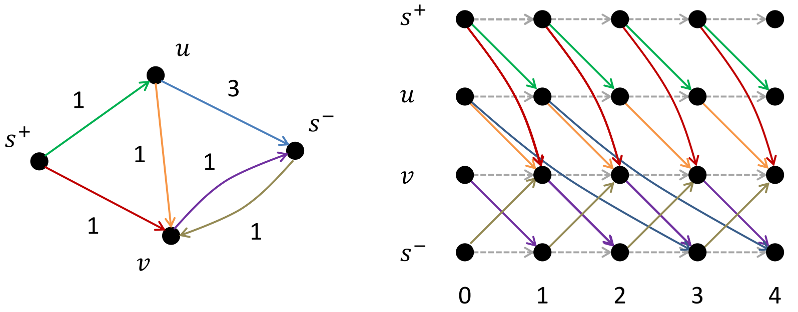


Figure: Time-expanded network representation (sourced from [?])

Multi-Commodity Flow Formulation

- Each drone is a separate commodity flowing from start to goal node.
- Ensuring each drone reaches its goal within a given time horizon T.

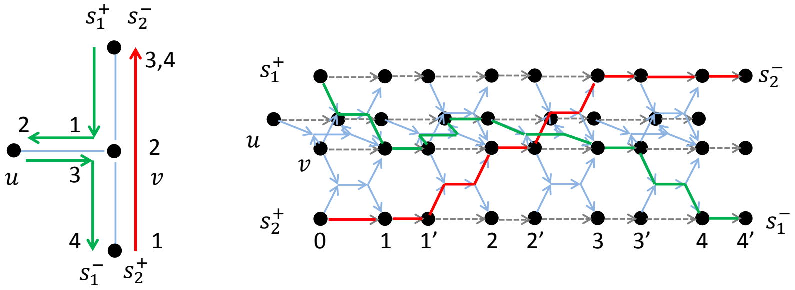


Figure: Equivalence of MAPF to multi-commodity network flow (sourced from [?])

Summary

MILP Model - Graph Modelling

- Proposal of a Mixed-Integer Linear Programming (MILP) model to solve the problem.
- Graph construction representing spatio-temporal permits shared by the drones.
- Modeled as a directed graph (digraph) $G = (V, A)$.
- V: set of nodes representing airspace and virtual drone locations.
- A: set of directed arcs representing permitted transitions between nodes.

MILP Model - Virtual Nodes

- Source node b_k: Starting point of drone k's mission.
- Sink node e_k: Ending point of drone k's mission.
- Focus on spatial topology, omitting temporal component.

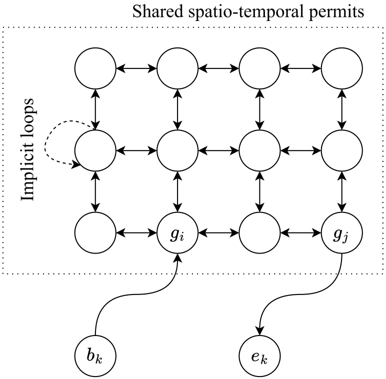


Figure: Graph modelling.
Source: The authors.

MILP Model - Parameters

- T: Maximum time allowed for the mission.
- b_k: Initial virtual vertex representing the initial position of drone k.
- e_k: Final virtual vertex representing the final position of drone k.
- $\mathcal{R}$: Set of drones.
- $\mathcal{G}$: Digraph $(\mathcal{V}, \mathcal{A})$ representing the airspace.
- $\mathcal{V}$: Set of vertices of $\mathcal{G}$.
- $\mathcal{B} \subset \mathcal{V}$: Set of initial virtual vertices b_k.
- $\mathcal{E} \subset \mathcal{V}$: Set of final virtual vertices e_k.
- $\mathcal{S}$: Set $\mathcal{V} \setminus (\mathcal{B} \cup \mathcal{E})$.
- $\mathcal{A}$: Set of arcs $(i, j) \in \mathcal{A}$ of $\mathcal{G}$.

MILP Model - Variables

- Decision Variables:
  - $x_{i,j,t}^{k} = 1 \iff$ drone k jumps from i to j at time t.
- Indices:
  - k: Drone $\implies k \in \mathcal{R}$.
  - t: Time $\implies 1 \leq t \leq T$.
  - i, j, l: Vertices $\implies i, j, l \in \mathcal{V}$.

MILP Model - Objective Function

- Minimize the total sum of the number of drone movements:

$$\min \sum_{k \in \mathcal{R}} \sum_{t=1}^{T} \sum_{(i,j) \in \mathcal{A}:\ j \notin (\mathcal{E} \cup \mathcal{B})} x_{i,j,t}^{k}$$

- Minimize the total number of drone movements, counting $n-1$ jumps for each drone that performs n jumps.

MILP Model - Constraints

- Ensure each drone starts its mission:

$$\sum_{t=1}^{T} \sum_{j \in \mathcal{S}} x_{b_k,j,t}^{k} = 1, \quad \forall k \in \mathcal{R}$$

- Flow conservation:

$$\sum_{j \in \mathcal{V}} x_{i,j,t-1}^{k} = \sum_{l \in \mathcal{V}} x_{j,l,t}^{k}, \quad \forall j \in \mathcal{V}, \forall k \in \mathcal{R}, \forall t \in \{2, \dots, T\}$$

MILP Model - Constraints (Cont.)

- Border condition at time $t = 0$:

$$x_{i,j,0}^{k} = \begin{cases} 1, & \text{if } i = b_k \wedge j = b_k, \\ 0, & \text{otherwise.} \end{cases} \quad \forall k \in \mathcal{R}, \forall (i,j) \in A$$

- Mutual exclusion of vertex occupation:

$$\sum_{k \in \mathcal{R}} \sum_{j \in \mathcal{V}} x_{i,j,t}^{k} \leq 1, \quad \forall j \in \mathcal{V}, \forall t \in \{1, \ldots, T\}$$

MILP Model - Mission Accomplishment

- Ensure each drone completes its mission:

$$\sum_{t=1}^{T} \sum_{i \in \mathcal{S}} x_{i,e_k,t}^{k} \geq 1, \quad \forall k \in \mathcal{R}$$

Heuristic Approach

- Utilize distance measure as heuristic metric [?].
- Organize drones in ascending order (prioritized planning) based on start and end points.
- Employ iterative Breadth-First Search (BFS) on temporal graph.
- Dynamic constraints update of occupied positions (conflict-based search).
- Combine heuristic sorting and iterative BFS for efficient path planning and adaptability.

Algorithm Notation

Table: Notation used in the Algorithm.

Notation	Definition
$\mathcal{V}$	Set of vertices in the graph : (i, j, t)
$\mathcal{E}$	Set of edges
$\mathcal{D}$	Set of drones
$\mathcal{S}$	Set of already scheduled vertices
$\mathcal{P}_d \subseteq \mathcal{V}$	Path of drone d
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Temporal Graph

Heuristic Algorithm Steps

1. **Drones Sorting**: Ascending sort using Euclidean Distance.

$$\mathcal{D}_{\text{sorted}} = \text{sort}(\mathcal{D}, \text{heuristic}) \tag{1}$$

2. **Path for Each Drone**: Compute path P_d using BFS on graph $\mathcal{G}$.

$$\forall d \in \mathcal{D}_{\text{sorted}} : \quad \mathcal{P}_d = \text{BFS}(\mathcal{G}, d) \tag{2}$$

3. **Constraints Update**: Update set of already scheduled vertices $\mathcal{S}$.

$$\mathcal{S} = \mathcal{S} \cup \bigcup_{d \in \mathcal{D}_{\text{sorted}}} \mathcal{P}_d \tag{3}$$

Algorithm Visualization

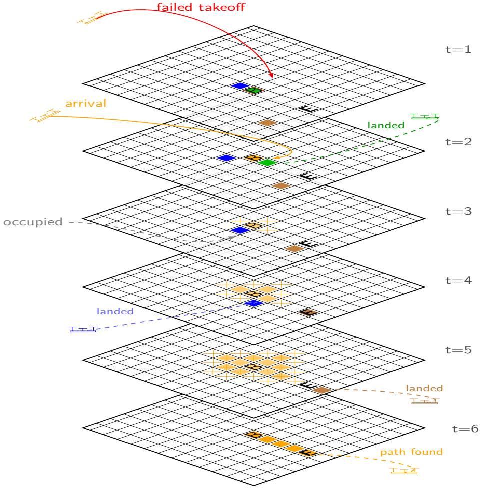
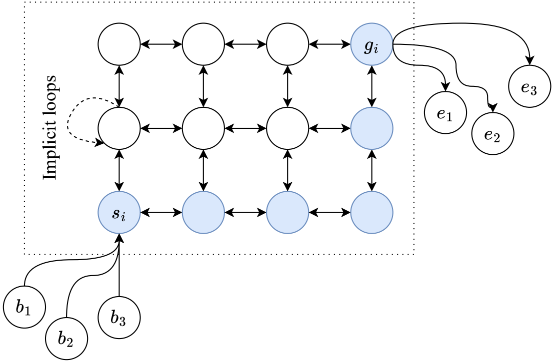


Figure: Algorithm visualization.
Source: The authors.

Complexity Analysis and Boundedness

- Worst-case complexity: $\mathcal{O}((N+M)KNM\log((N+M)KNM))$.
- Approximation: $\mathcal{O}(N^3K\log(N^3K))$ for square grids.

Routing 3 drones in shared space



Hybrid Methodology

- **Heuristic Solution Generation**
  - Quickly generates an initial feasible solution.
  - Determines a plausible time horizon $T_{\text{heuristic}}$.
- **MILP Model Refinement**
  - Uses $T_{\text{heuristic}}$ and initial feasible solution from heuristic.
  - Refines the solution to ensure global optimality.
- **Advantages**
  - Combines computational speed with solution accuracy.
  - Skips multiple iterations to determine T, reducing computational expense.

Summary

Results

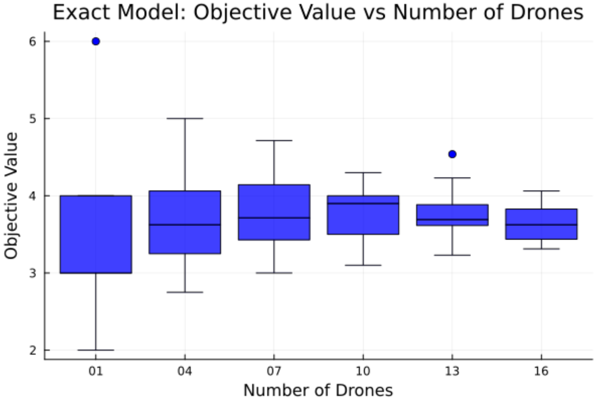


Figure: Exact Model Objective. Source: The authors.

Results

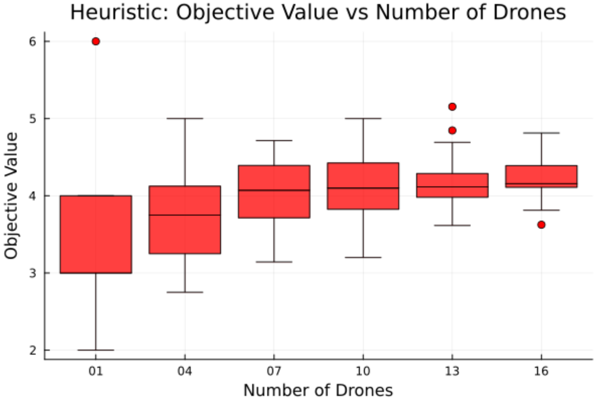


Figure: Heuristic Objective. Source: The authors.

Results

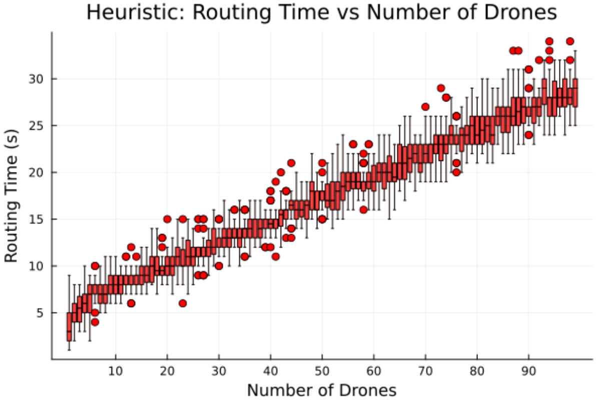


Figure: Heuristic Routing Time (T) corresponding to Figure **??**. Source: The authors.

Summary

Conclusion

Advancements

- Addressing the Last Mile Delivery Drones problem through a MAPF approach ;
- Ensuring collision avoidance and efficient airspace control.
- Significant advancement over previous algorithms with guaranteed finite and bounded polynomial time convergence for the heuristic

Limitations

- Does not address real world problems like energy, fuel and speed of the drones;
- The experiment using binary search to find the T for the MILP in $\mathcal{O}(\log(T))$ is not conducted;
- Small grid sizes in experiments;

Future Works

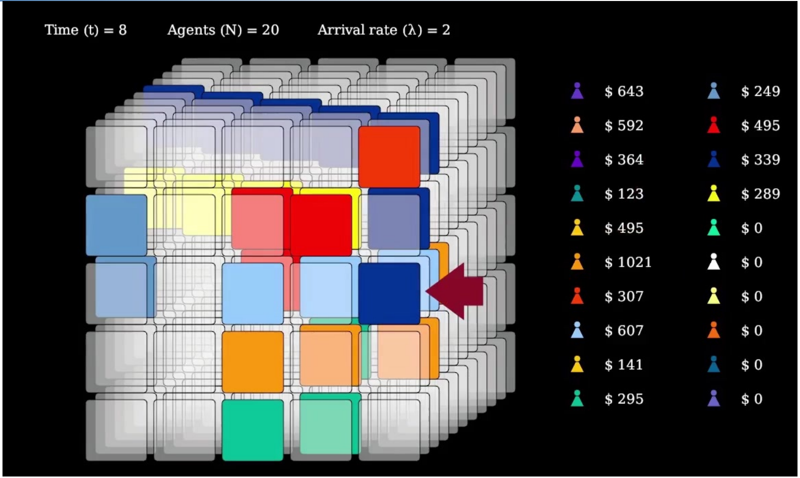


Figure: Tradable Permit Model in the Decentralized LMDD. Source: [?]

Future Works

Decentralized Model from [?]

- naive agents
- greedy random choices

Decentralized Model with Intelligence

- Reinforcement Learning
- Graph Dynamical System
  - Complex Networks
  - Graph Neural Networks

Summary

References I

- D. R. Amancio, C. H. Comin, D. Casanova, G. Travieso, O. M. Bruno, F. A. Rodrigues, and L. da Fontoura Costa, *A systematic comparison of supervised classifiers*, PLOS ONE, 9 (2014), pp. 1–14.
- I. Chami, S. Abu-El-Haija, B. Perozzi, C. RÃ©, and K. Murphy, *Machine learning on graphs: A model and comprehensive taxonomy*, Journal of Machine Learning Research, 23 (2022), pp. 1–64.
- L. d. F. Costa, *Characterization of complex networks: A survey of measurements*, Advances in physics, 56 (2007), pp. 167–242.
- T. Silva and L. Zhao, *Machine Learning in Complex Networks*, Springer International Publishing, 2016.

References II

- F. A. N. Verri, P. R. Urio, and L. Zhao, *Advantages of edge-centric collective dynamics in machine learning tasks*, Journal of Applied Nonlinear Dynamics, 7 (2018), pp. 269–285.
- F. A. N. Verri and L. Zhao, *High level data classification based on network entropy*, in The 2013 International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1–5.
- J. You, X. Ma, Y. Ding, M. J. Kochenderfer, and J. Leskovec, *Handling missing data with graph representation learning*, Advances in Neural Information Processing Systems, 33 (2020), pp. 19075–19087.