A comparison of the GRA and GMRA algorithms vs a Neural Network

Evan Wells

Abstract- This project provides a quantitative analysis of the Group Role Assignment (GRA), Group Multi-Role Assignment (GMRA), and Neural Network (NN) approaches to solving assignment problems. By training a neural network using supervised learning on a large dataset, we aim to replicate or approximate the performance of the classic GRA algorithms. The comparison will be made using many metrics: Group Score (σ), computation time (ms), accuracy of assignment match rate, scalability and resource consumption.

I. Problem statement/Objectives

The Role Assignment step in Role-Based Collaboration is a crucial algorithmic process that assigns agents to roles through intricate optimization calculations based on limited input data. Depending on the context, it applies either GRA or GMRA algorithms, both of which are based on variations of the Hungarian (Kuhn-Munkres) algorithm with a time complexity of $O(n^3)$. As data size increases, this can result in slow processing and overhead. As a result, we can consider neural networks for their computation speed and scalability.

This project aims to evaluate whether Neural Networks can serve as a computationally lighter alternative while still producing high-quality role assignments. This evaluation will be carried out by comparing assignment accuracy, Group Score (σ), scalability, resource consumption and execution time on previously unseen data.

II. Methodology

Python will be the primary development language due to its large amount of scientific libraries. The GRA and GMRA algorithms will be implemented using the PuLP library, which is optimized for solving linear programming problems. Neural Networks will be developed and trained using either PyTorch or TensorFlow. Comparative evaluation will be done through simulations with controlled datasets and visualized using graphs for clarity. This will provide a holistic view of both methods, as well as the comparison between them.

III. Libraries/APIs

- PyTorch, for NN – https://pytorch.org/
  OR
  TensorFlow, for NN – https://www.tensorflow.org/
- PuLP, for GRA and GMRA algorithm - https://coin-or.github.io/pulp/
- Matplotlib, for visualization – https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html
- NumPy, for computing – https://numpy.org/
- Pandas, for data analysis - https://pandas.pydata.org/

IV. Expected Contributions

This project aims to provide a clear, data-driven comparison between GRA algorithm approaches and neural networks in the context of role assignment problems. While AI and Machine Learning are

often seen as go-to solutions, they are not always the optimal choice - especially when simpler models suffice. At the end of the project, we hope to show whether neural networks offer practical benefits (or drawbacks) when applied to assignment problems with small data, and under what conditions each method excels.

V. Desired Timeline

The timeline can be approximated as follows:

Week 1-2	Implement GRA algorithm, GMRA algorithm and NN. Train NN off training data (random Q, L and L_a)
Week 3-4	Simulation, Verification and Comparison
Week 5-6	Analyse and formalize results
Week 7-8	This project would not take that long
Week 9-10	This project would not take that long
Week 11-12	This project would not take that long
Week 13-14	This project would not take that long
Week 15-16	This project would not take that long

VI. Feasibility

I do not expect many difficulties with this project, as it is an analysis between two algorithms/concepts I'm quite familiar with and comfortable with. Most of the time will be spent ensuring the correlations, results and comparisons are properly visualized through graphs and plots. This project is very feasible.

VII. Datasets Required

Randomized data (Q, L and La) that is solved by GRA or GRMA to label for NN

VIII. Similar\Related Research

This research is related to the Role-Based Collaboration framework, and involves comparing it to the results of generic machine learning.