

FriendHub Social Media App

To setup environment

- pip install rustworkx
- pip install quiskit

To run

1. Navigate to project directory
2. Run 'python quant_algo.py'

This project is a social media network where users can create accounts and make friends with other users on the platform, send messages to them and view messages sent to them. The admin account is represented as a sponsorship account with the

special privilege of being able to see the most influential accounts (accounts with the most followers). This is done using the Quantum Approximate Optimization Algorithm to find the minimum vertex cover over the social network.

As a result, the sponsorship account can send promotional messages to these users to promote products to their followers.

- The social network is modeled as a graph where nodes represent users and edges represent friendships. Rustworkx provides an efficient way to manage and visualize this graph structure.
- The Quantum Approximate Optimization Algorithm is used to find the minimum vertex cover, which identifies the smallest subset of users that cover all friendships. These accounts are the most influential in the network.
- The admin focuses on targeting influential users to promote products, the admin isn't allowed to see all the accounts in the network nor broadcast to all accounts in the network.
- Users can communicate directly with their friends or broadcast messages to all their connections, fostering engagement.

Explanation of Code:

In the code, users are nodes, and friendships are edges, and each user has an associated dictionary in `user_data` to store their friends and messages.

New users can be added dynamically via `add_user` and they can form friendships via `add_friendships`, which are bidirectional edges in the graph. Users can also send direct messages via the `send_message` method to friends or broadcast messages to all their friends.

The admin can view influential accounts and send messages to users.

Explanation of Algorithm

The `qaoa_min_vertex_cover` function uses Qiskit's QAOA to find the Minimum Vertex Cover. Constraints ensure every edge (friendship) is covered by at least one user, where each user is represented as a binary variable (x_i) indicating inclusion in the vertex cover. The objective is to minimize the sum of selected users. The problem is converted to a Quadratic Unconstrained Binary Optimization (QUBO) format and QAOA is run with the COBYLA optimizer and a statevector simulator.

How the Algorithm is used

The graph of users as nodes and friendships as edges is converted into a QUBO problem and then QAOA is run on a quantum simulator to find the optimal solution which is the minimum number of nodes which cover all the edges. The admin (sponsor account) then uses this to target influencers effectively without having to message every person individually.

Technical Details

Libraries Used:

- Rustworkx: For graph management.
- Qiskit Optimization: For formulating and solving the optimization problem.
- Qiskit Aer: For simulating quantum circuits.

Graph Structure:

- Nodes represent users.
- Edges represent friendships with optional weights