Jp Booth

11/28/23

The purpose of these scripts is to download the large honeypot data sheet stored on a Splunk server in the Marist ECRL lab, then run a selection of the honeypot data through Max Cut algorithms using quantum computer simulators to generate ideal Max Cut solutions.

Requirements

* A Splunk Server
  + Our current server (Splunk 9.1.1) is running RHEL9 Linux in the Marist ECRL lab and to access it you must be either on the Marist network or in it via the Marist Cisco VPN
* Python version 3.11+
  + Make sure python is in your PATH environment, and make sure PIP is installed as well. If PIP is not installed, you can learn to get it [here](https://pip.pypa.io/en/stable/installation/)
* An IDE, preferably the latest version of visual studio
* Pip install all dependencies including
  + csv
  + random
  + \_\_future\_\_
  + future
  + urllib
  + httplib2
  + splunklib
  + getpass
  + xml.dom
  + time
  + re
  + qiskit\_ibm\_runtime
  + qiskit
  + qiskit\_aer
  + qiskit\_optimization
  + splotly
  + matplotlib
  + pandas
  + numpy
  + network
  + time
  + copy
  + typing
  + datetime
  + humanfriendly
  + os
  + It’s possible that more libraries could be prompted to import, in which case visual studio should describe which libraries need importing
* Finally, the scripts from this GitHub repository (do not edit the folder)

Downloading a data sheet

The SplunkConnector script can download the Honeypot Data sheet in csv format from Splunk, but requires the username and password for the Marist Splunk instance running on 10.11.17.220. If the you want to reformat the code for use in another Splunk server, simply change the HOST ip and the query to a csv formatted data sheet you need.



The data sheet we used can be found in this GitHub repository and Honeypot\_Data.csv

Running the Quantum Algorithms for Max Cut

Running the Capping Quantum script should allow for options to run groups in visual studio. Running all groups is fine, just make sure to run with your python version (3.11 was used in testing)



First the code will ask you to choose the directoru you want the results to go, it will create a subdirectory here with a timed stamp. Make sure to end with a “\”.



Created file: 

Next you will be asked for an input to use ibm\_quantum or ibm\_cloud. Currently the credentials are in for using “q” for ibm\_quantum

A black background with white text

Description automatically generated

Next you will choose a simulator “sim” or real backend. Choose sim unless IBM has public quantum backends open in the future.

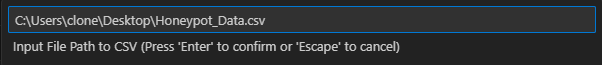
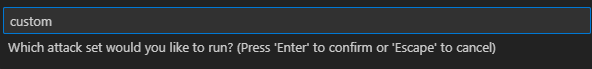
A screen shot of a computer

Description automatically generated

Next you can choose to add a noise simulation onto the quantum computer. For the purpose of this project noise does not affect results and just adds computation time, so “n” should generally be used.



You will then be prompted to choose an attack set. There are build in attack sets but choosing “custom” will allow you to entire the file path to the csv and extract a new elist.



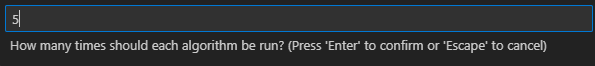
You can see some information in the program about the data that was selected and the elist that was created

A screenshot of a computer

Description automatically generated

In the following prompts you can choose to print the circuit for more results, then choose how many times to run each algorithm.





Progress can be tracked live for the algorithms, and graphs are made both in program and exported to the results folder.

