Programming frameworks 1: Cirq

Monday, September 28, 2020

Rutgers University

Yipeng Huang

Seminar schedule will be released very soon

Class grading rubric will be as follows:

- 2 × 25 points on the presentations

- 2 × 25 points on the programming exercises

- Week-to-week, I may have 5-point assignments to encourage engagement and to spur discussion about the papers.

Seminar presentation grading rubric

15 points: Content

- 5 points: Breadth
- 5 points: Depth
- 5 points: Evaluation and analysis

10 points: Quality of seminar presentation

- 4 points: Structure
- 3 points: Professional preparation and delivery
- 3 points: Timing

Seminar presentation grading rubric: Content (15 points)

5 points: Breadth

Overall, what is this article about?

- Give a high-level overview of the article's background, problem statement, methodology, and major results.

5 points: Depth

After giving an overview of the paper, in the time that you have, dive deep and teach your classmates about a specific detail.

- This may be a specific algorithm, equation, table, or graph.

5 points: Evaluation and analysis

Demonstrate thoughtful evaluation and analysis of the article, which may include:

- What are its impacts on the research field and on society as a whole?
- Who should care about this article, and what might be its long-term impacts?
- What is the deeper insight of why this article is important?
- Is the article believable? What are potential limitations?

Seminar presentation grading rubric: Quality of seminar presentation (10 points)

4 points: Structure

- Do you provide adequate summaries of the whole talk and each section, before going into details?
- Do you outline the content of your talk, so listeners know where you are going?

3 points: Professional preparation and delivery

- Is the presentation speech well rehearsed?
- Are the slides well organized and clear?

3 points: Timing

- Does the talk fit properly in the 30 minutes for the presentation, plus 10 minutes for questions?

Programming lab assignment: QAOA on Cirq
Due Friday, October 30 (then, VQE on Qiskit)

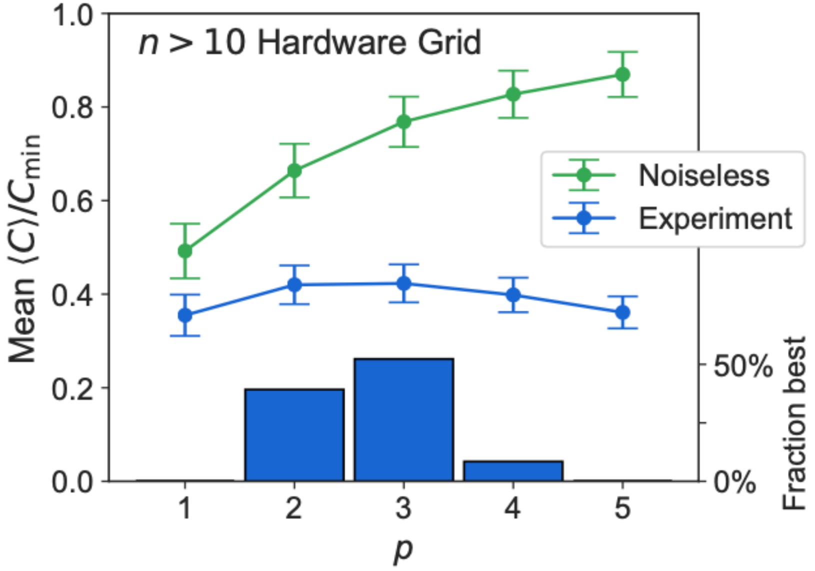

FIG. 5. QAOA performance as a function of depth, p. In ideal simulation, increasing p increases the quality of the solution. For experimental Hardware Grid results, we observe increased performance for $p > 1$ both as measured by the mean over all 10 instances studied for each value of $n \in [11, 23]$ (lines) and statistics of which p value maximizes performance on a per-instance basis (histogram). At larger p, errors overwhelm the theoretical performance increase.

Source: Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor. Arute et al.

For full marks, replicate parts of this experiment in simulation.

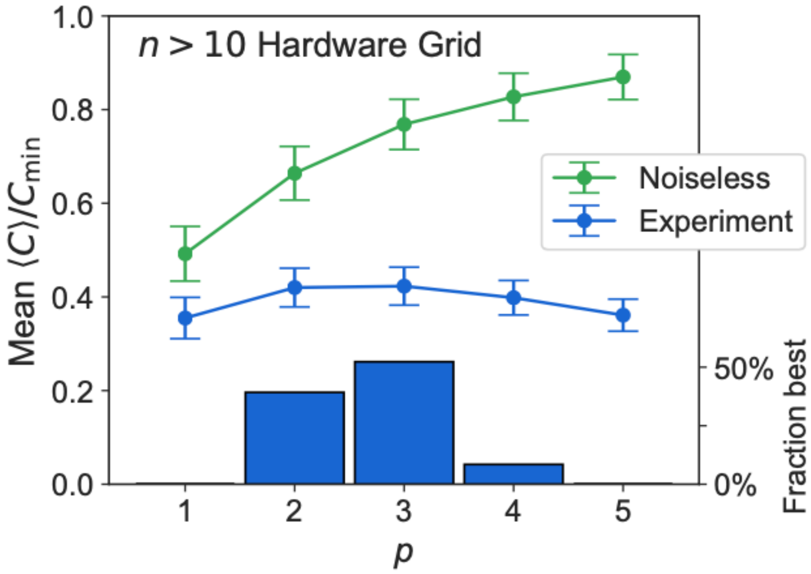

FIG. 5. QAOA performance as a function of depth, p. In ideal simulation, increasing p increases the quality of the solution. For experimental Hardware Grid results, we observe increased performance for $p > 1$ both as measured by the mean over all 10 instances studied for each value of $n \in [11, 23]$ (lines) and statistics of which p value maximizes performance on a per-instance basis (histogram). At larger p, errors overwhelm the theoretical performance increase.

Source: Quantum Approximate Optimization of Non-Planar Graph Problems on a Planar Superconducting Processor. Arute et al.

Accessing Rutgers Computer Science iLab

https://resources.cs.rutgers.edu/docs/new-users/getting-started/

```
ssh netid@ilab1.cs.rutgers.edu
ssh netid@ilab2.cs.rutgers.edu
ssh netid@ilab3.cs.rutgers.edu
```

Beware of ilab4, seems like it is currently missing packages.

You probably want to do work from this directory:

```
cd /common/users/netid/
```

See https://resources.cs.rutgers.edu/docs/file-storage/storage-technology-options/

Cloning Google Cirq

https://github.com/quantumlib/Cirq

Optional: fork if you want to hack and contribute to Cirq:

https://docs.github.com/en/free-pro-team@latest/github/getting-started-with-github/fork-a-repo

```
git clone git@github.com:quantumlib/Cirq.git
cd Cirq
```

Create a Python virtual environment and install Google Cirq

https://cirq.readthedocs.io/en/stable/install.html

```
python3 -m venv ./env
source env/bin/activate
which python3
python -m pip install --upgrade pip
python -m pip install cirq
```

(Optional) Install it as a development environment so you can modify Cirq source code

https://cirq.readthedocs.io/en/stable/dev/development.html

```
python -m pip install -e .[dev_env]
python -m pip install -r dev_tools/conf/pip-list-dev-tools.txt
pytest .
```

Great! Now we can simulate quantum circuits

```
cd examples/
python hello_qubit.py
```

See effect of single H|1>

See effect of HH|1>

See effect of Bell state entanglement circuit

Next, let’s try a minimal QAOA example

wget https://raw.githubusercontent.com/yipenghuang0302/Cirq/master/examples/2020_672_qaoa.py

First, create our example graph

https://networkx.github.io/documentation/stable/tutorial.html

Next, let’s try a minimal QAOA example

Walk through the QAOA algorithm again, this time with code:

https://yipenghuang.com/wp-content/uploads/2020/09/2020_09_23_nisq_algorithms_qaoa_ii.pdf