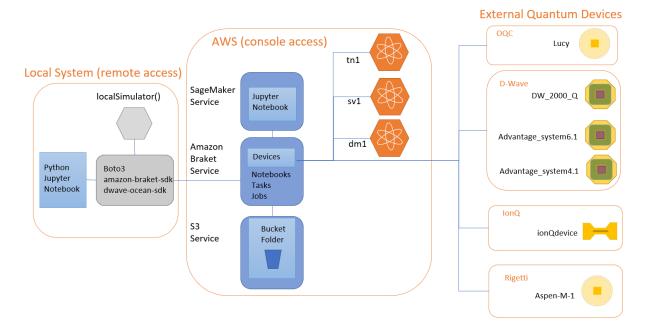
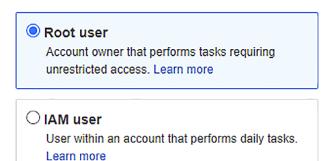
Chapter 1: Setting up Amazon Bracket



Sign in



Root user email address

username@example.com

Next

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

New to AWS?

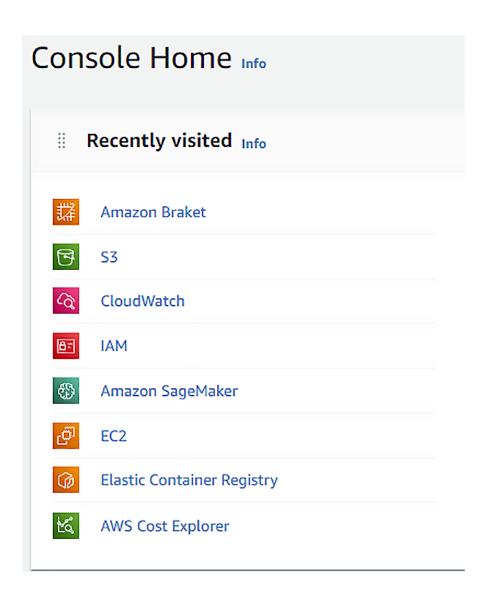
Create a new AWS account

Explore Free Tier products with a new AWS account.

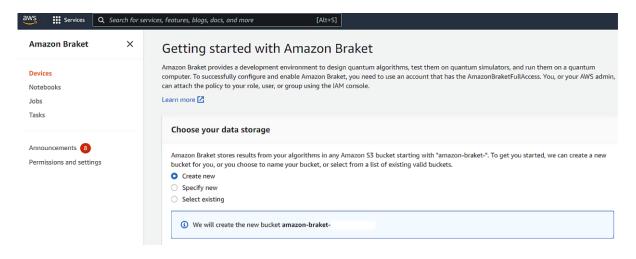
To learn more, visit aws.amazon.com/free.



Root user email address Used for account recovery and some administrative functions AWS account name Choose a name for your account. You can change this name in your account settings after you sign up. Verify email address OR Sign in to an existing AWS account







Account permissions

Amazon Braket creates a service-linked role in your account. The role allows Amazon Braket to access AWS resources on your behalf. The following permissions policy is attached to the role when you enable Amazon Braket. Learn more [7]

▶ Permissions

Terms & conditions

If you use Amazon Braket to access quantum computing hardware operated by one of the third-party hardware providers listed here (each a "Hardware Provider"), you: (1) acknowledge that the Content you provide in connection with your use of Amazon Braket may be processed by the Hardware Provider outside of facilities operated by AWS; and (2) authorize AWS to transfer such Content to the Hardware Provider for processing.;

I have read and accepted the above terms & conditions.

Amazon Braket

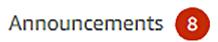


Devices

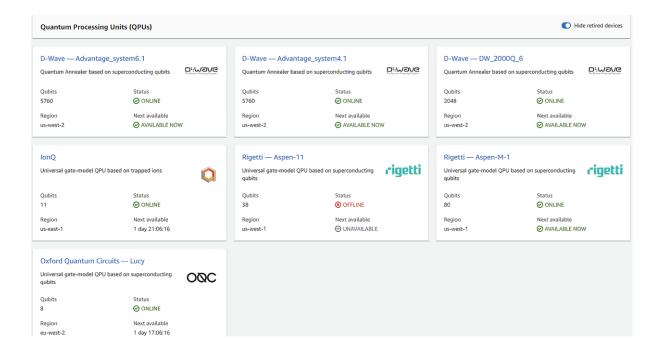
Notebooks

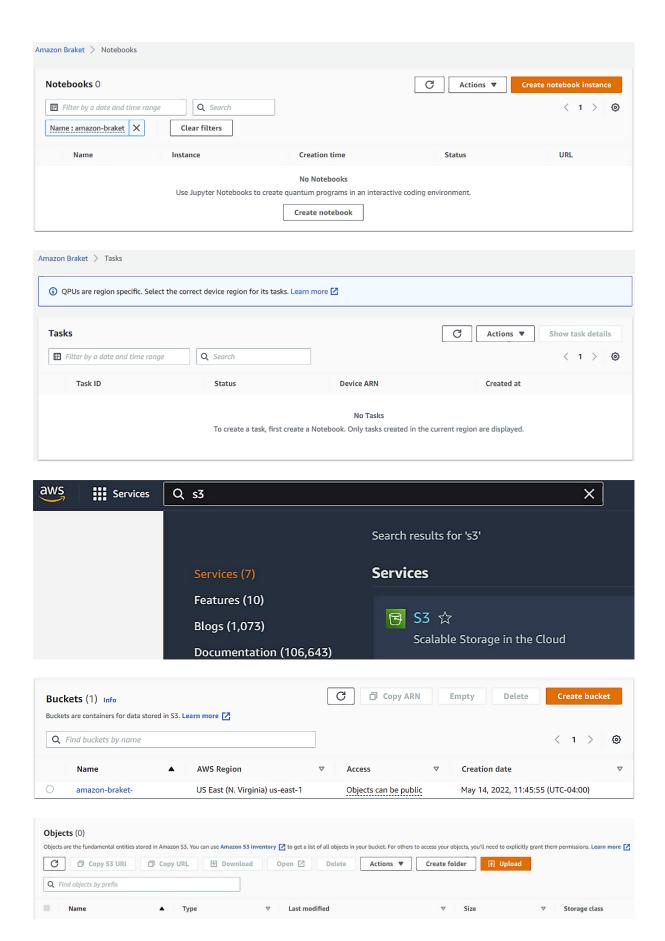
Jobs

Tasks



Permissions and settings

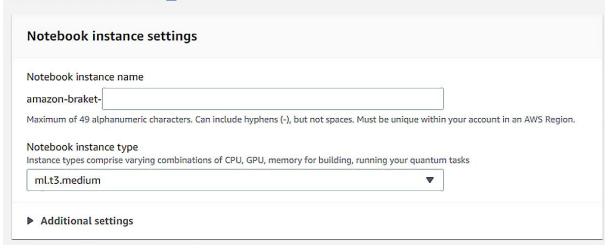


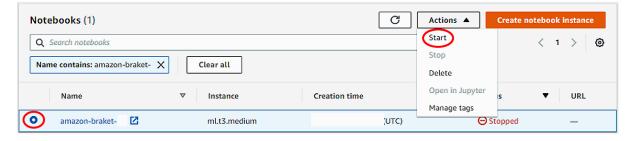


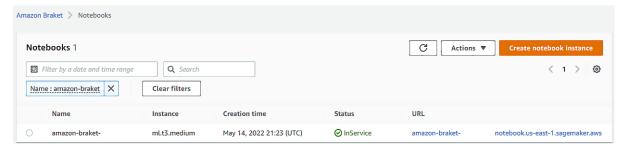
Amazon Braket > Notebooks > Create notebook instance

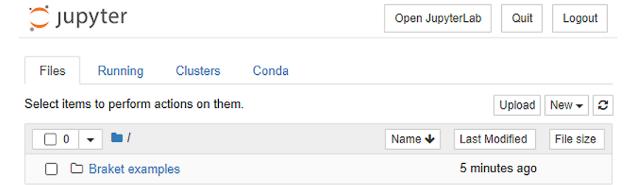
Create notebook instance

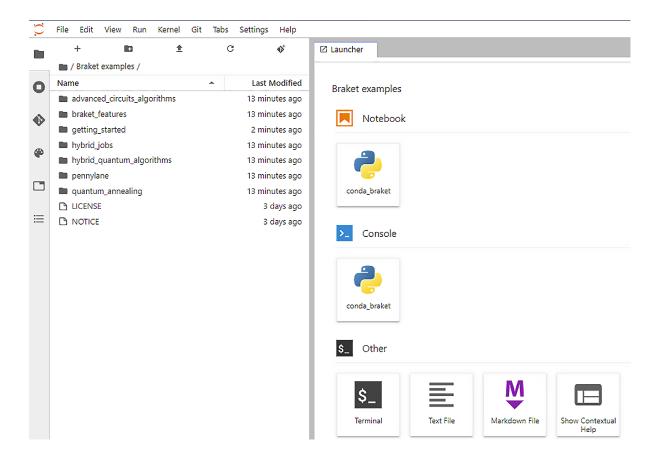
Amazon Braket provide fully managed notebook instances that run Jupyter. The notebook instances come preinstalled with the Amazon Braket SDK and include tutorials and example algorithms. Amazon Braket notebooks are based on SageMaker Notebook instances. Learn more [2]











Server stopped

You have shut down the Jupyter server. You can now close this tab. To use JupyterLab again, you will need to relaunch it.

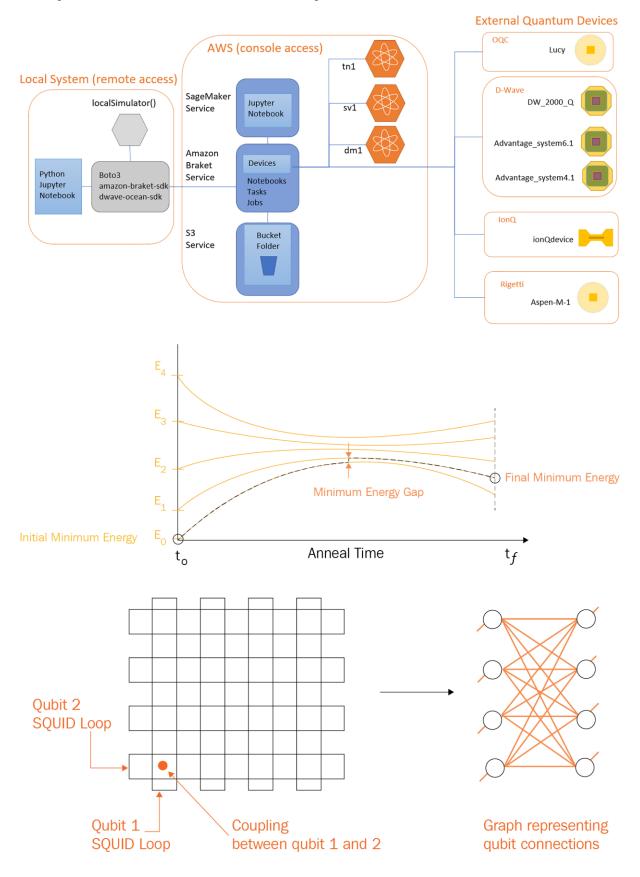
Server stopped

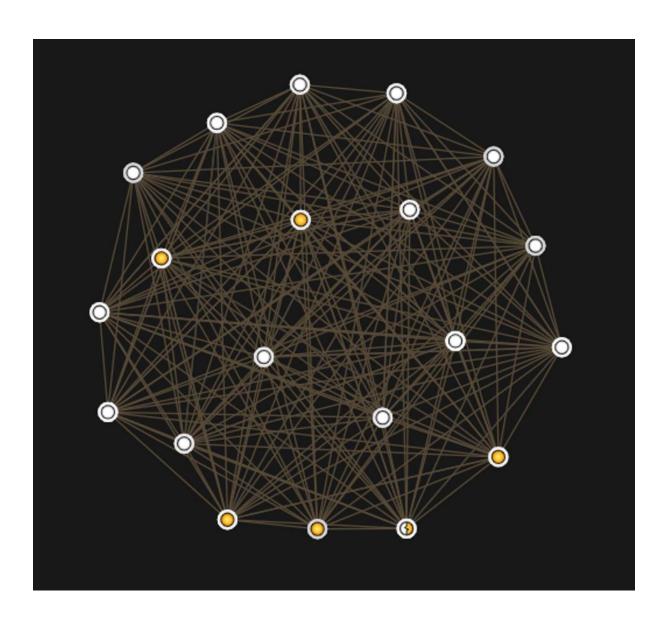
You have shut down Jupyter. You can now close this tab. To use Jupyter again, you will need to relaunch it.

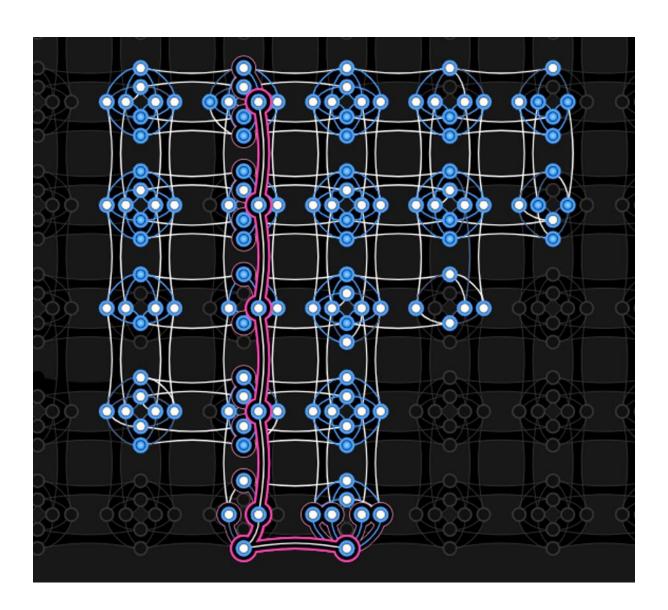


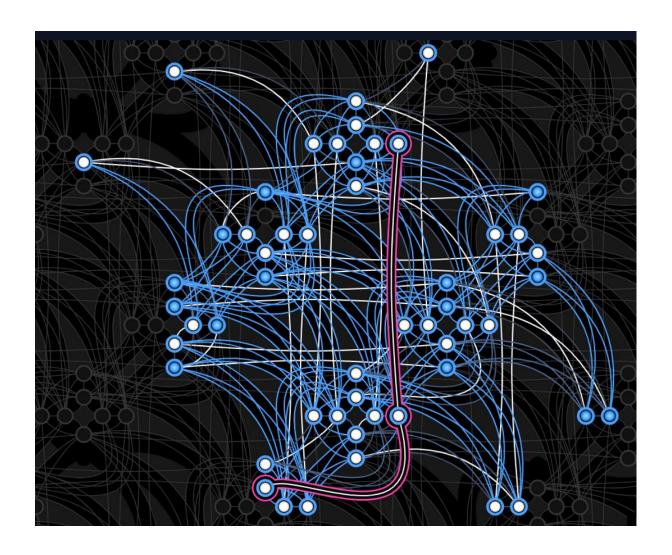


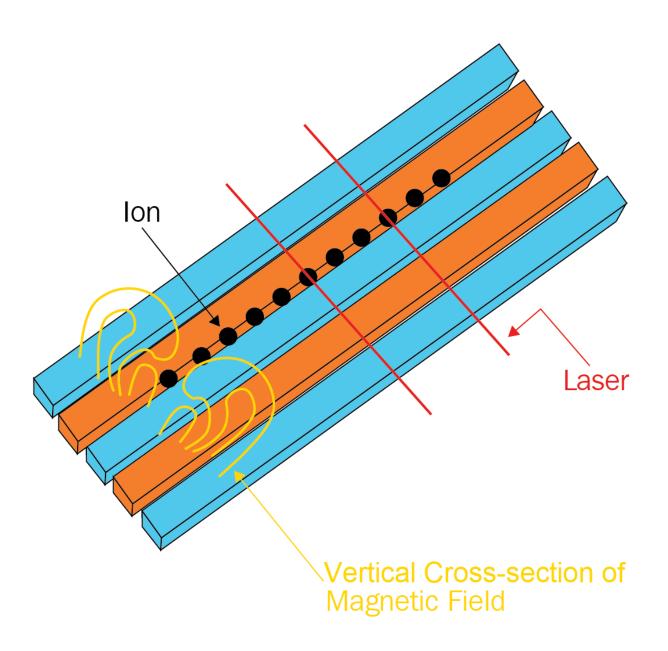
Chapter 2: Bracket Devices Explained

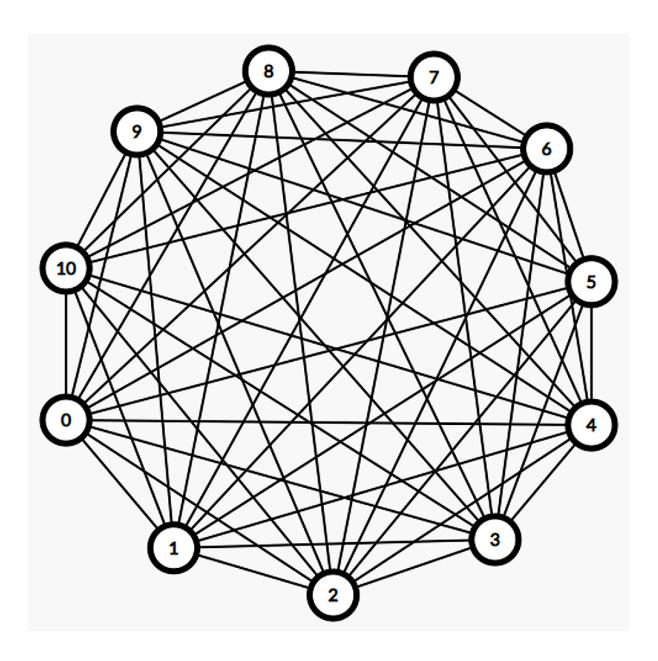






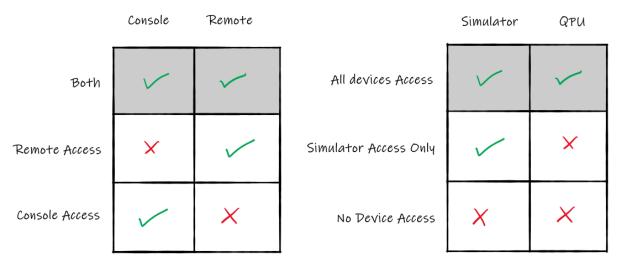






Chapter 3: User Setup, Tasks, and Understanding Device Costs

Security Considerations



Identity and Access Management (IAM)

×

|--|--|

Dashboard

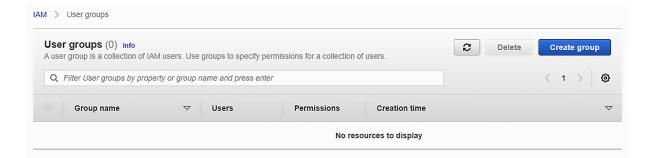
Access management

User groups

Users

Roles

Policies



IAM > User groups > Create user group

Create user group

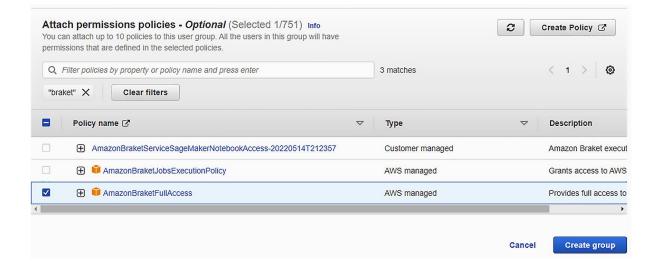
Name the group

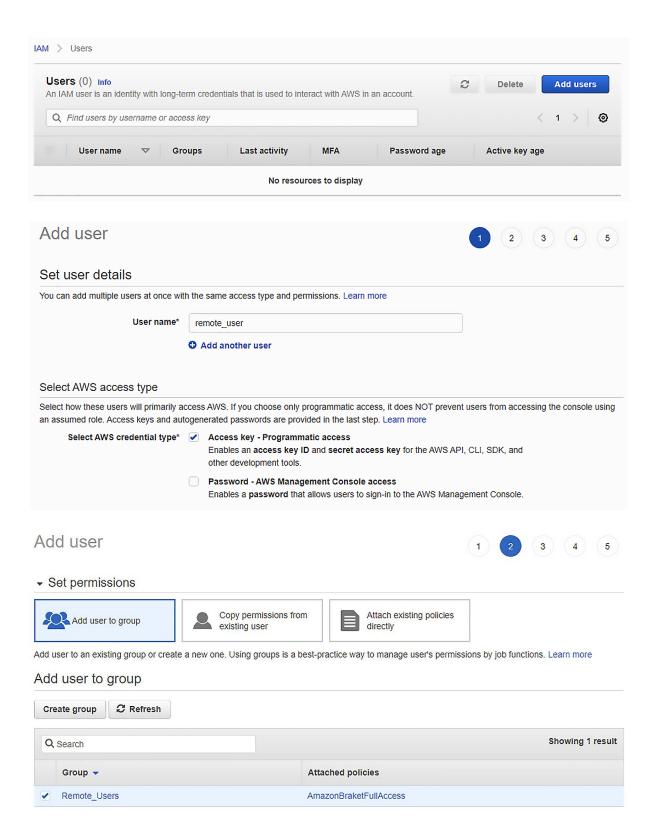
User group name

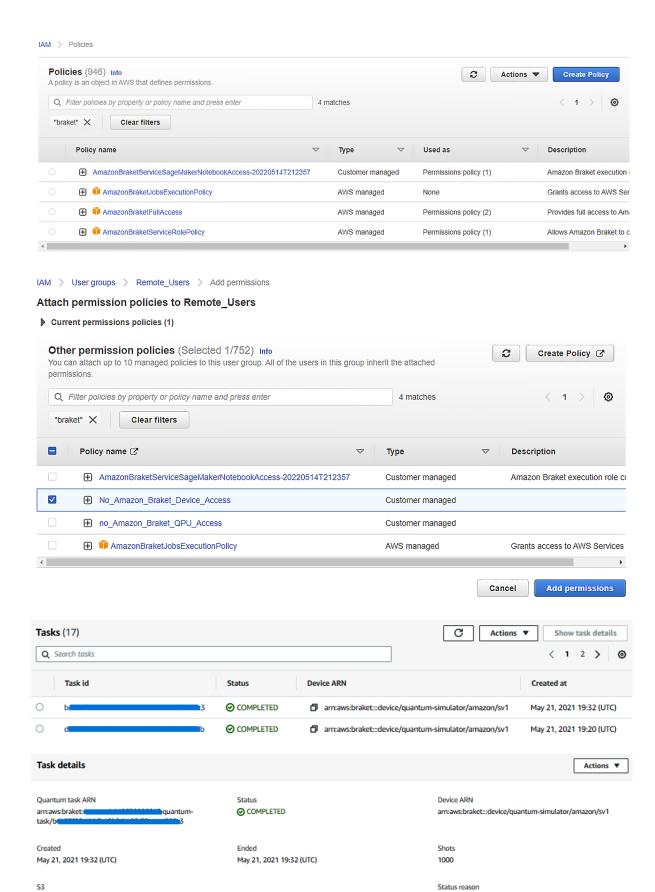
Enter a meaningful name to identify this group.

Remote_Users

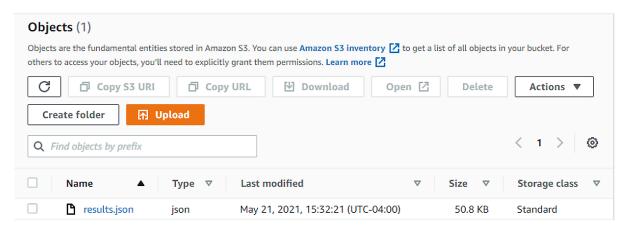
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.



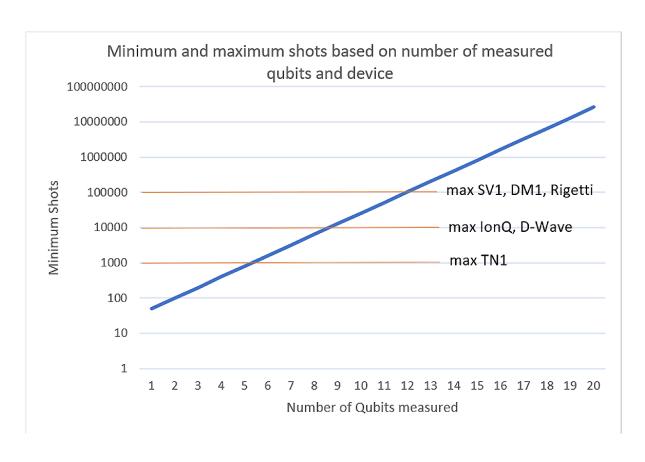




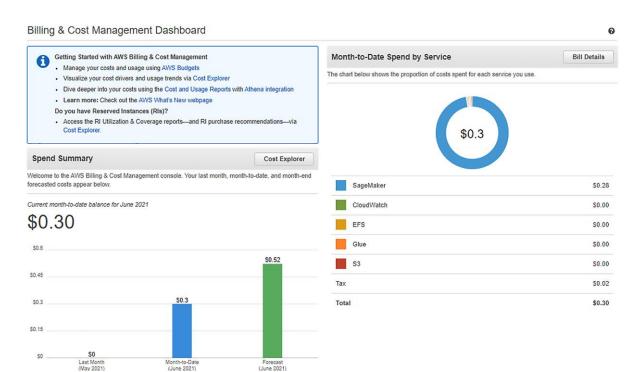
https://s3.console.aws.amazon.com/s3/buckets/amazon-braket-li

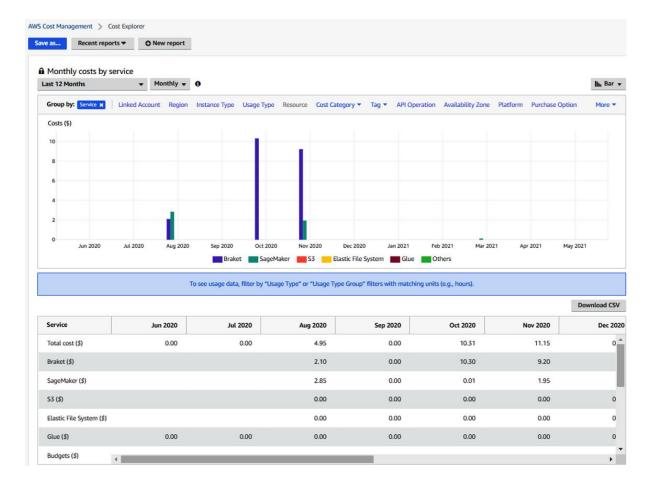


Hardware Provider	QPU family	Per-task price	Per-shot price
D-Wave	2000Q	\$0.30000	\$0.00019
D-Wave	Advantage	\$0.30000	\$0.00019
IonQ	IonQ device	\$0.30000	\$0.01000
OQC	Lucy	\$0.30000	\$0.00035
Rigetti	Aspen-11	\$0.30000	\$0.00035
Rigetti	M-1	\$0.30000	\$0.00035

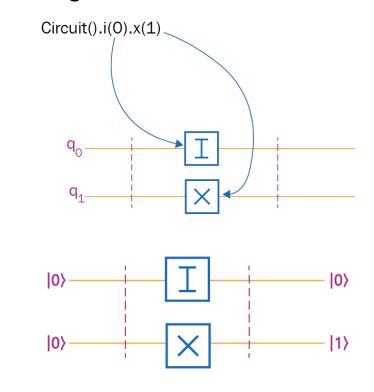


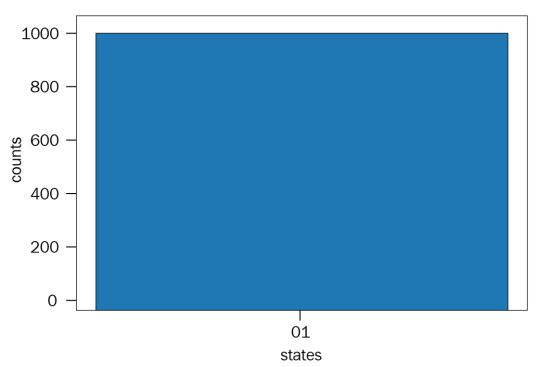
Simulator	cost per minute(\$)		
sv1	\$	0.075	
tn1	\$	0.275	
dm1	\$	0.075	

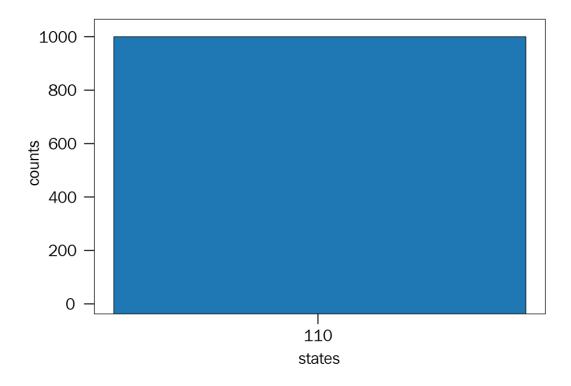


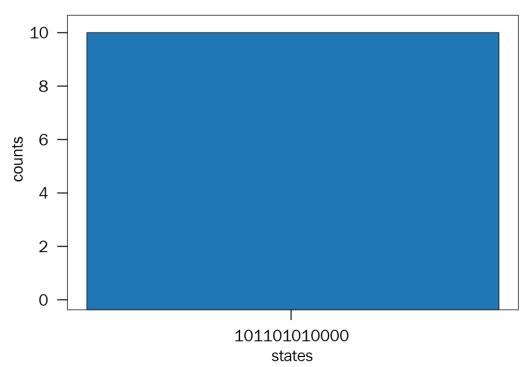


Chapter 4: Writing Your First Amazon Braket Code Sample

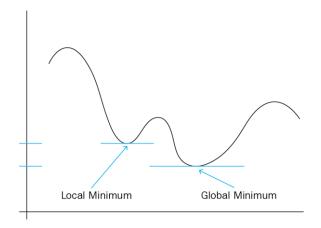


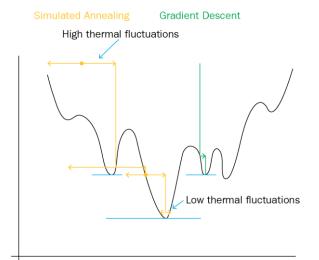


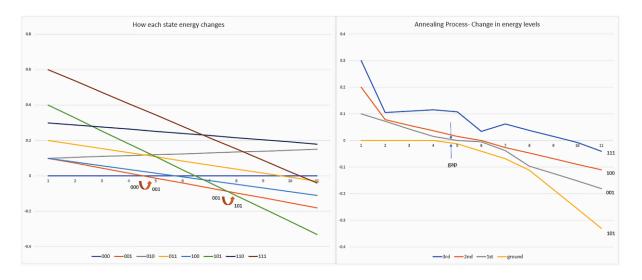


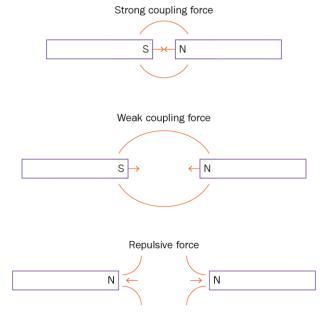


Chapter 5: Using a Quantum Annealer – Developing a QUBO Function and Applying Constraints

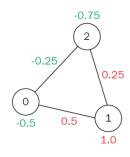








	a _o	a ₁	a ₂
a _o	-0.5		
a ₁	0.5	1.0	
a ₂	-0.25	0.25	-0.75



	$X_{o}=1$			$X_{2} = 1$
		a _o	a ₁	a ₂
$X_0 = 1$	a _o	-0.5		
	a ₁	0.5	1.0	
X ₂ =1	a ₂	-0.25	0.25	(-0.75)

	Alice	Bob	Charlie	David	Eve	Frank
Alice	0.5					
Bob	1	0				
Charlie	0.5	0.25	-0.25			
David	-0.5	-1	0	-0.5		
Eve	-1	-1	0.25	0.5	-1	
Frank	0.25	-0.25	0	0.5	-0.5	-0.5

Step 1:

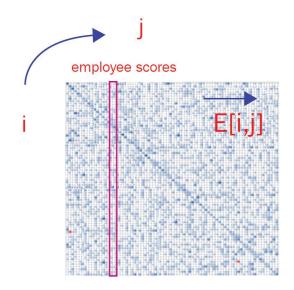
- A. Fully understand the traditional formulation and goal
- B. Create the traditional formulation and find a solution

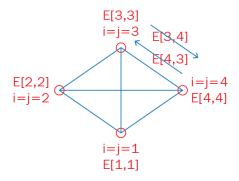
Step 2:

- A. Understand how to translate the traditional formulation into a QUBO
- B. Present the data as a matrix
- C. Add appropriate constraints or penalty
- D. Visualize the landscape

Step 3:

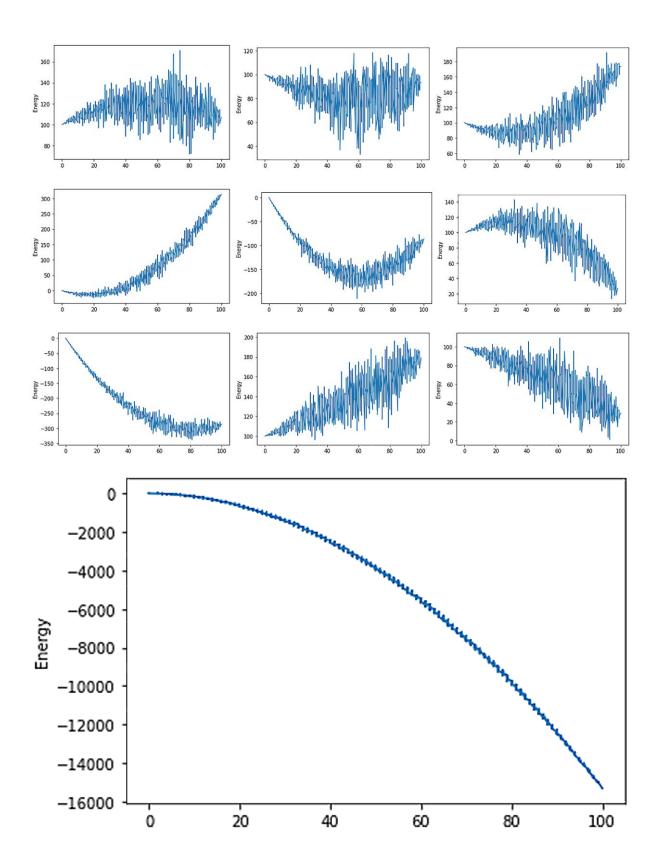
- A. Solve the matrix using Classical solvers or Simulated Annealing
- B. Convert the matrix into a Binary Quadratic Model and solve using Quantum Annealing devices
- C. Validate results by checking with original formulation
- D. Improve results of Quantum devices by modifying embedding and parameters

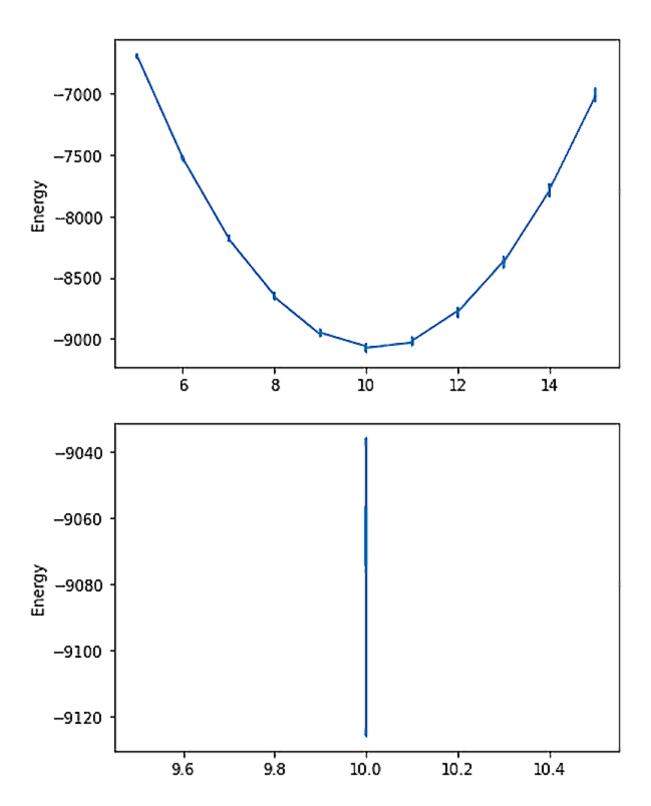




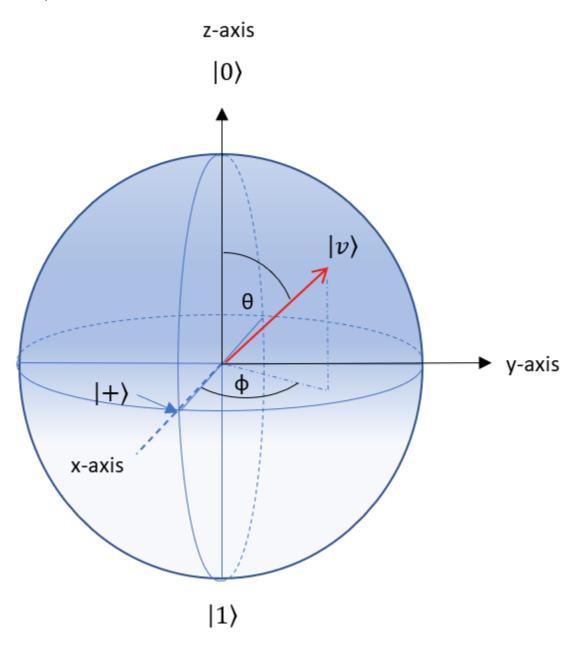
- 1. Total employees 100, (i=0...i=99)
- 2. E[i,j] not always equal to E[j.i] (except when i=j)
- 3. E[i,j] is the score by employee i given to employee j when i is not equal to j, and is the score given by manager to employee when i=j (eg, E[i,i])
- 4. E[i,j] values range from 0 to +5
- 5. Final Team employee count constrain to 10
- 6. If final team has employees 1,9 then the total team score is E[1,1]+E[9,9]+E[1,9]+e[9,1]
- 7. Team Score = Sum ((manager employee terms) + (employee employee terms))
- 8. Goal: Maximize Team score
- 9. Assume weight of manage employee value = employee employee value

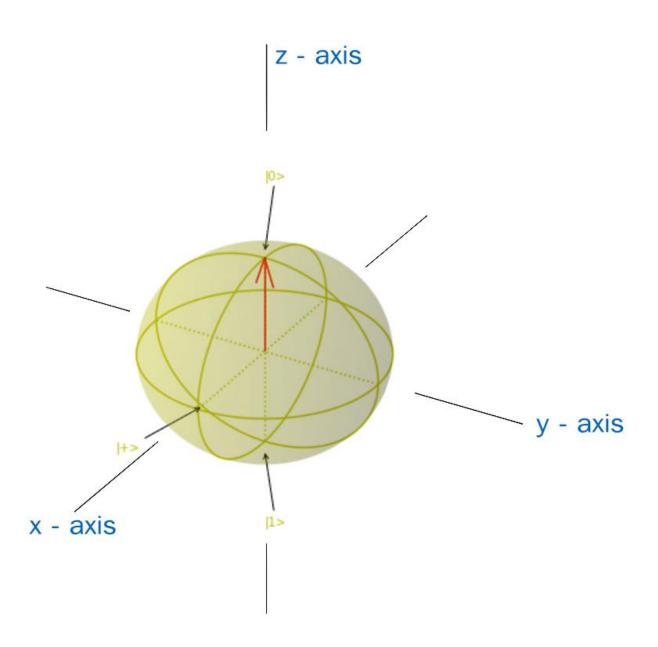
(Note "-" refers to " to" and is not the negative sign as used above).

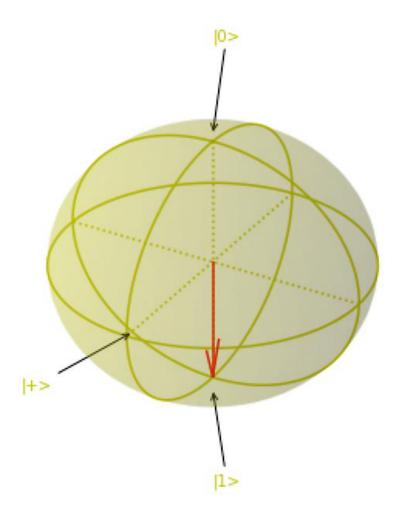


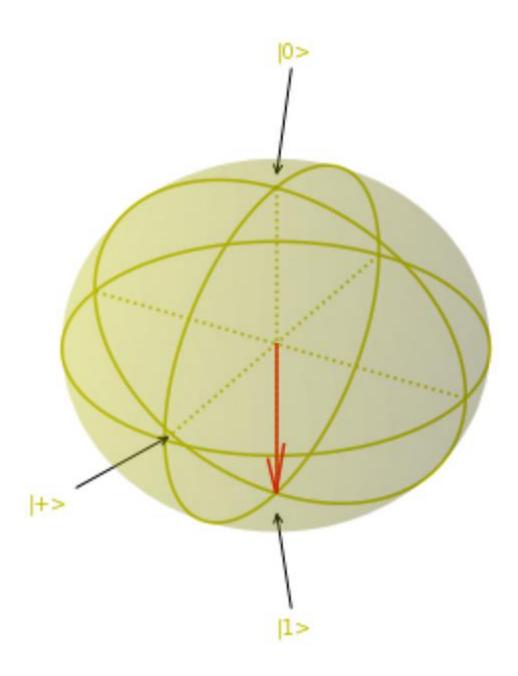


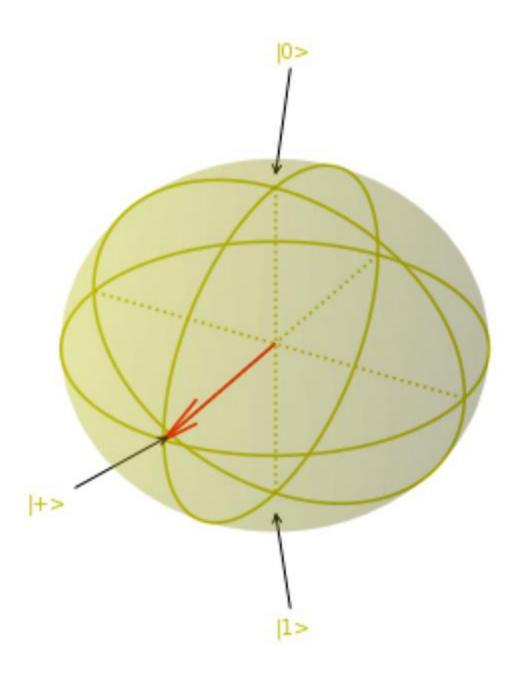
Chapter 6: Using Gate-Based Quantum Computers – Qubits and Quantum Circuits

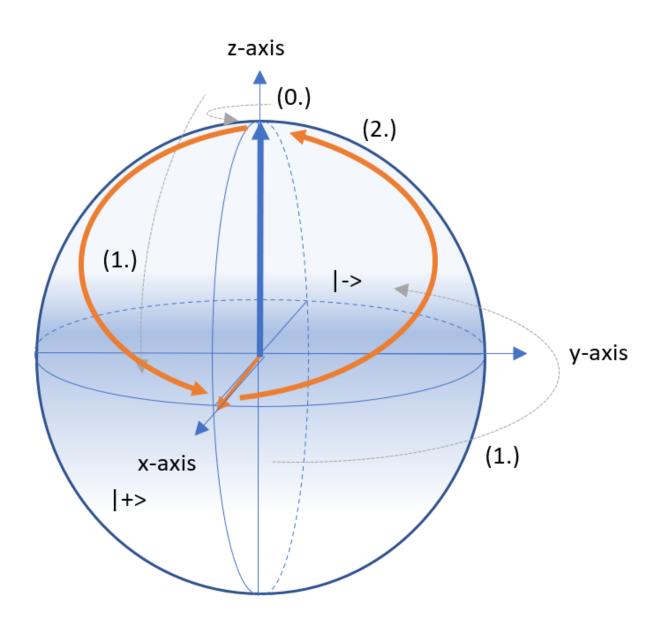


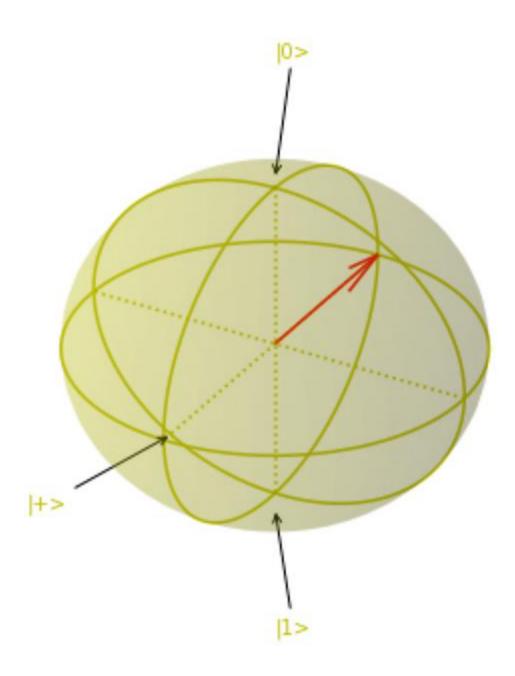


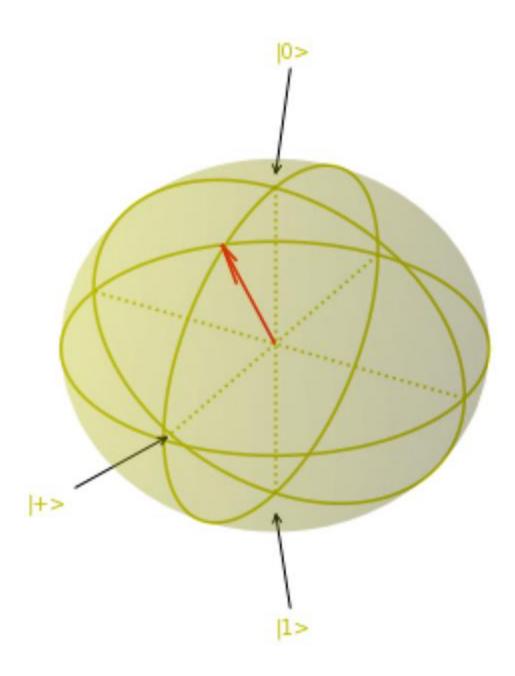


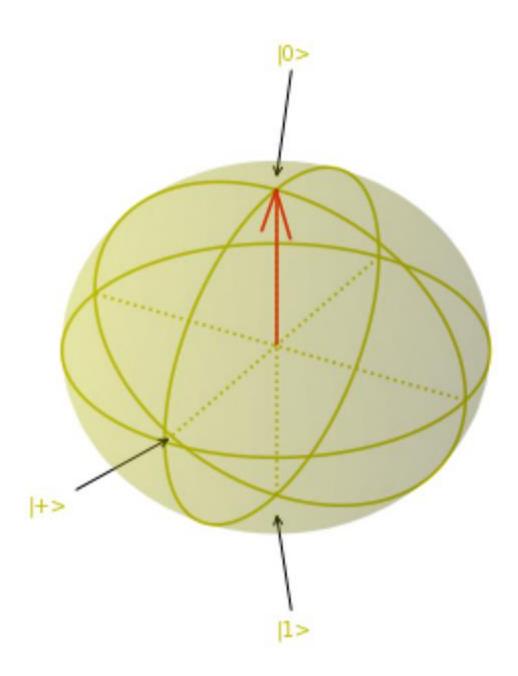


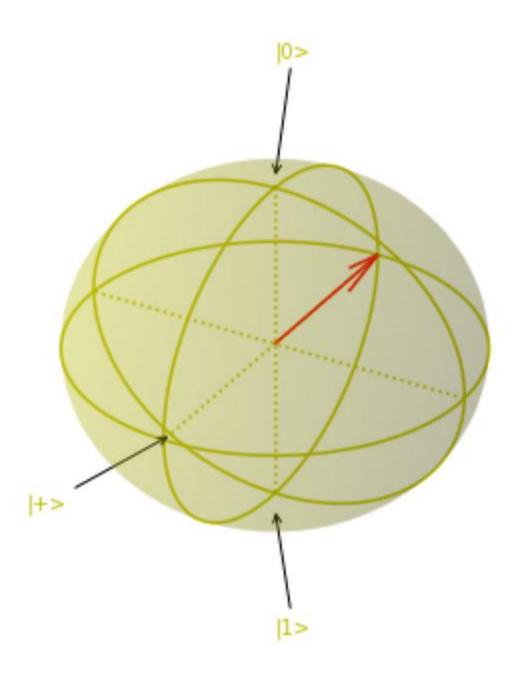


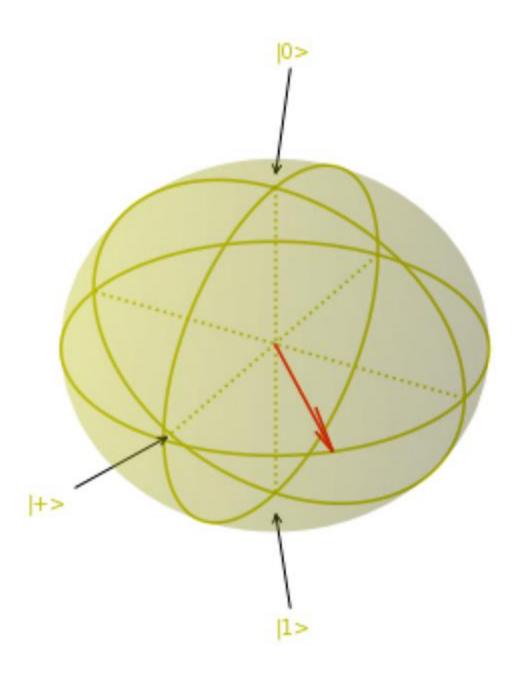


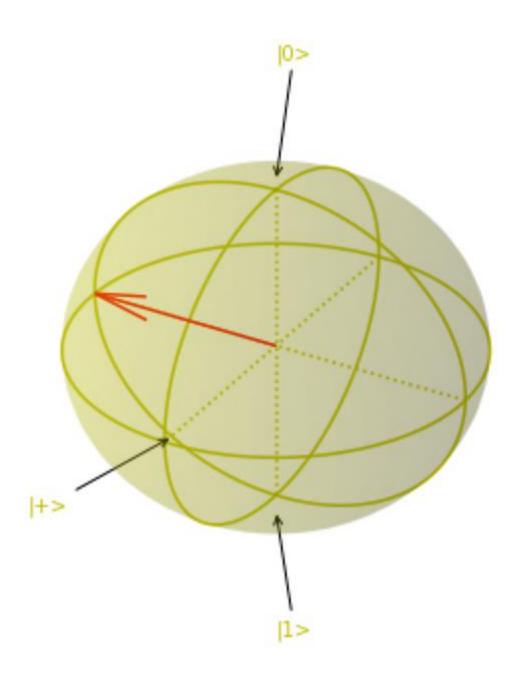


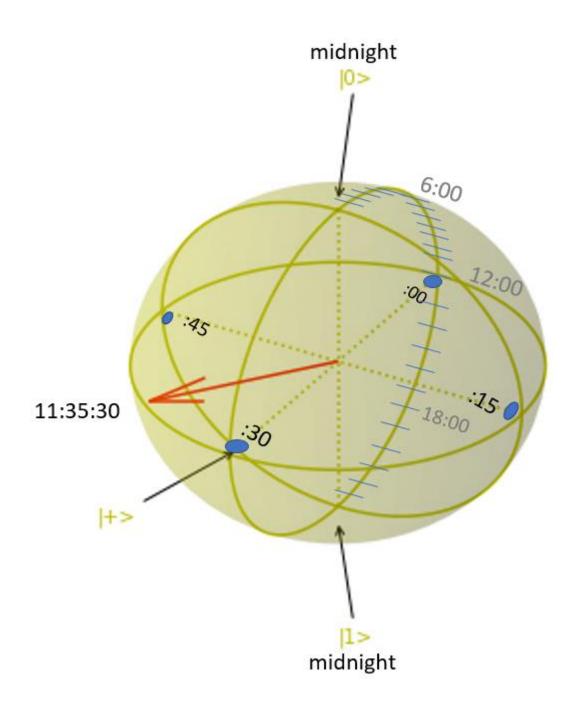


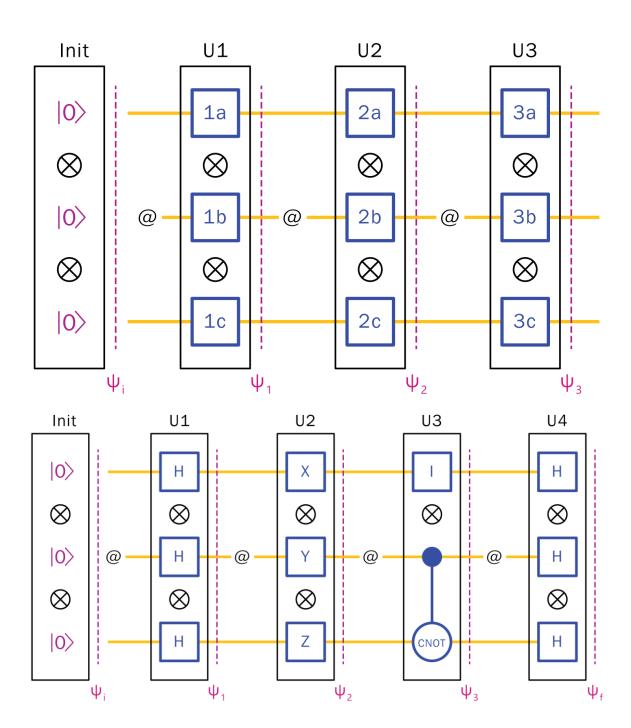


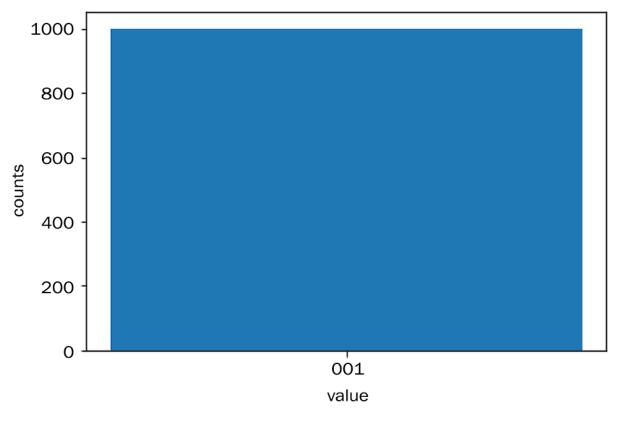


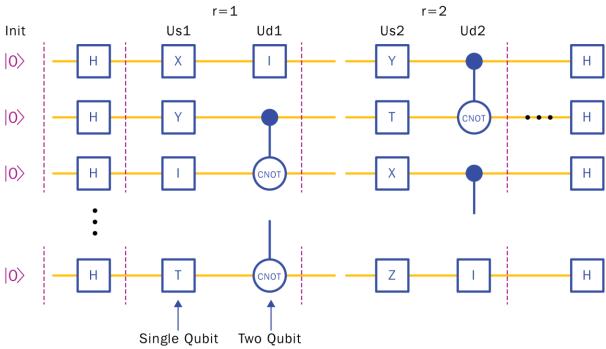


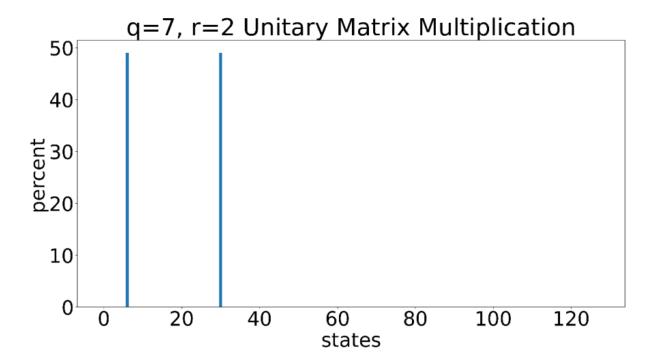


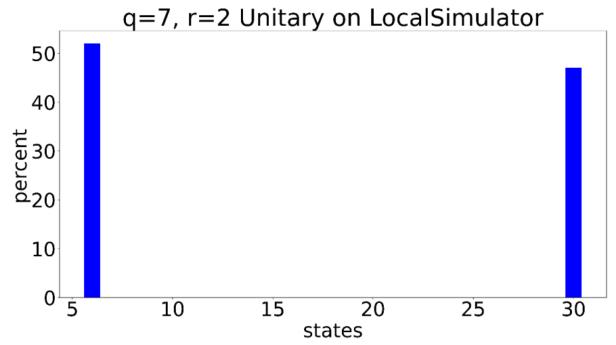


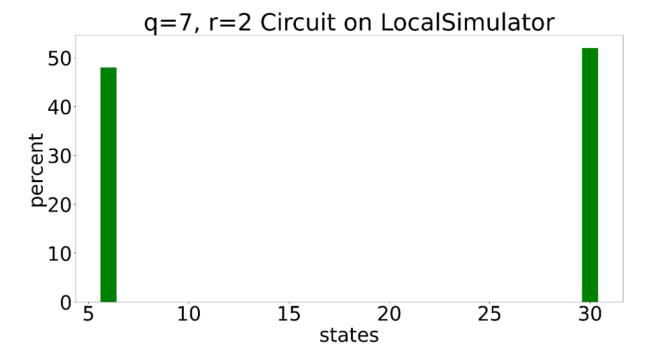


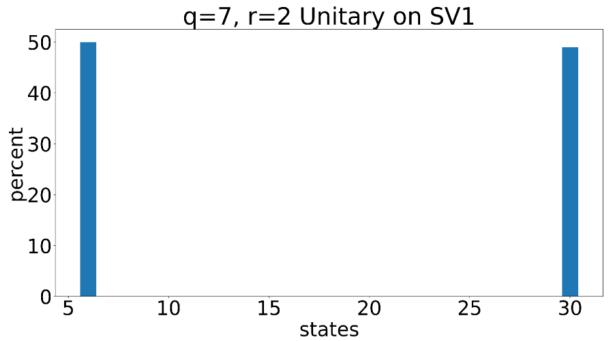


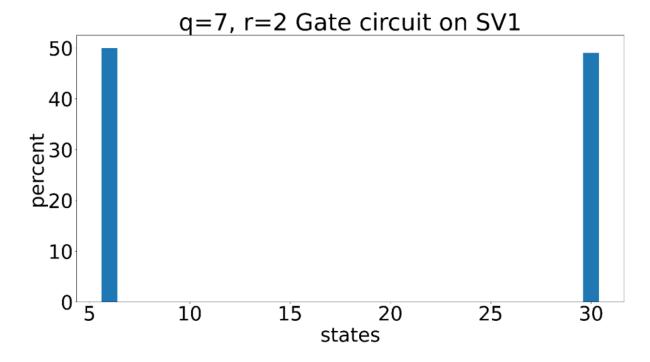


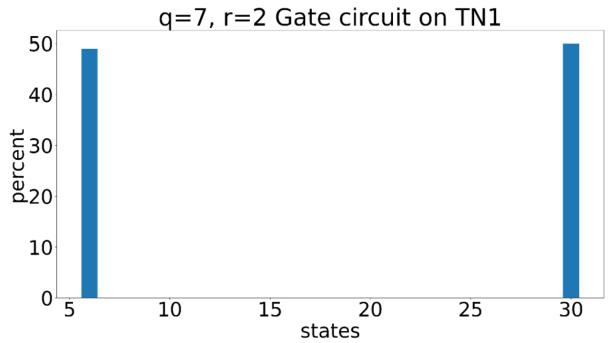


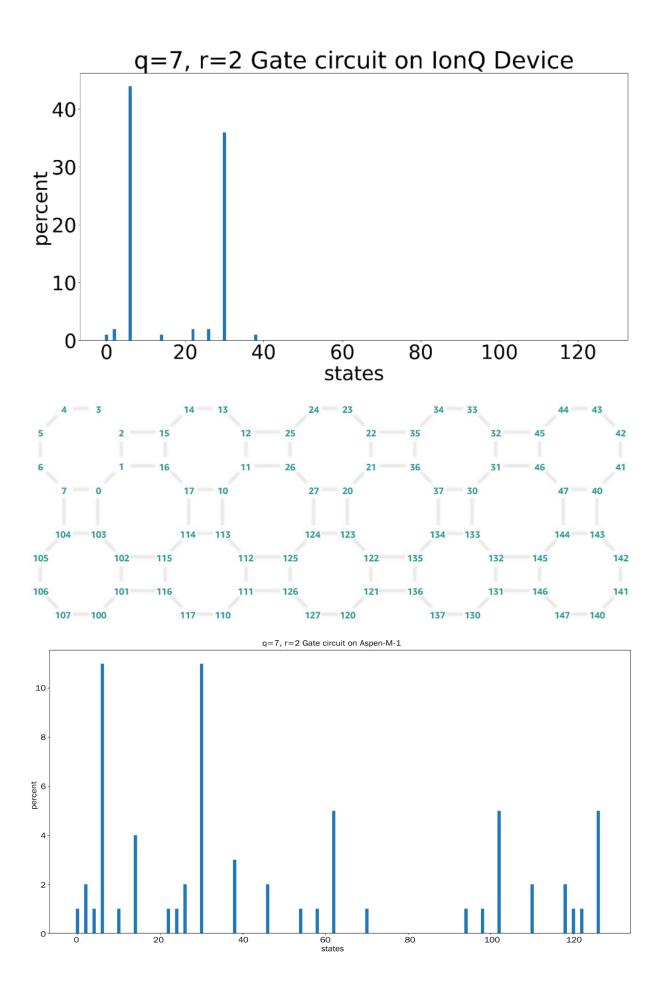




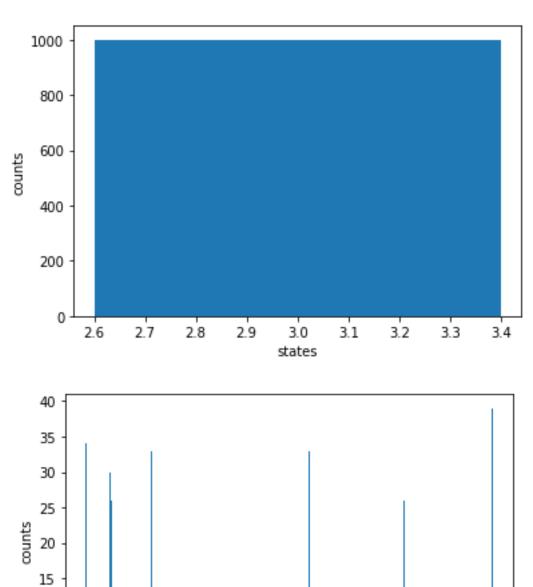




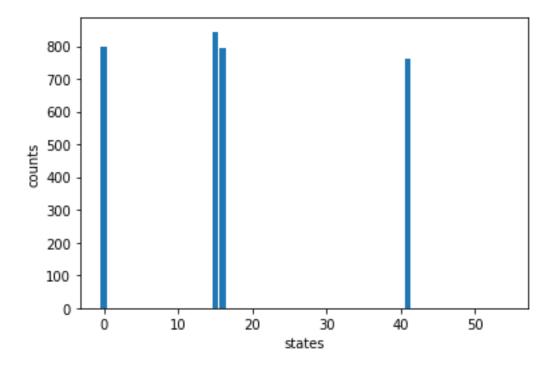


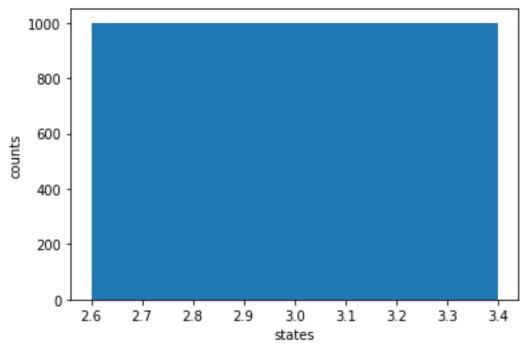


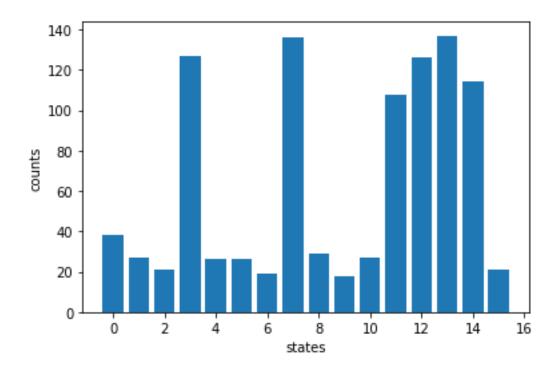
Chapter 7: Using Gate Quantum Computers - Basic Quantum Algorithms

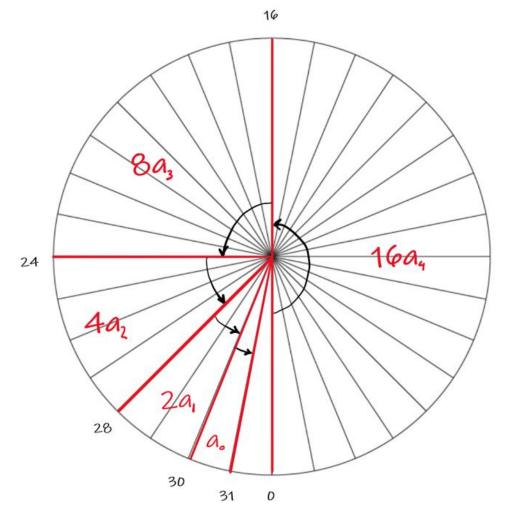


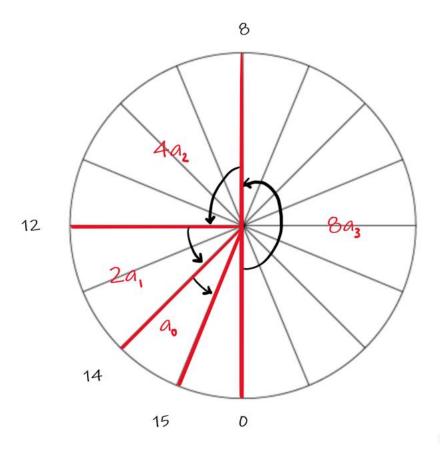
states

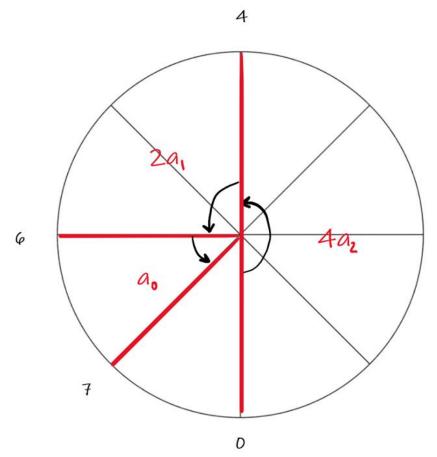


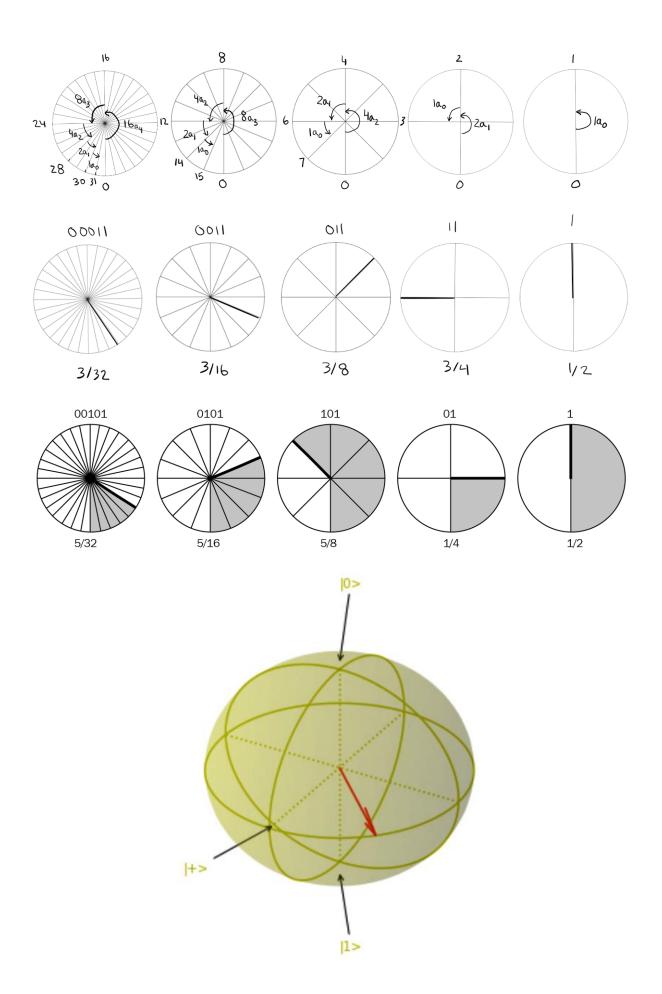


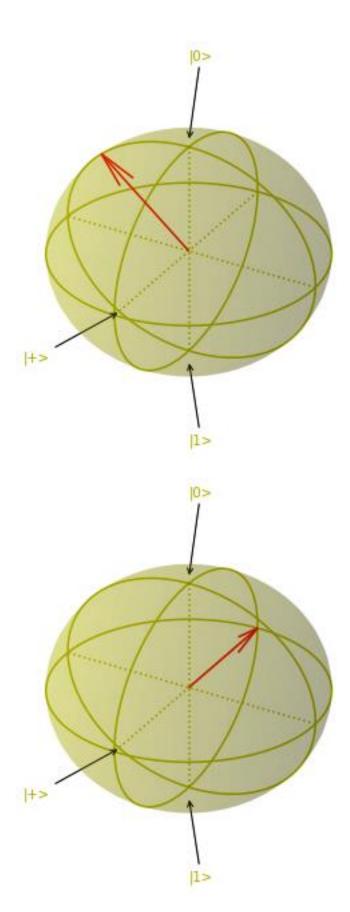


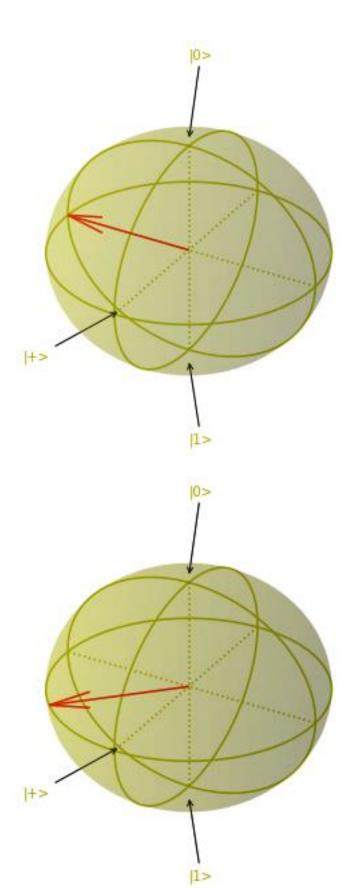


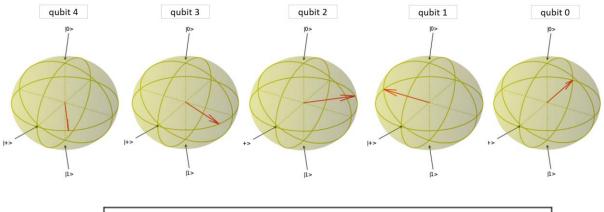


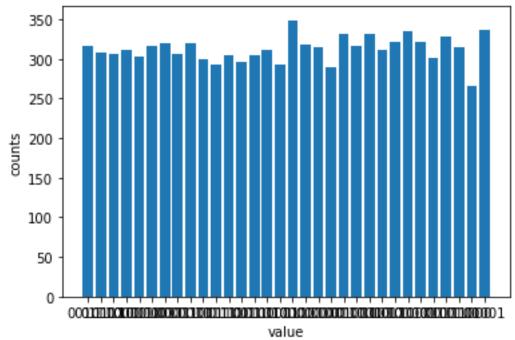


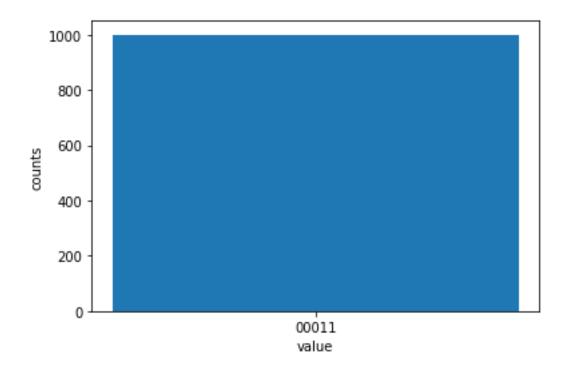




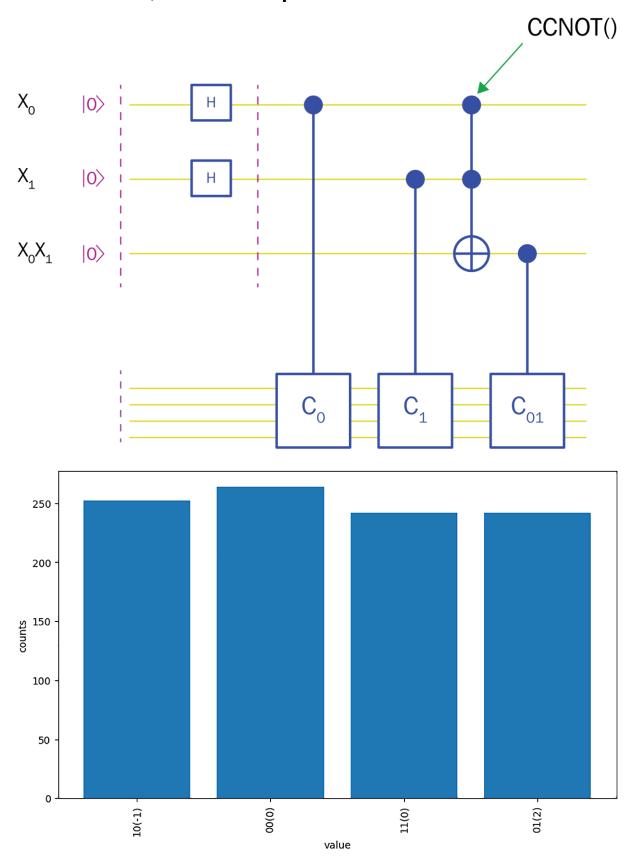


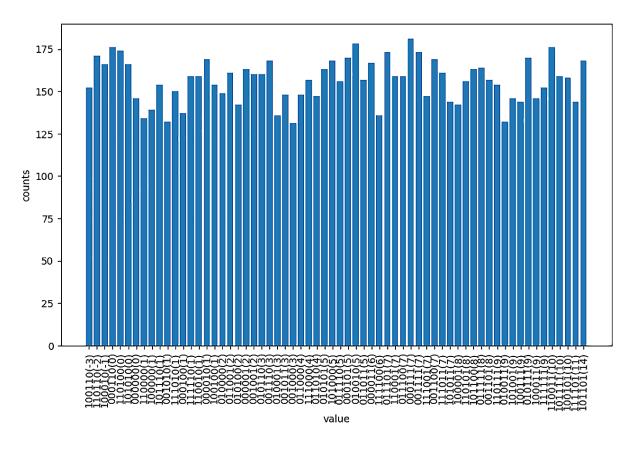




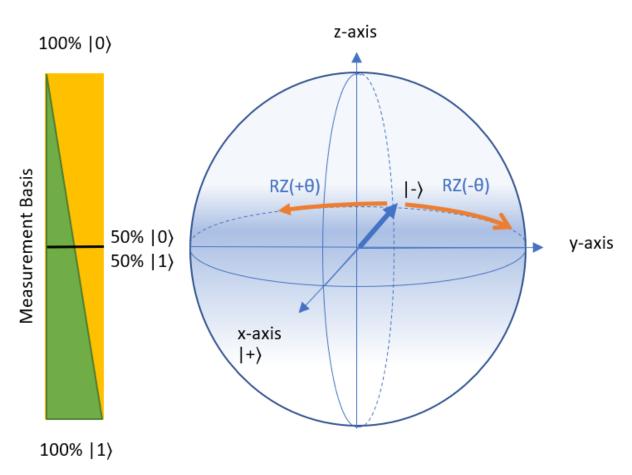


Chapter 8: Using Hybrid Algorithms – Optimization Using Gate-Based Quantum Computers

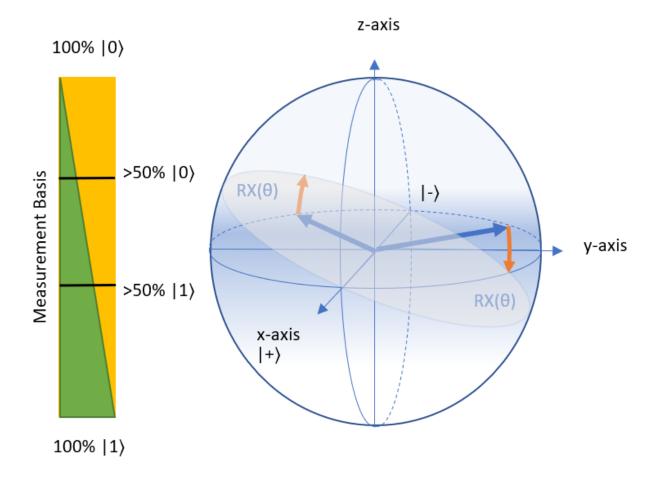




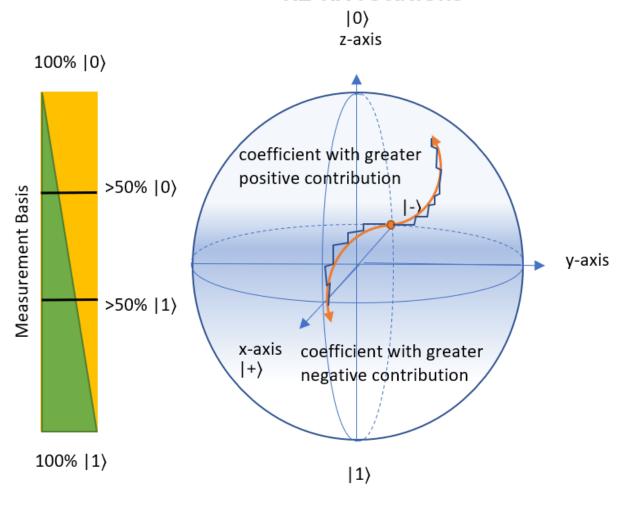
1. RZ rotation

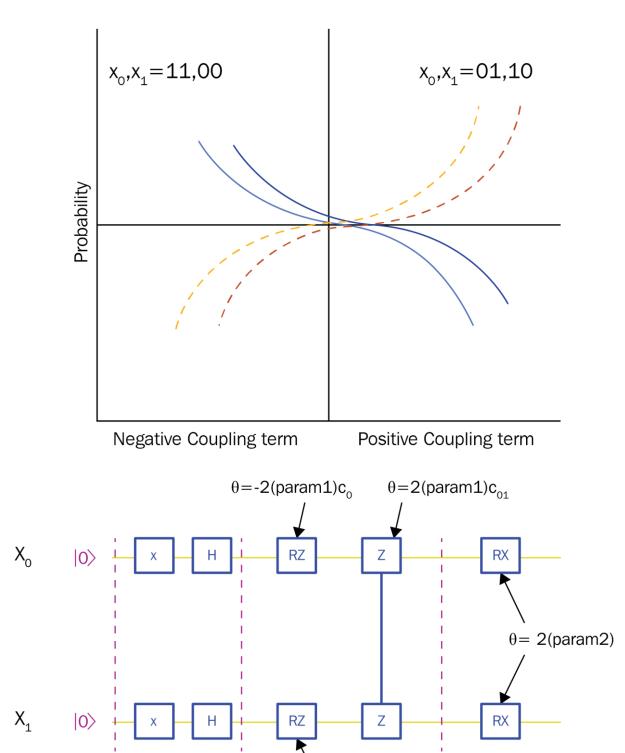


2. RX rotation



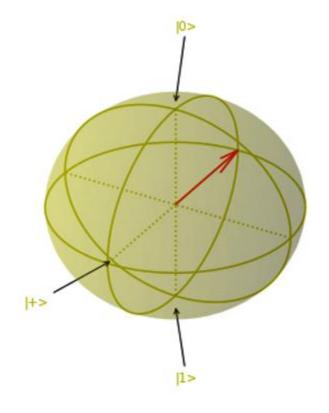
RZ-RX rotations



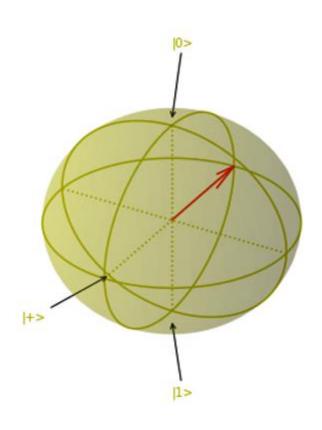


 θ =-2(param1)c₁

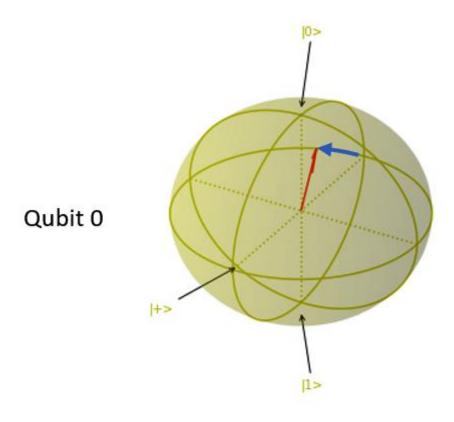
qubit state |->

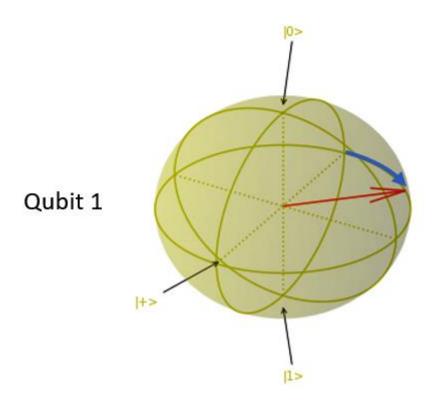


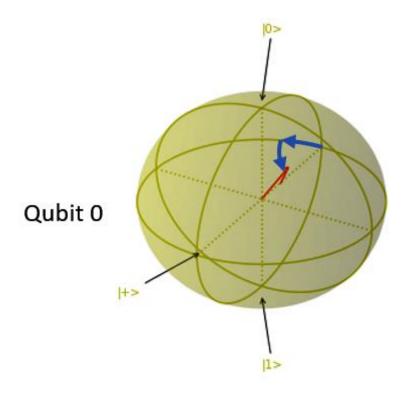
Qubit 0

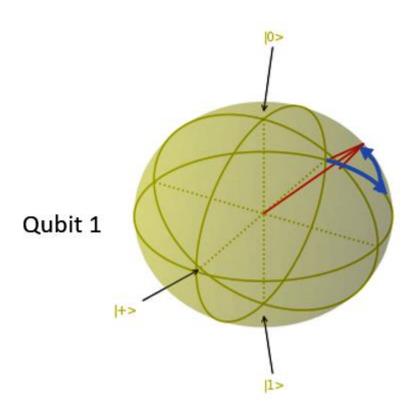


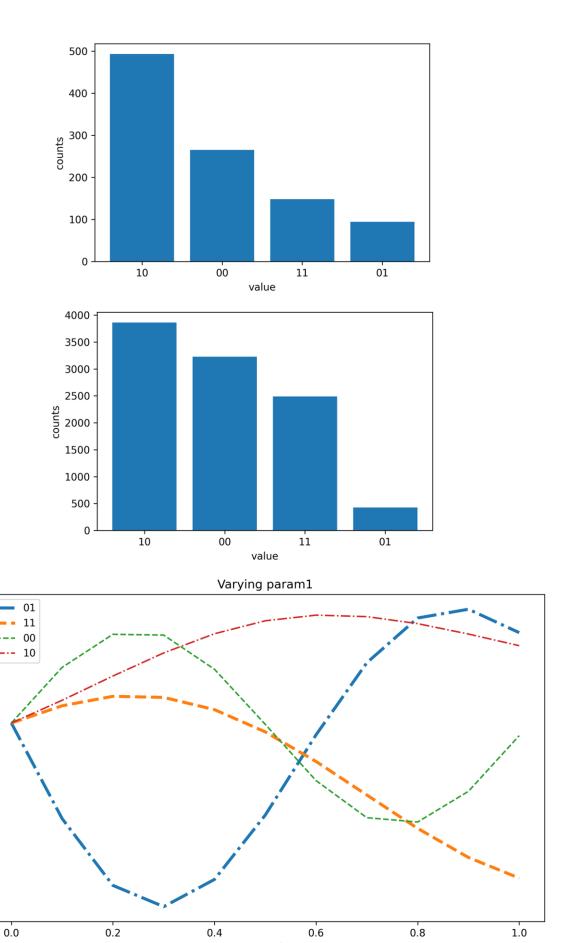
Qubit 1











param1

0.30

0.28

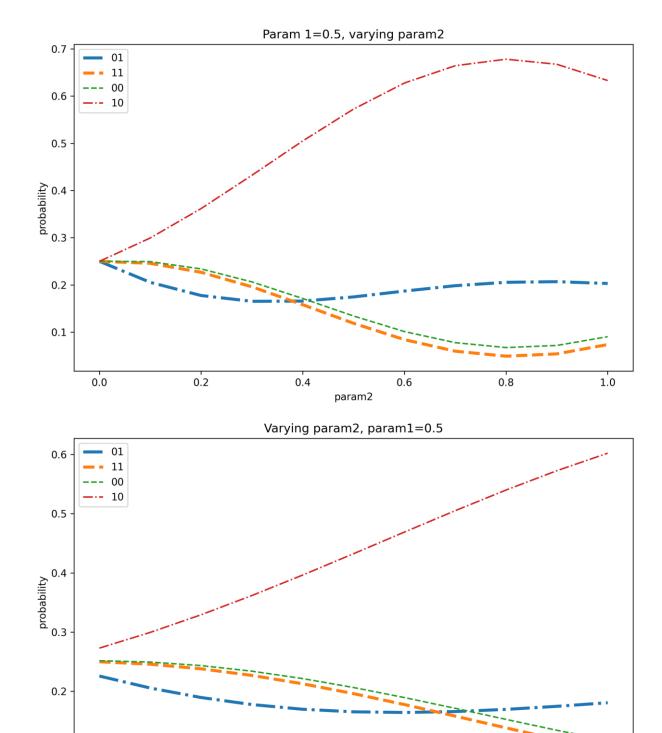
0.26

probability 0.27

0.20

0.18

0.16



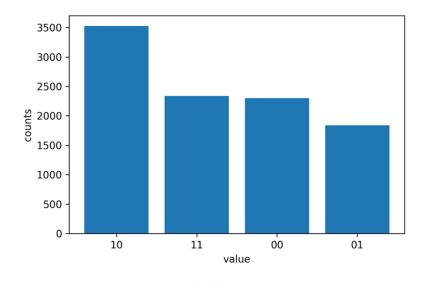
0.1

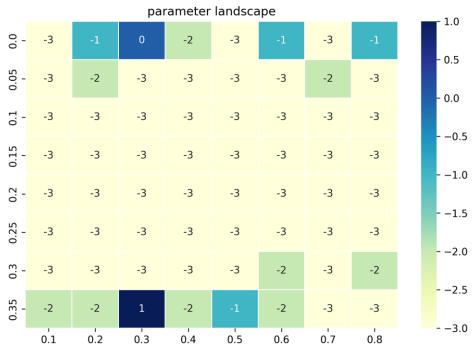
0.1

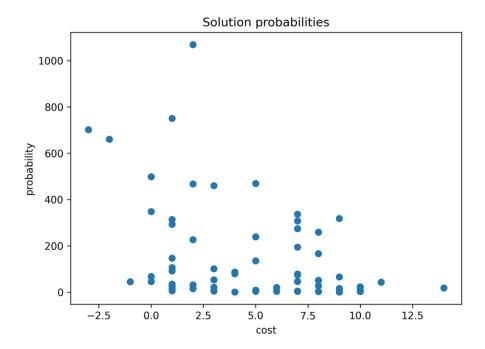
0.2

0.3 param2 0.5

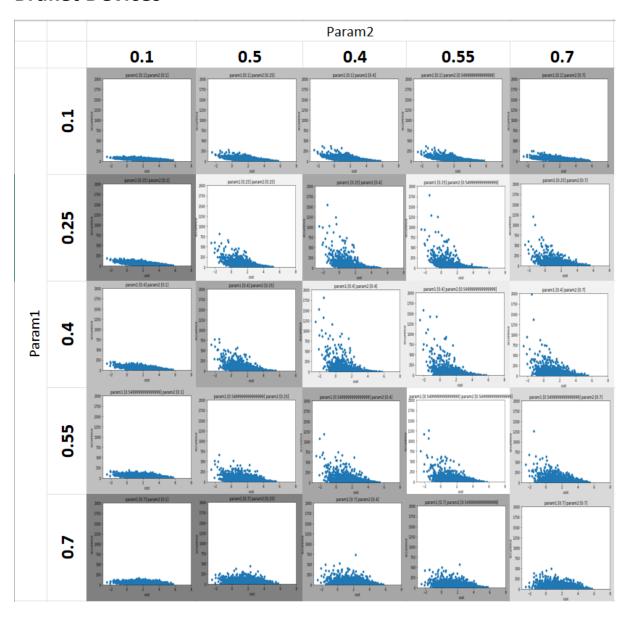
0.4

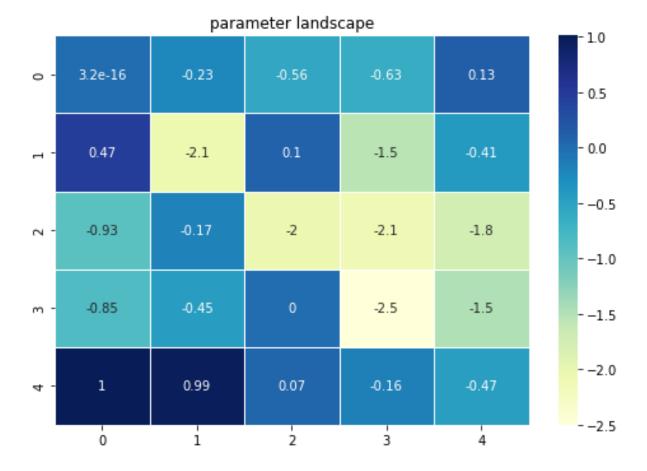


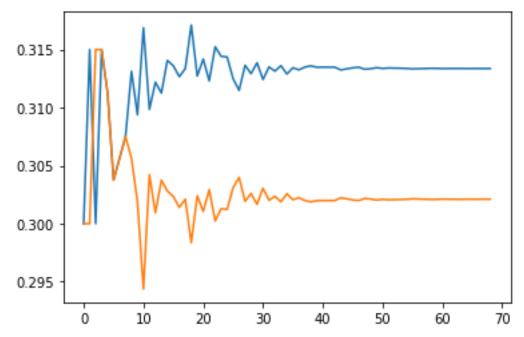


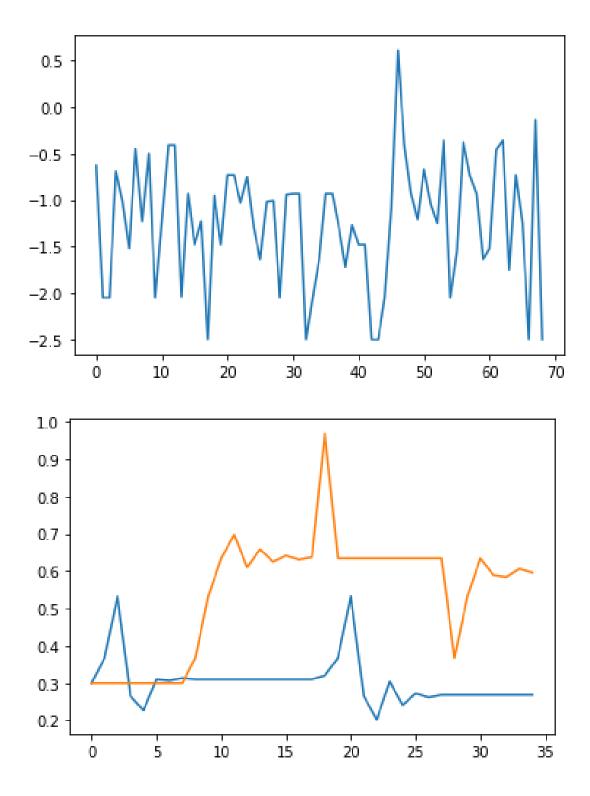


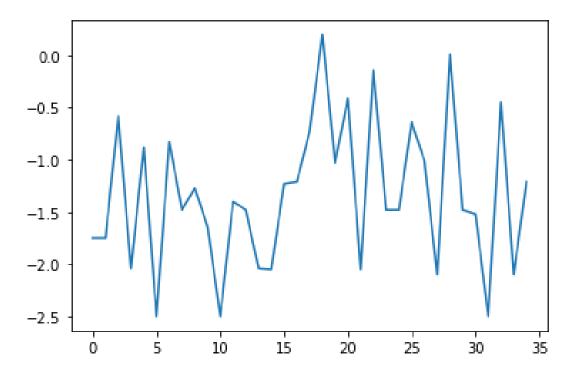
Chapter 9: Running QAOA on Simulators and Amazon Braket Devices

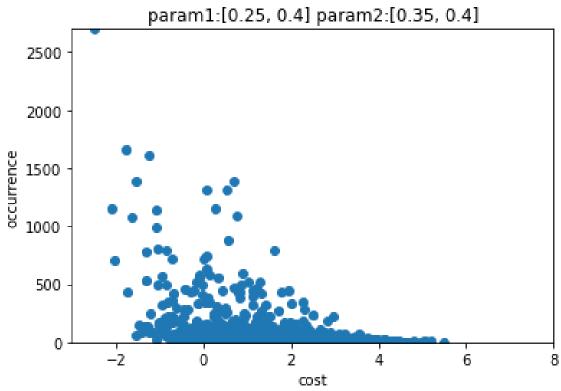


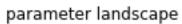


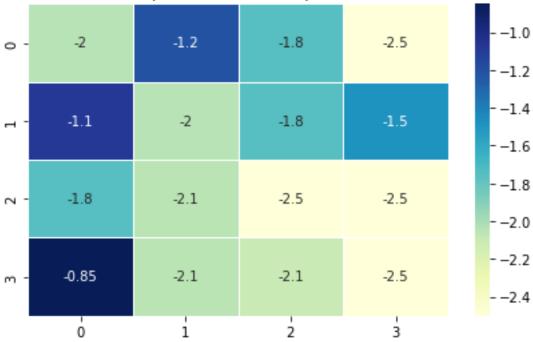




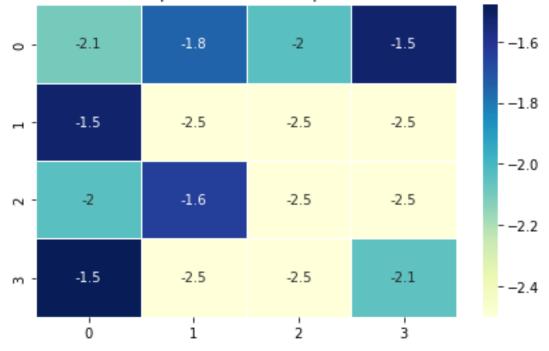


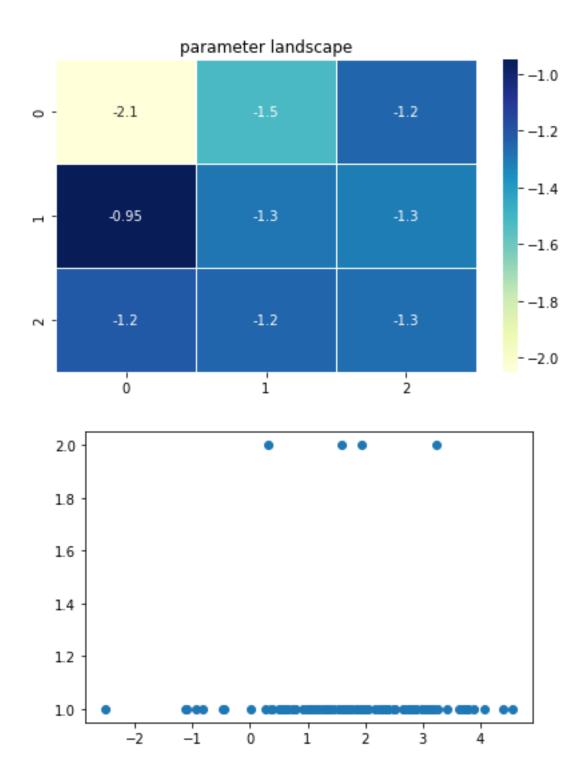


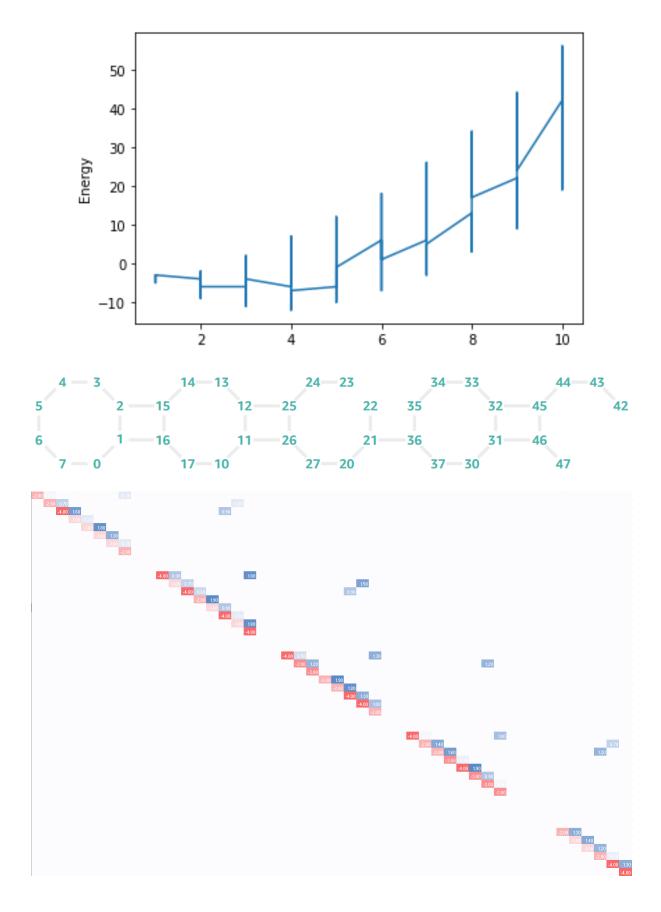


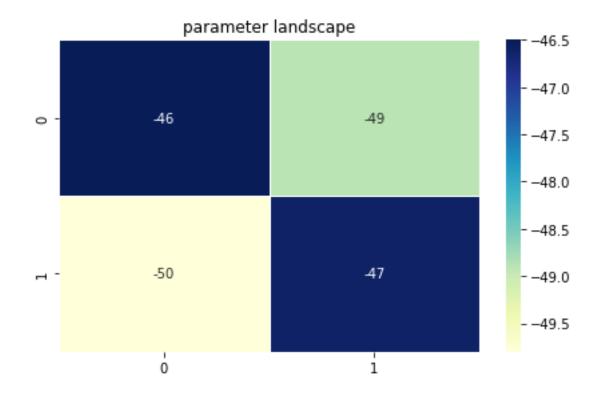


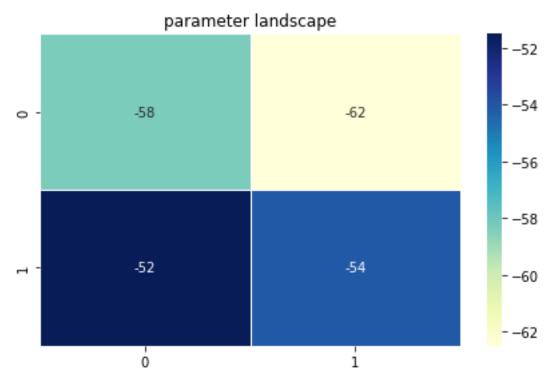
parameter landscape

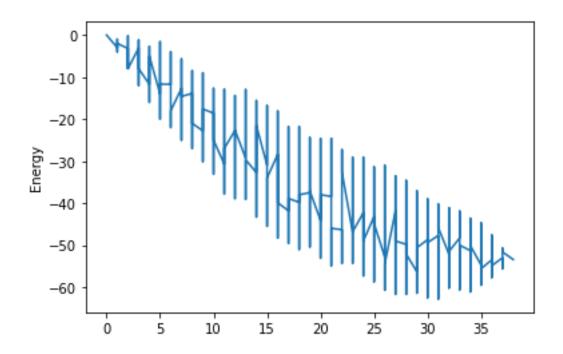


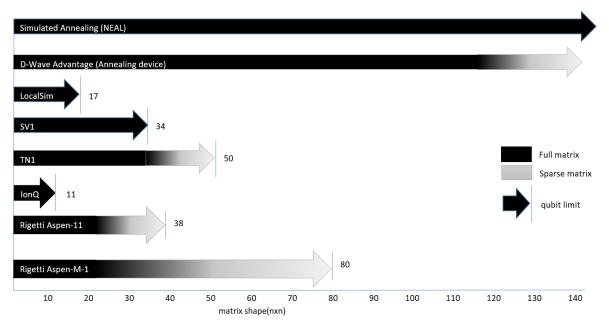




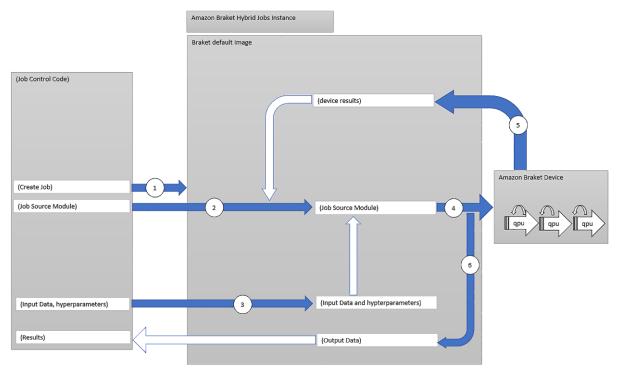


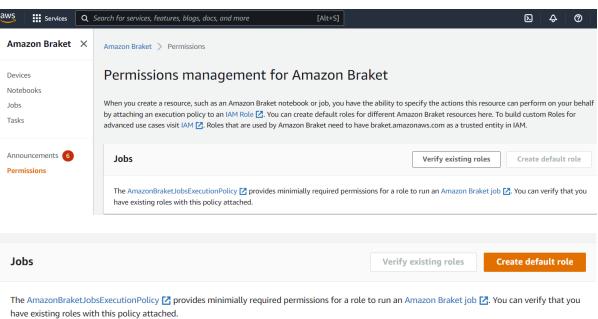


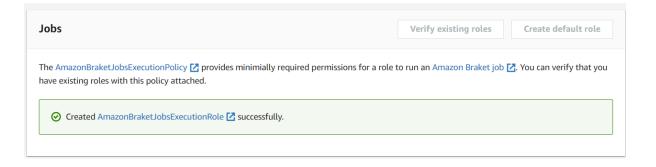




Chapter 10: Amazon Braket Hybrid Jobs, PennyLane, and Other Braket Features







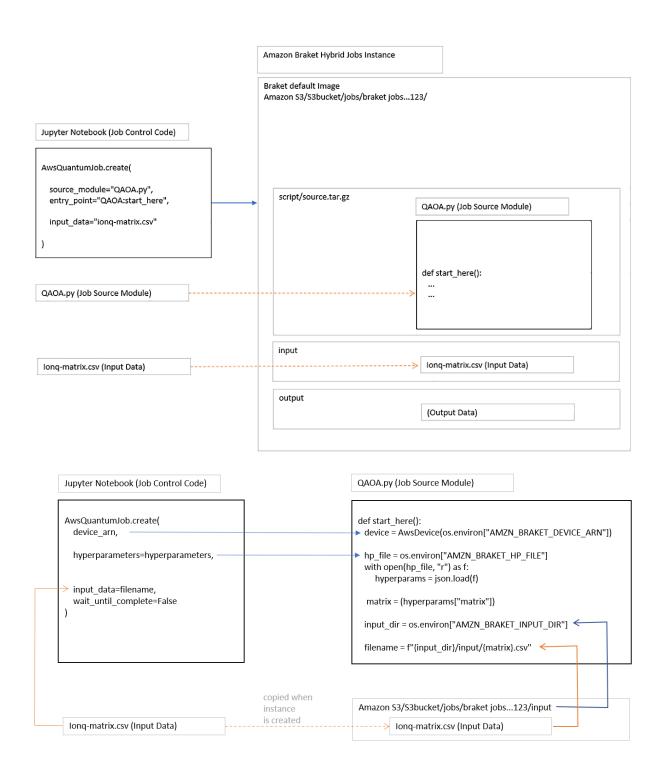
Jupyter Notebook (Job Control Code)

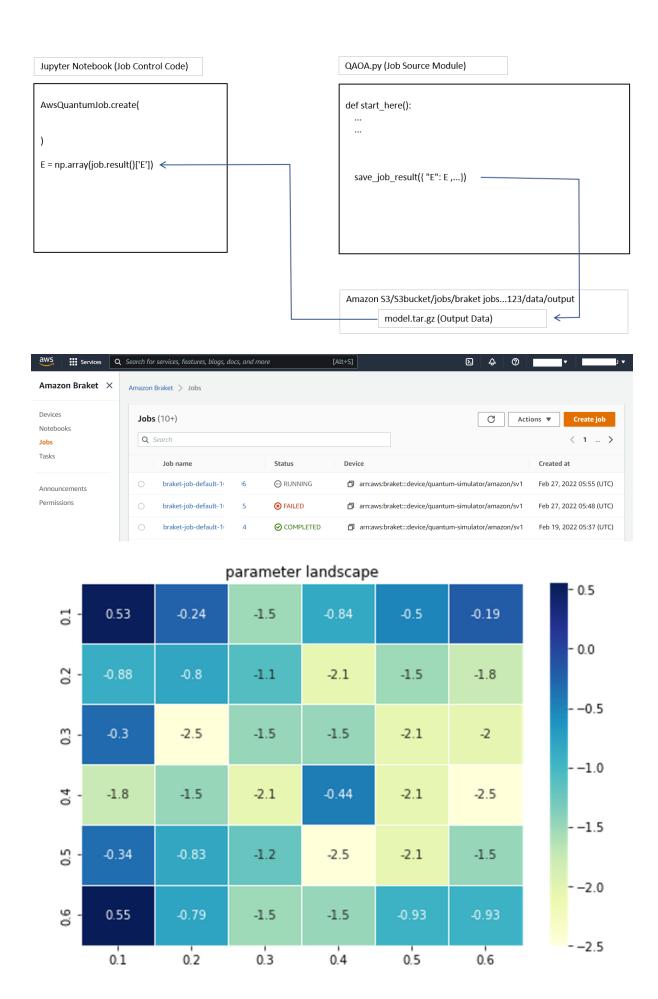
```
AwsQuantumJob.create(

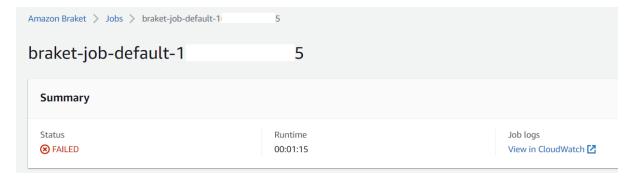
source_module="QAOA.py",
entry_point="QAOA:start_here",
)
```

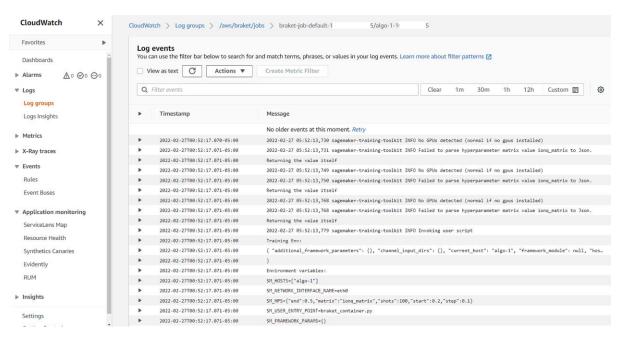
```
QAOA.py (Job Source Module)
```

```
def start_here():
...
...
```

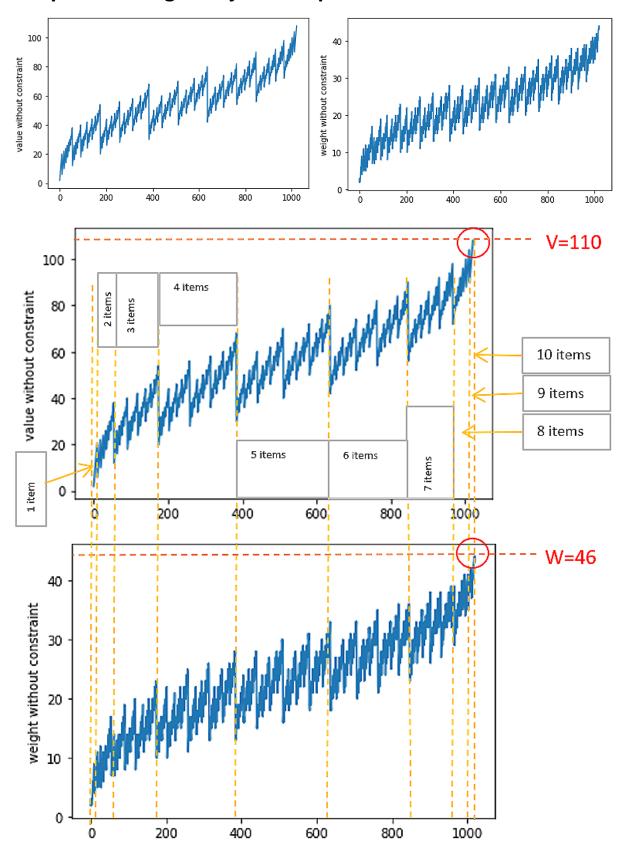


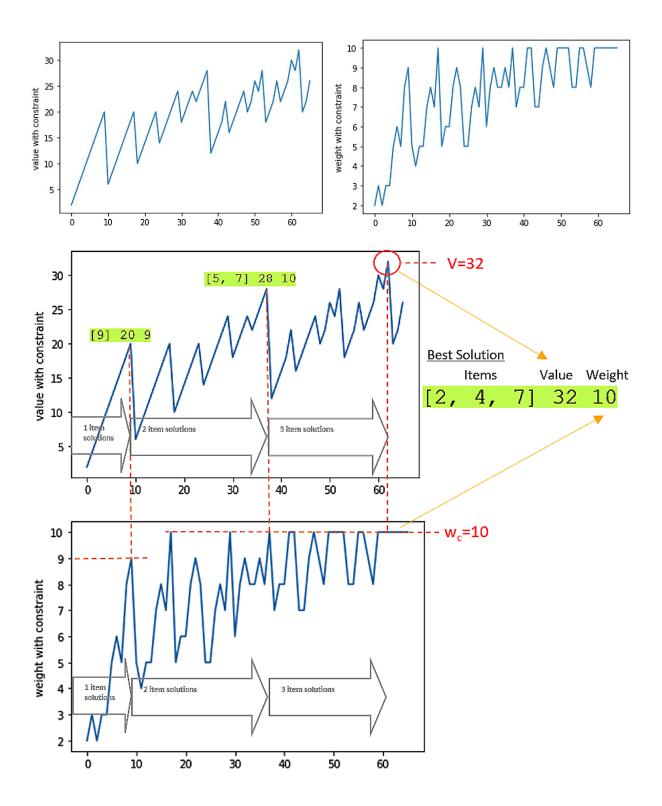


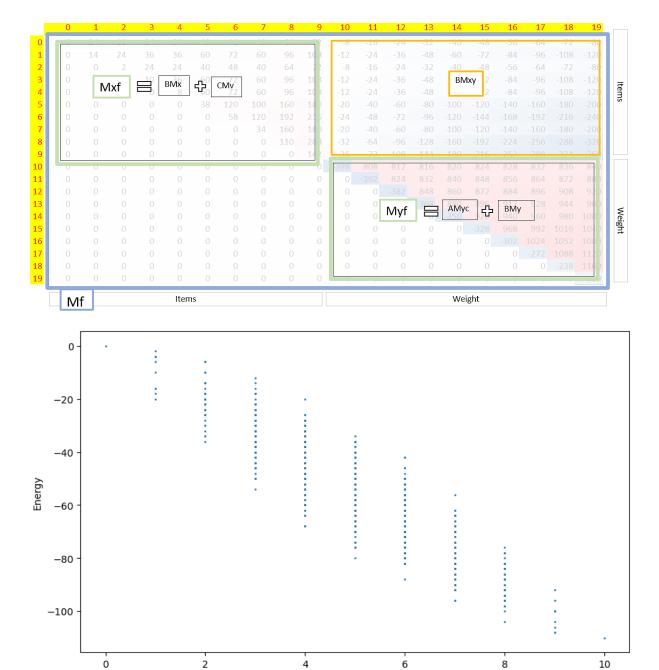


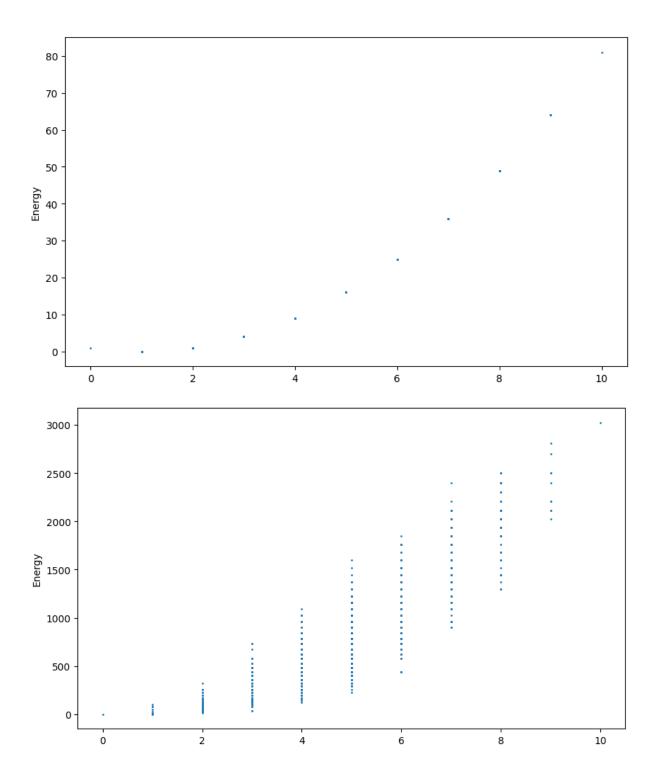


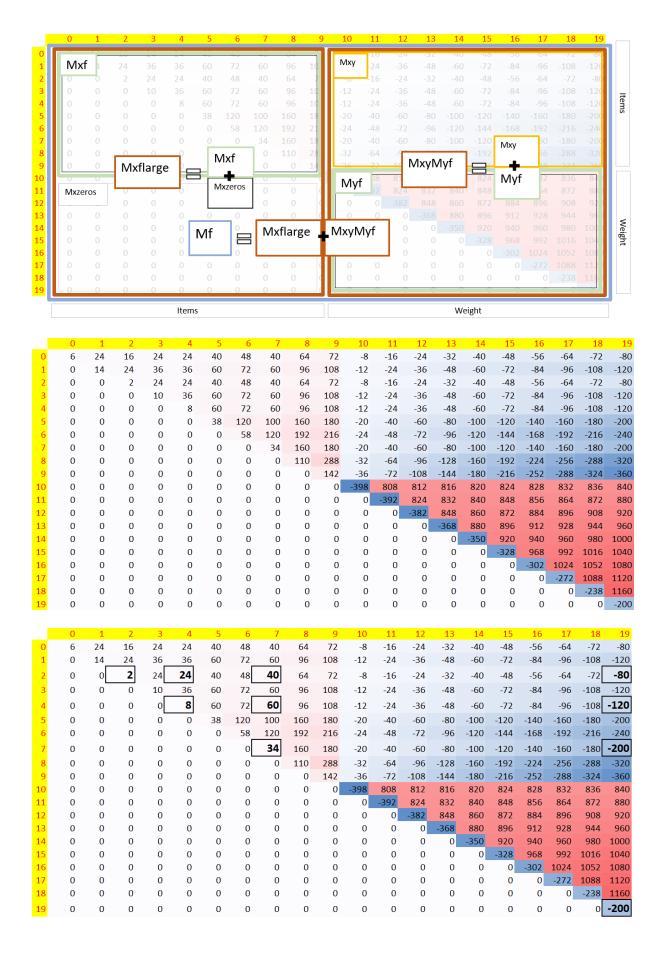
Chapter 11: Single-Objective Optimization Use Cases

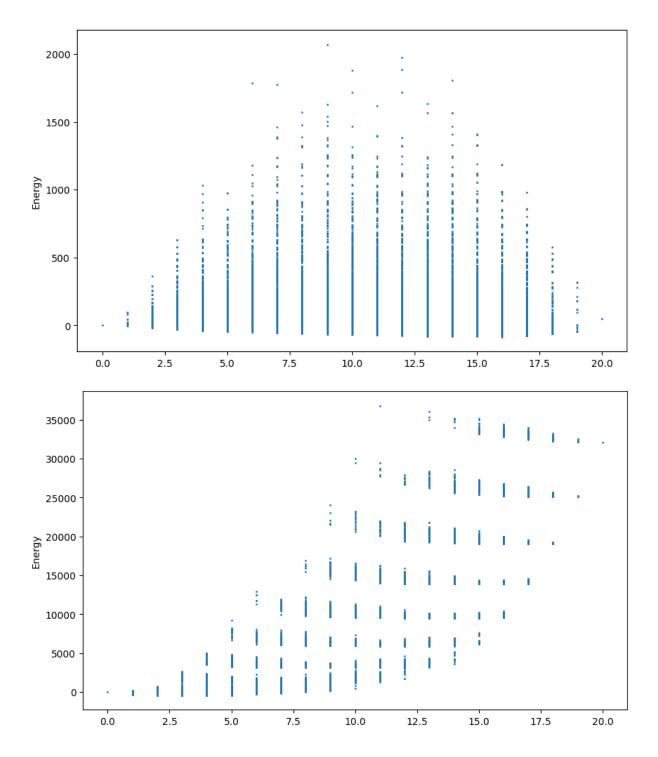


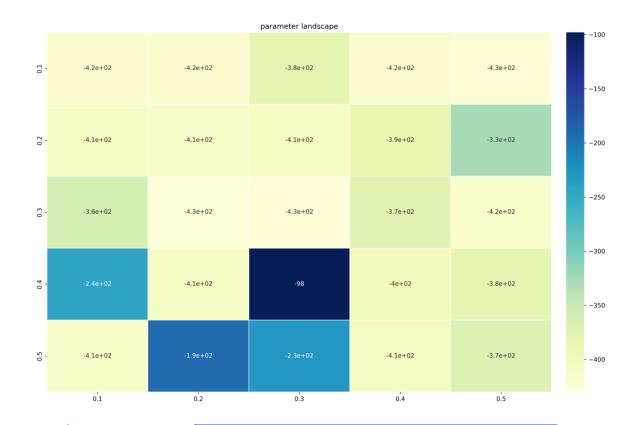












Step 1: Classical

- Develop classical formulation
- Evaluate classical results

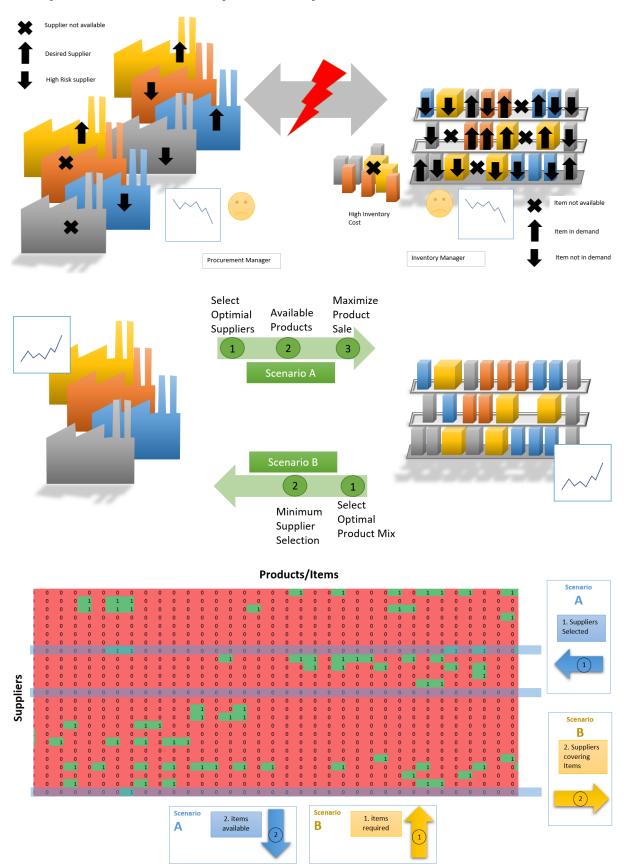
Step 2: QUBO Formulation

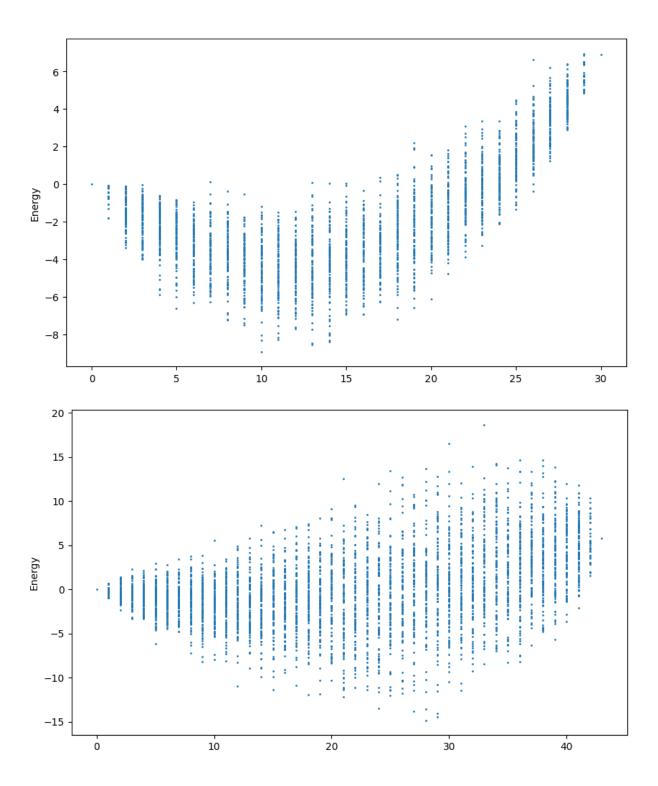
- Develop QUBO matrix based on objective and constraints
- Ensure correct strength of constraints
- Use probabilistic solver() to evaluate the energy landscape
- Extract solution values and filter for feasible solutions
- Ensure best solution has the lowest energy

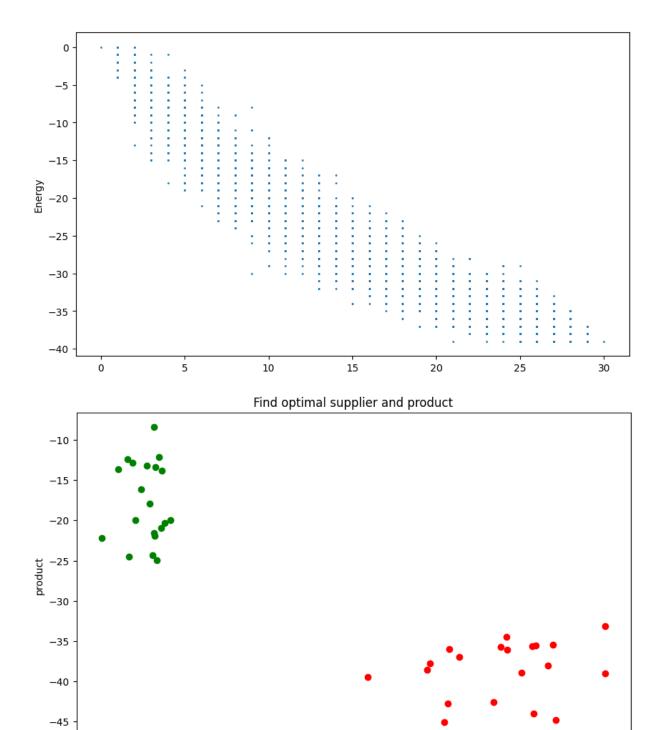
Step 3: QUBO Solvers

- Run Matrix on QUBO solvers
- Convert QUBO matrix to required format if needed:
- BQM for D-Wave
- QAOA Ansatz for gate-based quantum computers
- Adjust device and hardware specific parameters for optimal results
- eg. chain-strength, shots, QAOA parameters, qubit embedding
- Extract solution values and filter for feasible solutions

Chapter 12: Multi-Objective Optimization Use Cases







-2

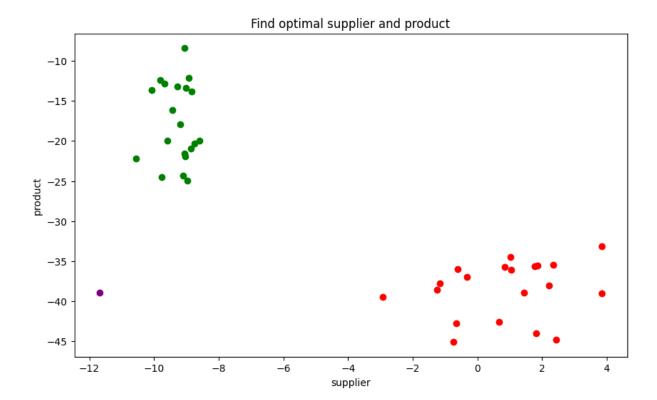
Ó

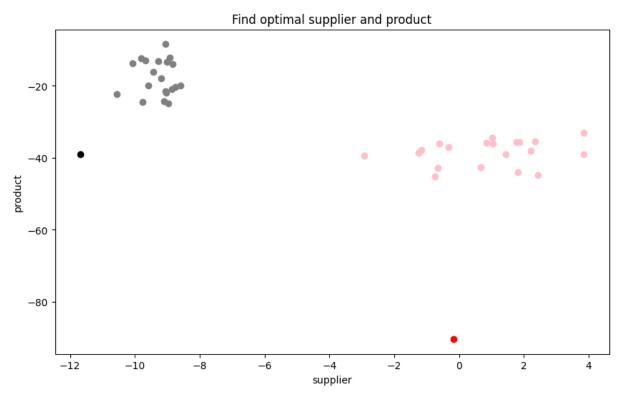
–4 supplier

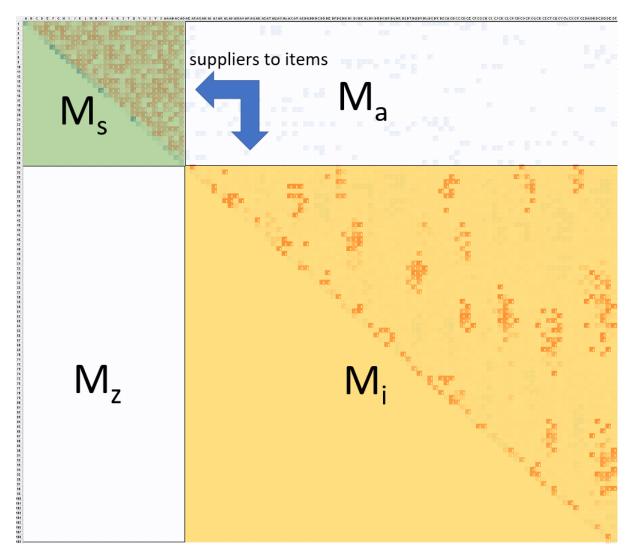
-10

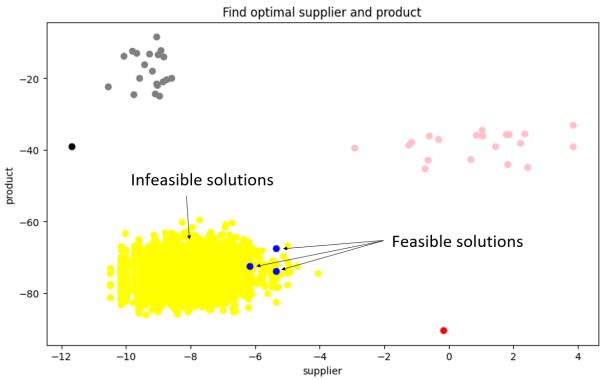
-8

-6

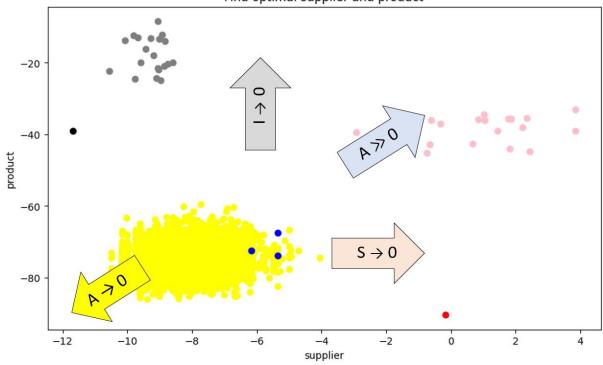




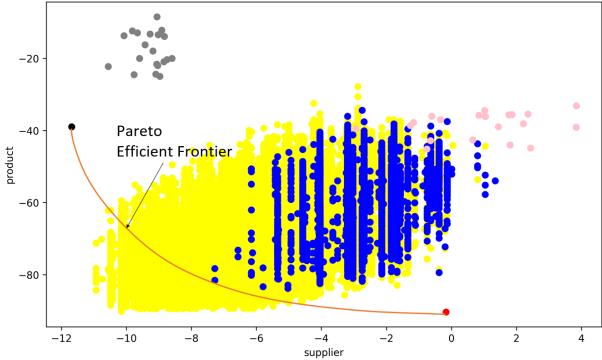


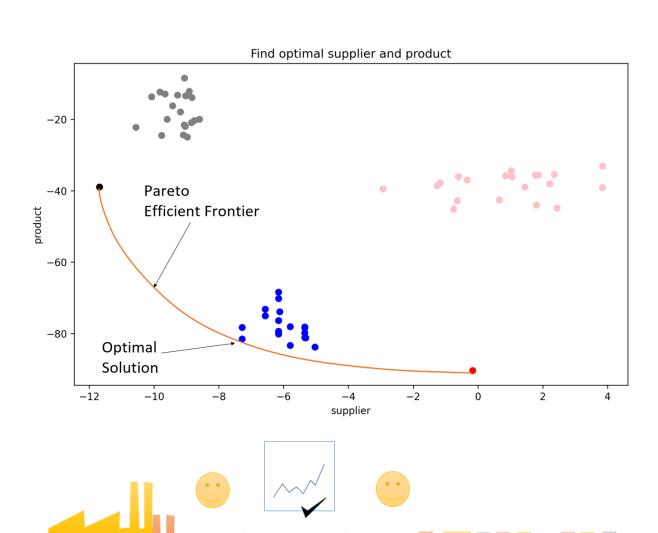












Chapter 13: Knapsack BQM Derivation

No Images...