QAOA를 이용한 TSP 문제 풀이 시도

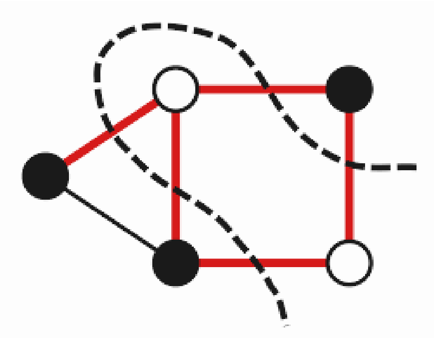
한양대학교 김도하

Index

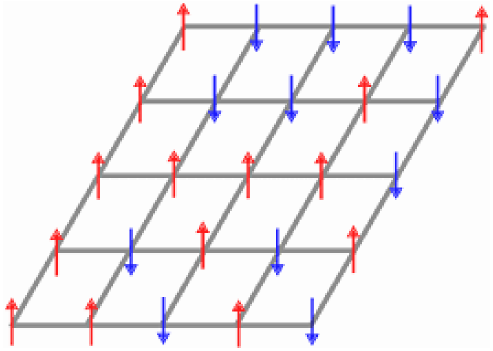
- 연구 동기
- 풀려는 문제: TSP는 무엇인가
- TSP를 Ising model에 대응시키기
- 실험 결과
- 결과 분석 및 개선사항
- 참고 논문

연구 동기

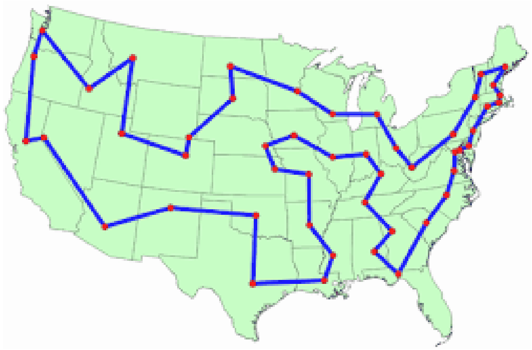
- TSP는 NP-complete 문제. 양자컴퓨터로 쉽게 풀 수 있다면 Quantum Superemacy 증명에 기여할 수 있을 것.
- 이론상 QAOA로 모든 그래프 문제를 풀 수 있지만, 대부분의 QAOA는 MAX-CUT 문제를 푸는 것임: MAX-CUT 문제를 Ising Model에 대응시킬 수 있기 때문
- TSP를 QAOA로 풀려는 시도에서는 대부분 3개의 도시만을 가진 문제에 대해서만 풀던데, 4개 이상의 도시에 대해서도 풀어보고 싶었음



MAX-CUT



Ising Model



TSP

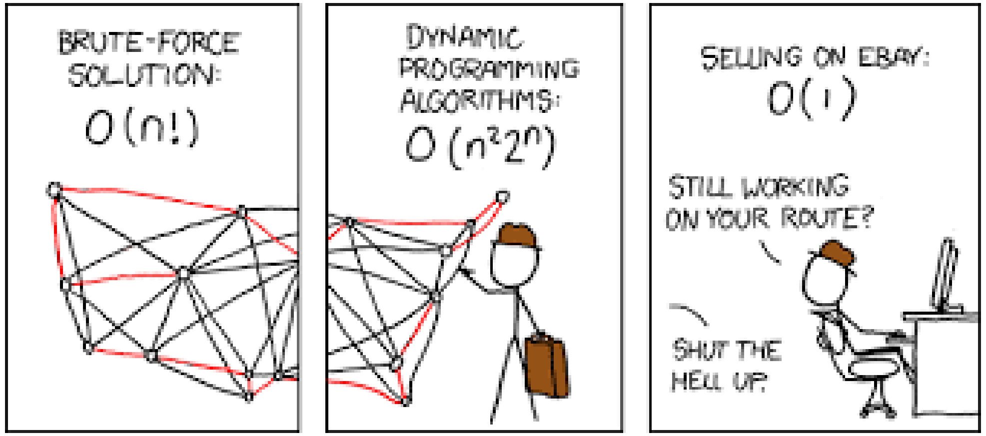
풀려는 문제: TSP는 무엇인가?

주어진 그래프 $G = (V, E)$ 가…
- 연결되어 있고(Connected)
- 가중치가 있는(Weighted)
- 완전한 (Complete)

그래프라고 하자.

이 그래프의 출발 정점에서 다른 모든 정점을 방문하고 , 원래의 출발 정점으로 다시 돌아오는 순환 경로 중, 가중치의 합이 **최소**가 되는 경로를 찾아라.

풀려는 문제: TSP는 무엇인가?



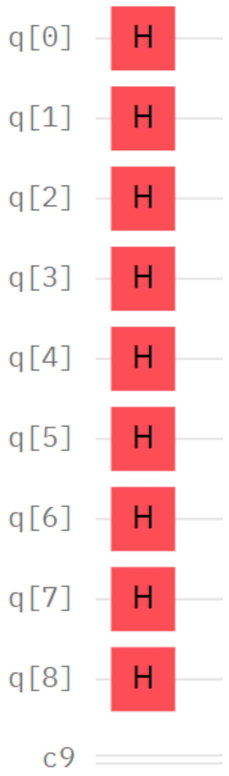
Brute Force: $O(n!)$
Dynamic Programming: $O(n^2 2^n)$

의 시간 복잡도로 풀 수 있음이 알려져 있음

TSP를 Ising Model(=QAOA)에 대응시키기

- QAOA에는 2개의 Hamiltonian을 설정함: Cost Hamiltonian, Mixer Hamiltonian
- 즉, QAOA로 문제를 풀고 싶다면 아래의 과정을 거치면 된다:
  - 최적화 하려는 문제를 인코딩 할 Cost Hamiltonian H_C를 정의
  - Mixer Hamiltonian H_M을 정의
  - Cost Layer과 Mixer Layer를 구성하는 회로 $e^{-i\gamma H_C}, e^{-i\alpha H_M}$을 구성
  - $n \geq 1$의 n을 설정하고, cost layer과 mixer layer가 반복되는 $U(\gamma, \alpha) = e^{-i\alpha_n H_M} e^{-i\gamma_n H_C} \ldots e^{-i\alpha_1 H_M} e^{-i\gamma_1 H\_C}$ 회로를 구성
  - Initial state를 준비하고, $U(\gamma, \alpha)$를 적용 한 후, classical한 방법을 통해 optimize를 진행한다.
  - 측정을 통해 문제 해결

Preparing Initial State



Initial State를 준비하는 것은 간단하다:

$$\sum_x \psi_x |x\rangle$$

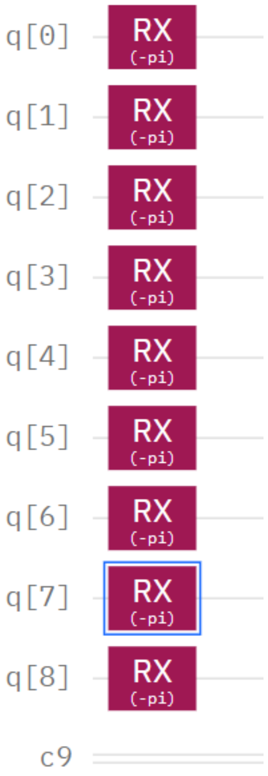
를 준비하면 된다.

이번에는 각각의 matrix index를 embedding 하였다.

$$\begin{matrix} 0 & 3 & 4 \\ 3 & 0 & 5 \\ 4 & 5 & 0 \end{matrix} \quad \Longleftrightarrow \quad \begin{matrix} q_0 & q_1 & q_2 \\ q_3 & q_4 & q_5 \\ q_6 & q_7 & q_8 \end{matrix}$$

Mixer Hamiltonian

Mixer Hamiltonian을 정의하는 것은 어렵지 않다



Mixer에 사용하는 Gate로 일반적으로 RX Gate나 RY Gate를 사용하는데, 이번에는 RX Gate를 채택했다.

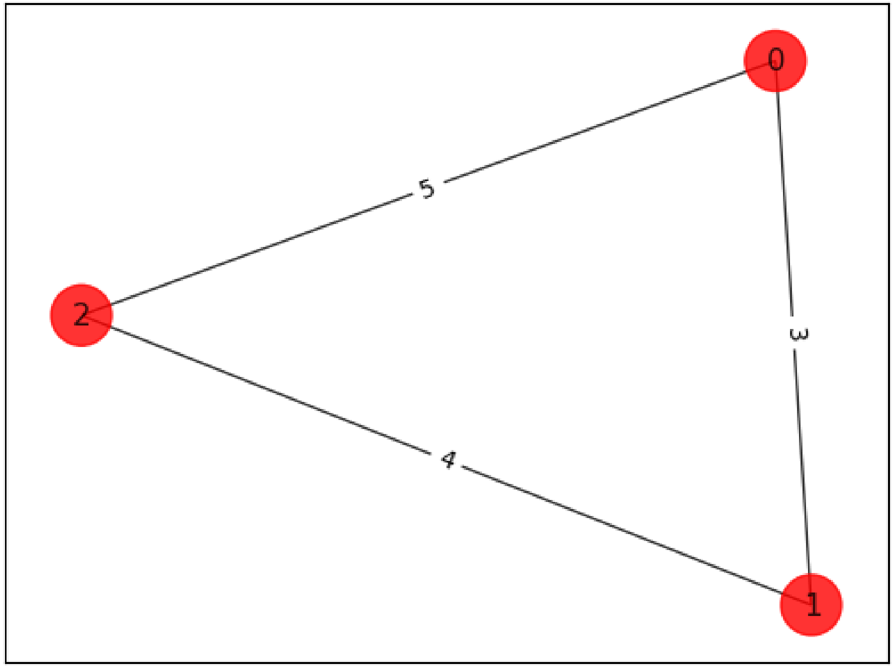
Mixer Hamiltonian을 정의하지 않으면 objective functio이 constan가 되기 때문에, optimize를 진행할 수 없다.

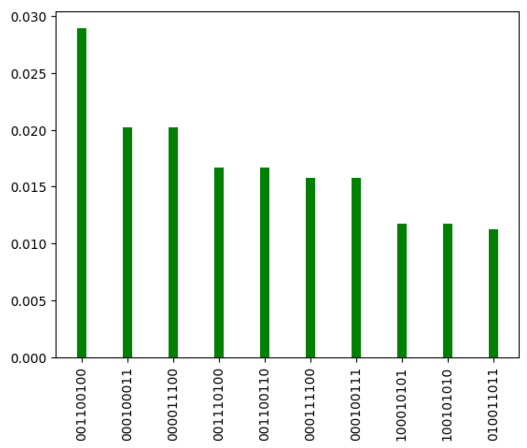
Cost Hamiltonian

- 가장 고민을 많이 하고, 어려웠던 부분
- Distance Matrix = $\begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}$에 대해, $e^{-iw_{ij}Z_iZ_j}, for\ (0 \le i < n, i \le j \le n)$를 distance hamiltonian으로 정의
- i=j에 대해서는 고려할 필요가 없으므로, penalty를 부여.(에너지를 높게 하는 Gate를 추가한다)
- Undirected Graph이므로 diagonal에 대해 대칭인 부분은 고려할 필요 없음. 이 역시 penalty 부여.
- 위 3개를 모두 합친 것을 Cost Hamiltonian으로 정의

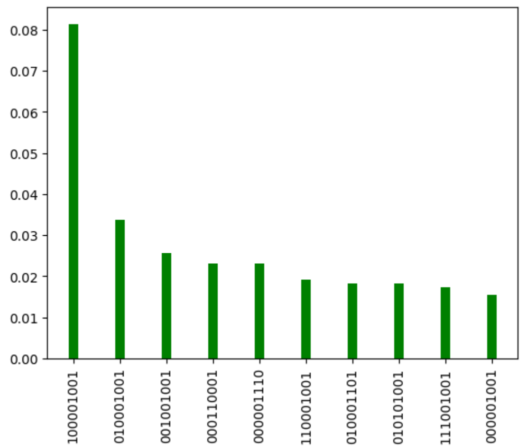
실험 결과(도시 3개)

- 먼저, 도시 3개에 대해 잘 작동하는지를 확인해보자.

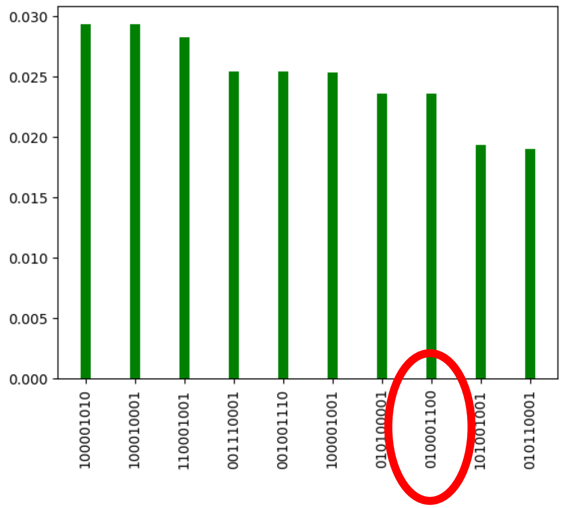




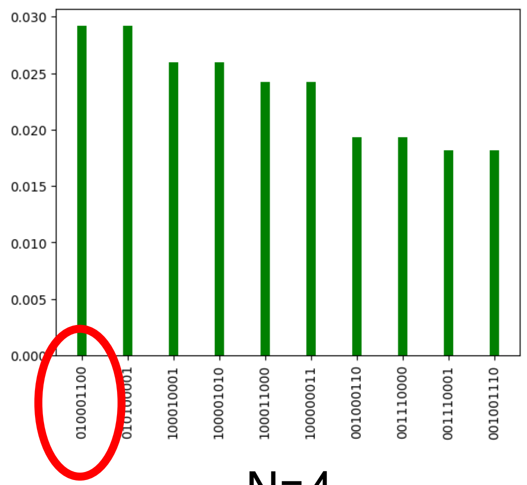
N=1



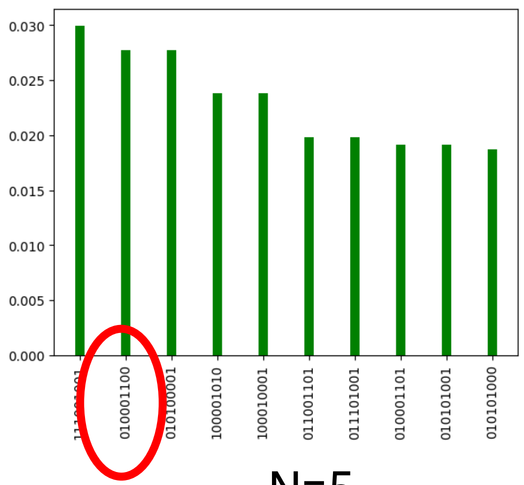
N=2



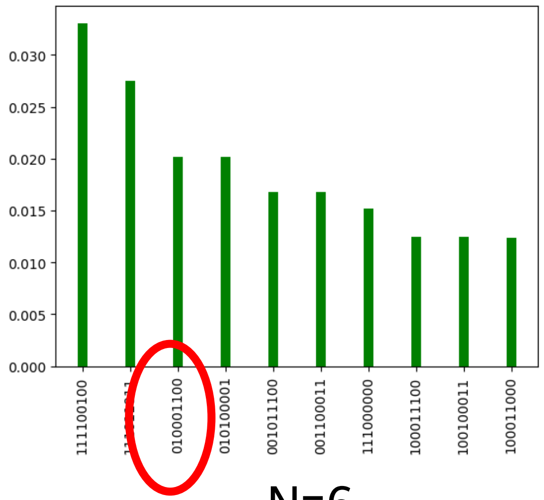
N=3



N=4



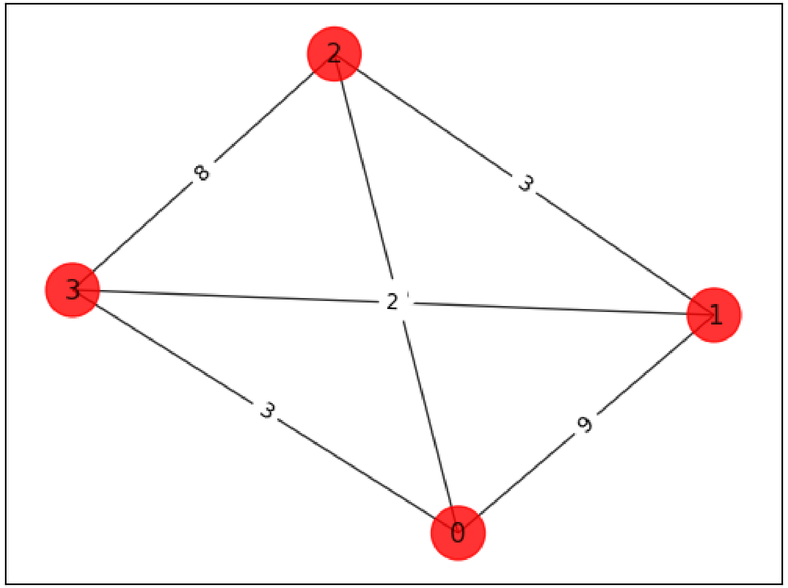
N=5

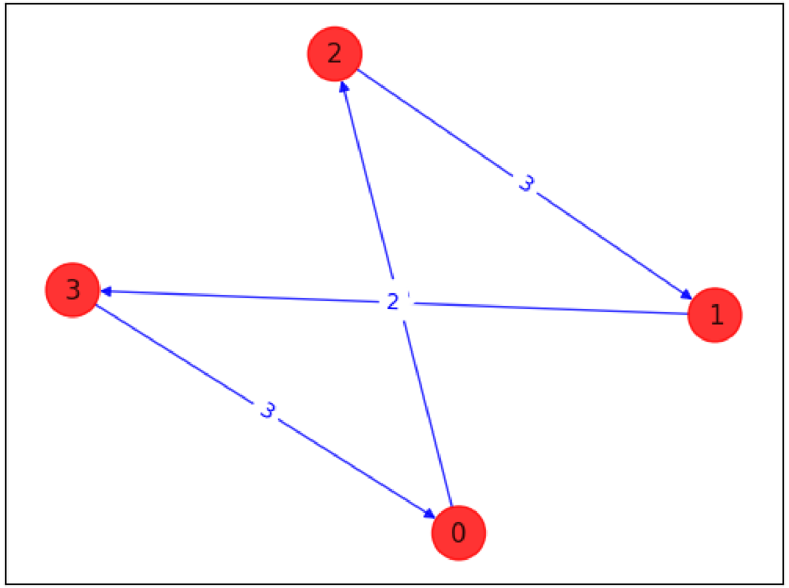


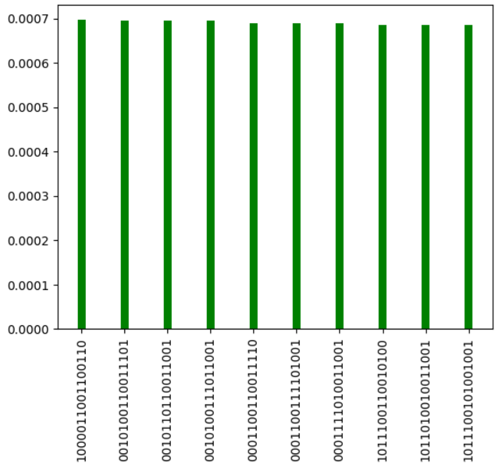
N=6

실험 결과(도시 4개)

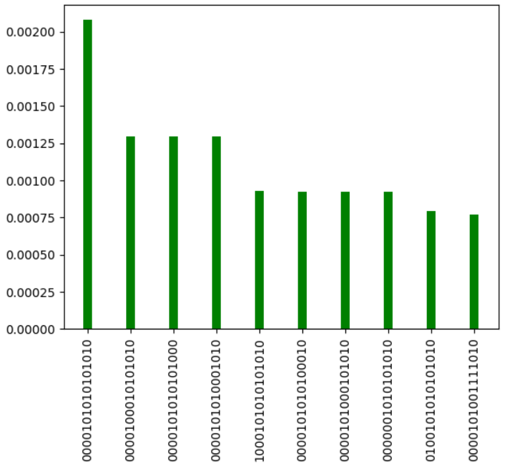
- 도시 4개에 대해서 잘 작동하는지 확인해보자.



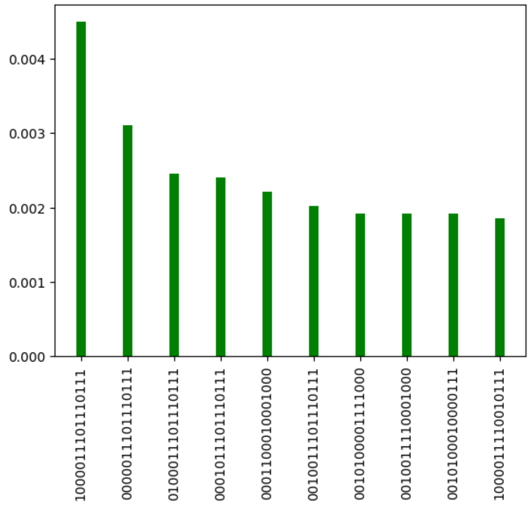




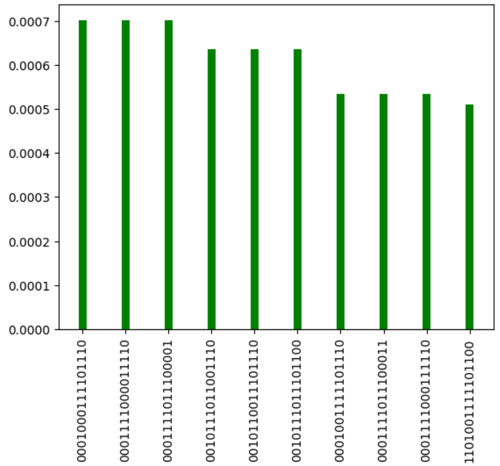
N=1



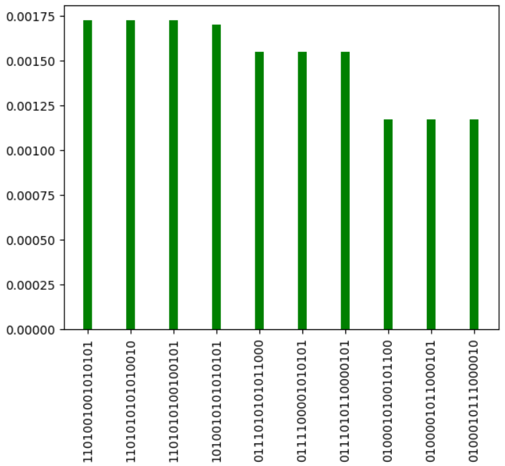
N=2



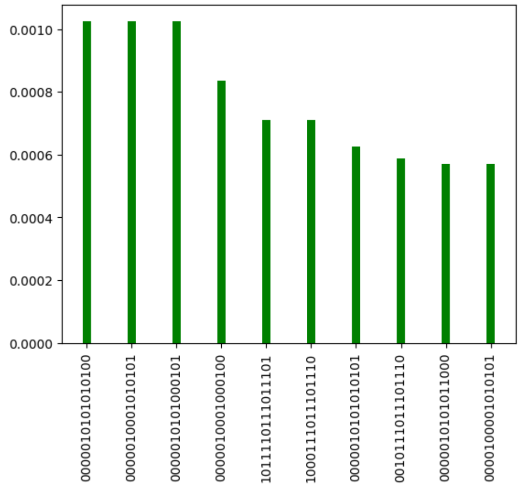
N=3



N=4



N=5



N=6

결과 분석 및 개선사항

- 3개의 도시에 대해서도 결과에 오류가 많았음. Penalty를 더 강화하는 방법이 필요할 듯.
- 4개의 도시에 대해서는 아예 정답이 나오지 않았다.
- Repetition을 늘려서 실험해보고 싶지만, query 시간이 너무 오래 걸려 제대로 해보지 못했다는 점이 아쉬움
- 5개 이상의 도시에 대해서는 중간에 컴퓨터가 뻗어서 제대로 못했다는 점도 아쉬움.
- 도시의 개수 ** 2개의 qubit이 필요하고, repetition이 반복됨에 따라 결과가 더 정확해지는 경향을 확인했으므로, 조금 더 손을 보면 TSP문제를 푸는데 도움이 될지도?

참고문헌

- A QAOA solution to the traveling salesman problem using pyQuil, Matthew Radzihovsky et al.
- Comparative study of variations in quantum approximate optimization algorithms for the Traveling Salesman Problem, Wenyang Qian et al. arXiv:2307.07243