Reinforcement Learning for Resource Generation in Quantum Networks

Efe Aksel Tacettin
E.A.Tacettin@student.tudelft.nl

Supervisor: Bethany Davies
Responsible Professor: Gayane Vardoyan

1. Introduction

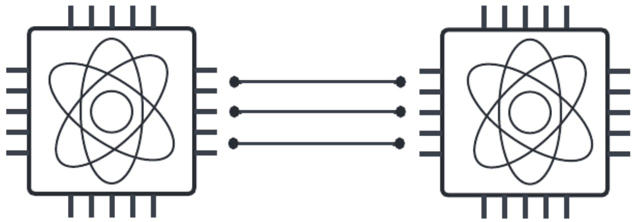

Figure 1: A representation of entangled links between two quantum processors

Quantum networks require entangled links as a resource to carry out applications.

Entangled generation is carried out by protocols with a **probability of success** and **fidelity** (link quality).

When we need multiple links, we store the links in a quantum memory, where the fidelity **decays** exponentially. Below a **threshold** fidelity, links are no longer useful and are **discarded** so we need to generate as quickly as possible to avoid discarding.

While generating, we can tune the protocol to adjust the probability of success and fidelity. How does this help?

2. Research Question

How useful is it to tune the protocol mid-generation process?

How finely should we tune the protocol? A fixed number of different tunings? Or a continuous parameter?

Can we learn heuristics from the policies we find?

5. Results

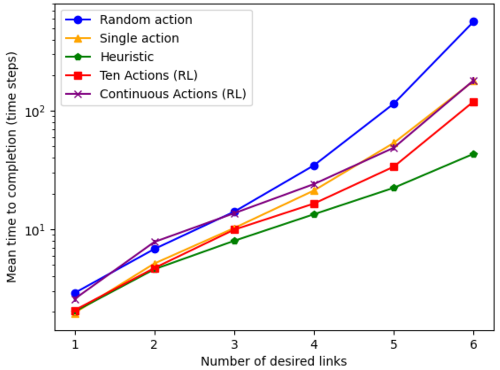

Figure 4: RL and heuristic policies compared to baselines to achieve desired numbers of links

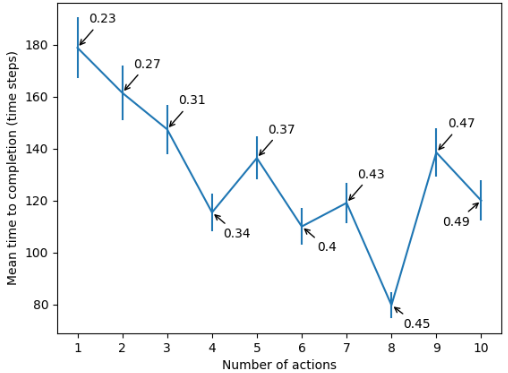

Figure 5: Number of actions versus mean time to achieve desired number of links. Values at each point indicate the probability of the next action.

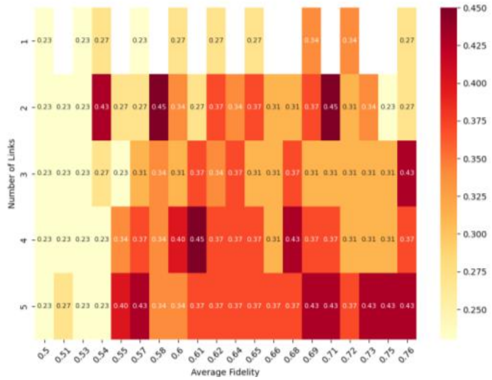

Figure 6: Heatmap for eight action RL policy for number of links and average fidelity

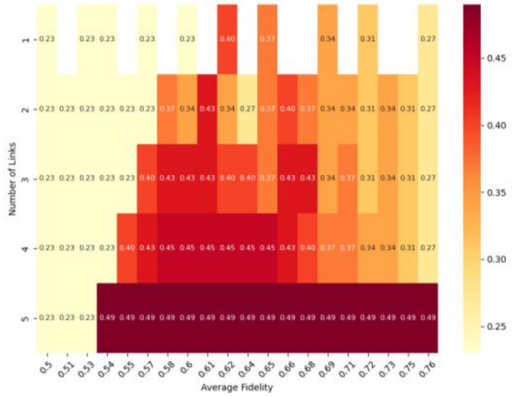

Figure 7: Heatmap for heuristic policy actions for number of links and average fidelity

6. Conclusion

Tuning of the protocol can improve mean time to achieve desired number of links. The limit should be the number of actions versus of number of bins.

It is difficult for RL policies to find good solutions in large numbers of links and bins. Heuristic might be more viable.

Future work can use more advanced RL algorithms like SAC or PPO for continuous action spaces.

3. Problem Modelling

Markov Decision Process:

State Space:

- Can group links between fidelities ranges into bins

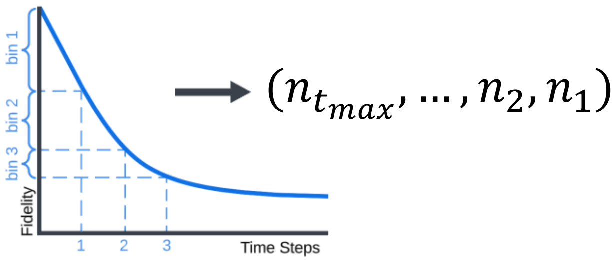

Figure 2: Transforming continuous state space to a discrete space

Action Space:

- Continuous: parameter $p \in [0, 1]$, and $F = 1 - \lambda p$
- Discrete: list of $((F_1, p_1), (F_2, p_2), \dots, (F_n, p_n))$

Reward:

- 0 if all links achieved, -1 otherwise

4. Methodology

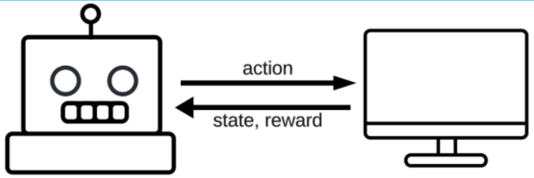

Figure 3: Reinforcement learning feedback loop

Categorical Deep Q-Networks:

- Estimates the distribution of future rewards for each action in a state using a neural network. Good for discrete action spaces.

REINFORCE:

- Directly estimates the policy function with a neural network. Able to work on continuous action spaces.

Heuristic:

- If a link already exists and has a possibility of surviving, match the fidelity bin that it will be in for the next link.
- Example: for four desired links : $(1,0,1,0) \rightarrow bin\ 3$