Experiment 1

(with oiriginal reward function given in NAS)

Experiment 2

(with original reward function but some changes in epsilon)

Experiment 3

(with new reward function without considering costs of actions in reward function)

Experiment 4

(with new reward function considering costs of actions)

As we can see from above flowchart, experiments were performed in 4 steps.

Experiment 1 and 2 are based on old reward function, and Experiment 3 and Experiment 4 are for new reward function.

The objective of experiment 1 is to check the performance of dqn agent with old reward function.

The objective of experiment 2 is to check the possibility to upgrade the performance of dqn agent and finds the drawbacks and their reasons of original method.

After experiments 1 and 2, we compared the performance of them with the results given in the preceding paper.

Experiment 3 and 4 are based on new reward function.

From experiments 1 and 2, we recognized the problems in original reward function and update it considering attacking procedures.

To see the effects of terms in reward function, we divided experiments into 3 and 4.

In experiment 3, we only consider the first term of reward function, i.e. value for actions.

The results showed that consideration of attacking procedures could upgrade the performance of dqn agent to avoid infinite loop and makes the training successful to reach a goal.

The problem in experiment 3 was that dqn agent has no ability to consider the length of attacking path.

To solve this, we append the second term of reward function.

In experiment 4, we considered costs of actions and it made almost rewards to be negative values.

The result showed that dqn agent in Experiment 4 could get a goal with smaller path length than experiment 3.

Experiments were concentrated on the selection of reward values for actions in dqn agent and the results showed that we could upgrade the performance of dqn agent largely if we selected the rewards as proper values.