**Qwhale**

**Content from path planning paper**

The Whale Optimization Q-learning Algorithm (WOQLA) is an algorithm for mobile robot

path planning that combines the original WOA with Q-learning initialization. To overcome

the shortcomings of Q-learning, such as slow convergence caused by initialization, WOQLA

optimizes the initialization of the Q-table instead of simply setting the Q-values to zero.

In the initialization phase, WOQLA generates a number (n) of whale populations in a

20 × 20 grid space and uses the Q-value calculation in Eq (1) to calculate the fitness value of

each whale. The position with the highest Q-value represents the best whale position. The

WOA is then used to optimize the Q-value of each whale in the whale population. When the

maximum number of iterations is reached, the initialization of the Q-table by WOA ends. The

original ε-greedy Q-learning is then used for path planning according to the newly obtained

Q-table. Based on the newly initialized Q-table, the Q-value calculation formula is used in the

iterative update of the Q-table. After the iteration is completed, the final Q-table is obtained.

From this Q-table, we can find the path with the largest Q-value, which represents the best

path. In this way, the Q-table containing previous experience is learned through WOA before

the Q-learning algorithm searches, which helps to reduce the subsequent calculation time and

accelerates the speed of Q-learning convergence

Algorithm 1 Pseudo-code of PWOA

Initialize the whales population Xi(i = 1,2,. . .,n)

Pair whales in the population

Calculate the fitness of each search agent

G = the best search agent

while (t < maximum number of iterations)

for each search agent

Update a, A, C, l, and p

if (p<0.5)

if (|A| < 1)

Update the position of the current search agent by the Eq

(2)

else if (|A|�1)

Select a random search agent (Xrand)

Update the position of the current search agent by the Eq

(10)

end if

else if (p�0.5)

Update the position of the current search by the Eq (6)

end if

end for

Check if any search agent goes beyond the search space and amend it

Calculate the fitness of each search agent

Update G if there is a better solution

for each paired individuals

Compare the fitness value of two paired individuals

Update the position of the two paired individuals to a higher

fitness value

end for

t = t+1

end while

return G