



Politechnika
Wrocławska

Solving the optimal path planning of a mobile robot using improved Q-learning - postęp prac

Przemysław Jaskuła 269995@student.pwr.edu.pl

WYDZIAŁ ELEKTRONIKI, FOTONIKI I MIKROSYSTEMÓW

Projekt Specjalnościowy

- 1 Przypomnienie
Q-learning
FPA
- 2 Wyniki
Algorytmy w Matlabie
Wyniki w Ros-Gazebo
- 3 Napotkane problemy
- 4 Następne kroki
- 5 Q & A

Section 1

Przypomnienie

Subsection 1

Q-learning

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \right]$$

s_t : Obecny stan

a_t : Akcja podjęta w stanie s_t

r_{t+1} : Nagroda po stanie s_t

α : Współczynnik nauki

γ : Współczynnik dyskontowy

Algorithm 1: Classical Q-learning algorithm

Begin

Initiate all Q-values, $Q(s, a)$ in Q-table to zero

Select a starting state, $Q(s_1, a_1)$

while (iteration < Max iteration)

while goal is not achieved

 Select an action, a within the available actions in the current state according to the highest Q-value in the next state

 Perform the selected action, a and reward or penalty, r will be given

 Update the Q-value using Equation (1)

 Move the state to new state, s'

end while

end while

end

Subsection 2

FPA

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - g^*)$$

x_i^t : Pyłek i w iteracji t

g^* : Najlepsze rozwiązanie w obecnej iteracji

γ : Współczynnik skalowania

$L(\lambda)$: Siła zapylania

$$x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$$

x_i^t : Pyłek i w iteracji t

$x_j^t - x_k^t$: Pyłki z tego samego gatunku, lecz innych kwiatów

ϵ : Liczba losowa z dystrybucji jednostajnej

Algorithm 2: Flower pollination algorithm

Begin

Define objective function $f(x), x = (x_1, x_2, x_3, \dots, x_d)^T$

Do initialization of a population of n flowers in random positions.

Obtain the best solution g^* in the initial population

Define the range of switch probability $p \in [0, 1]$

while ($t < \text{Max Generation}$)

for $i = 1:n$ (all n flowers in the population)

if ($\text{rand} > p$)

 Draw a (d -dimensional) state vector L from Levy distribution

 Obtain global population using the relation

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - g^*)$$

else

Draw ϵ from a uniform distribution in $[0,1]$

Obtain local pollination using $x_i^{t+1} = x_i^t + \epsilon (x_j^t - x_k^t)$

end if

Evaluate new solutions

If new solutions are better than update population

end for

Find the current best solution g^*

end while

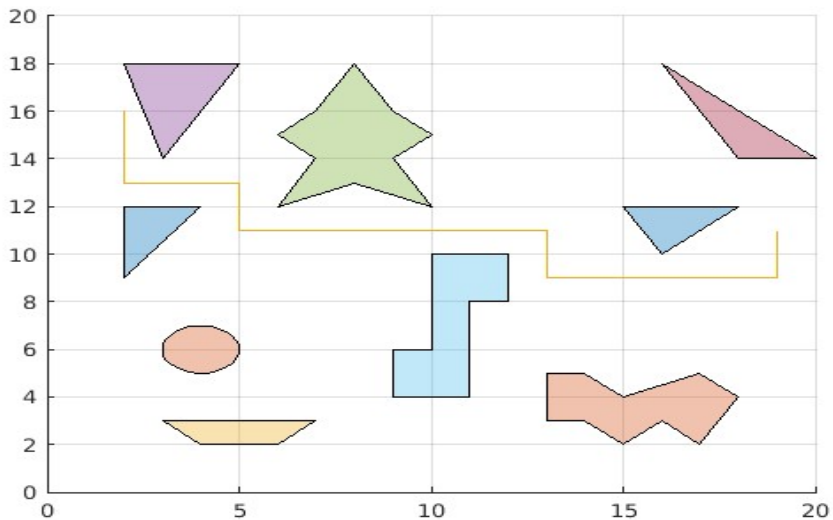
end

Section 2

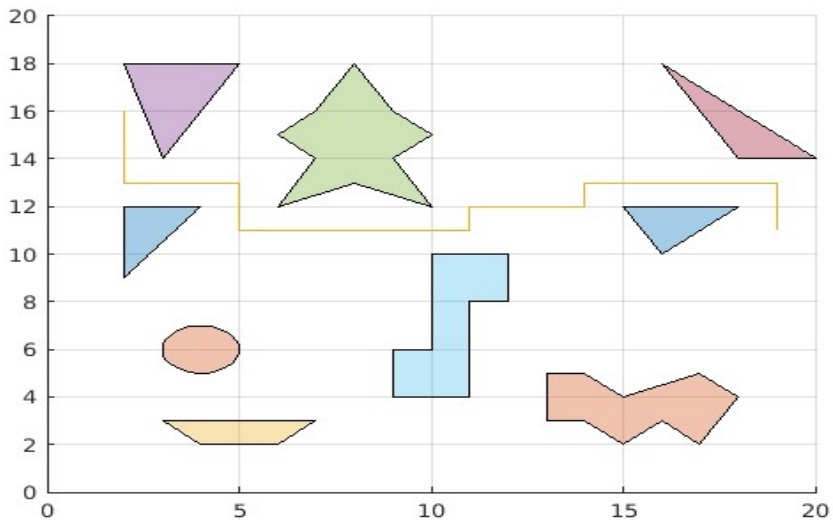
Wyniki

Subsection 1

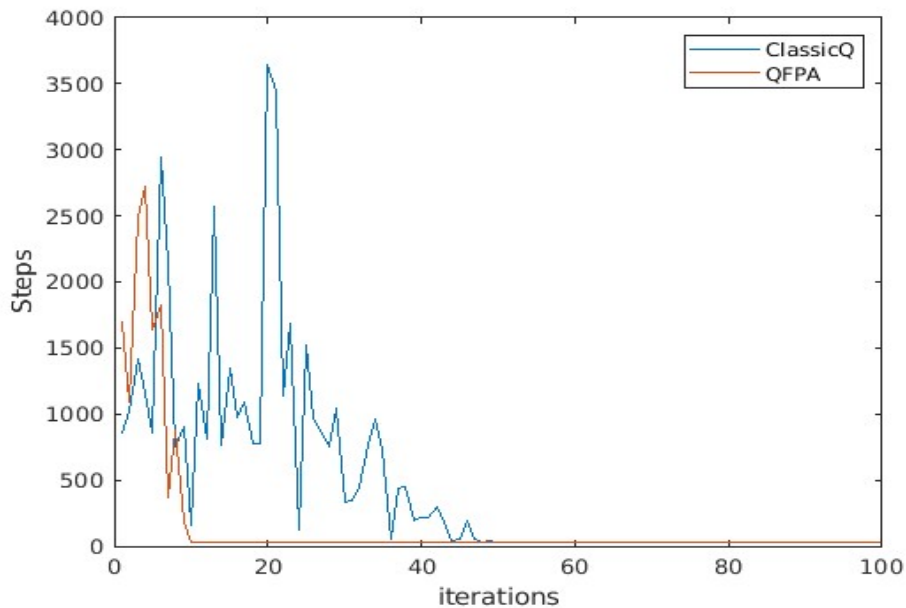
Algorytmy w Matlabie



Classic czas = 1.0034



FPA czas = 0.7780



Subsection 2

Wyniki w Ros-Gazebo

Czas na nagrania!

Section 3

Napotkane problemy

- Blokady w minimach lokalnych(taniec w miejscu)

- ▶ Blokady w minimach lokalnych(taniec w miejscu)
- ▶ Powrót do miejsca po przeszkodzie

- ▶ Blokady w minimach lokalnych(taniec w miejscu)
- ▶ Powrót do miejsca po przeszkodzie
- ▶ Czas obliczeń z Gazebo

- ▶ Blokady w minimach lokalnych(taniec w miejscu)
- ▶ Powrót do miejsca po przeszkodzie
- ▶ Czas obliczeń z Gazebo
- ▶ Kod w matlabie jest zbyt wolny

Section 4

Następne kroki

► Przepisanie wszystkiego w Pythonie

- ▶ Przepisanie wszystkiego w Pythonie
- ▶ Zaimplementowanie większej ilości algorytmów opartych na Q-learningu

- ▶ Przepisanie wszystkiego w Pythonie
- ▶ Zaimplementowanie większej ilości algorytmów opartych na Q-learningu
- ▶ Zaimplementowanie losowego generowania mapy

Section 5

Q & A

Dziękuję za uwagę