

WSI - ćwiczenie 2.

Algorytmy ewolucyjne i genetyczne

17 marca 2023

1 Sprawy organizacyjne

1. Ćwiczenie realizowane jest samodzielnie.
2. Ćwiczenie wykonywane jest w języku Python.
3. Ćwiczenie powinno zostać oddane najpóźniej na 3. zajęciach. W ramach oddawania ćwiczenia należy zademonstrować prowadzącemu działanie kodu oraz utworzyć pull request (z kodem oraz raportem) który prowadzący będzie mógł komentować.
4. Implementacja powinna być zgodna z interfejsem dostępnym w repozytorium <https://gitlab-stud.elka.pw.edu.pl/jlyskawa/wsi-template>
5. Raport powinien być w postaci pliku .pdf, .html albo być częścią notetbooka jupyterowego. Powinien zawierać opis eksperymentów, uzyskane wyniki wraz z komentarzem oraz wnioski.
6. Na ocenę wpływa poprawność oraz jakość kodu i raportu.
7. Można korzystać z pakietów do obliczeń numerycznych, takich jak *numpy*
8. Implementacja powinna być ogólna - w szczególności działać dla dowolnej funkcji $[0, 1]^n \rightarrow \mathbf{R}$

2 Ćwiczenie

Celem ćwiczenia jest implementacja algorytmu genetycznego z mutacją, selekcją ruletkową, krzyżowaniem jednopunktowym oraz sukcesją generacyjną.

Następnie należy zbadać działanie algorytmu na przykładzie problemu opisanego w rozdziale 3.

W tym celu należy znaleźć zestaw hiperparametrów który daje dobry wynik (przynajmniej dodatni), a następnie znaleźć zbadać wpływ wybranego przez siebie hiperparametru.

3 Opis problemu

Zadaniem jest optymalizacja zysku w następującym problemie sterowania dronem:

Kwant czasu trwa 0.1s.

Dron może mieć silnik włączony albo wyłączony przez pierwsze 10 sekund, następnie ma silnik cały czas wyłączony. Jeżeli dron ma silnik włączony, to nadaje mu przyspieszenie $30 \frac{m}{s^2}$. Ponadto działa na niego grawitacja ($10 \frac{m}{s^2}$) oraz spowolnienie wynikające z tarcie wynoszące $-0.1 \frac{m}{s^2} \cdot \frac{s}{m} v$, gdzie v to aktualna prędkość.

Zysk jest równy maksymalnej wysokości jaką osiągnie dron (licząc w metrach) minus 1500 jeżeli w chwili uderzenia w ziemię prędkość drona przekroczy $20 \frac{m}{s}$

Pojedyncza próba trwa do momentu aż dron spadnie na ziemię.

4 Kod symulacji

```
def fun(x):
    # constants:
    T = 100
    g = -10
    dt = 0.1
    P = 30
    r = 0.1
    v_crash = 20
    crash_penalty = 1500

    # simulation
    h = 0
    v = 0
    t = 0

    max_h = 0

    while h >= 0:
        a = g + (P*x[t] if t < T else 0) - v * r
        v += a * dt
        h += v * dt
        t += 1

        max_h = max(h, max_h)

    reward = max_h
    if v < -v_crash:
        reward -= crash_penalty

    return reward
```