Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Wprowadzenie do sztucznej inteligencji

Sprawozdanie z ćwiczenia nr 2

Tymon Kobylecki

Warszawa, 2022

Spis treści

1. Wstęp

W niniejszym sprawozdaniu opisane zostało rozwiązanie zadania oraz eksperymenty dotyczące zadania nr 2. Całość ćwiczenia została wykonana w języku Python.

2. Ćwiczenie

2.1. Eksperymenty

Hiperparametrami zmienianymi podczas eksperymentów były:

— wielkość populacji μ - w zakresie od 10 do 500
— liczba iteracji algorytmu - w zakresie od 2 do 10
— prawdopodobieństwo krzyżowania p_c - od 0,000 000 1 do 0,2
— prawdopodobieństwo mutacji p_m - od 0,000 000 1 do 0,4

Konkretnymi zestawami wykorzystanymi podczas eksperymentów były:

— H_1: iteracje = 10, $\mu = 10$, $p_m = 0{,}1$, $p_c = 0{,}1$
— H_2: iteracje = 10, $\mu = 100$, $p_m = 0{,}1$, $p_c = 0{,}1$
— H_3: iteracje = 3, $\mu = 50$, $p_m = 0{,}01$, $p_c = 0{,}01$
— H_4: iteracje = 2, $\mu = 50$, $p_m = 0{,}2$, $p_c = 0{,}4$
— H_5: iteracje = 2, $\mu = 100$, $p_m = 0{,}2$, $p_c = 0{,}4$
— H_6: iteracje = 5, $\mu = 100$, $p_m = 0{,}001$, $p_c = 0{,}001$
— H_7: iteracje = 7, $\mu = 70$, $p_m = 0{,}001$, $p_c = 0{,}001$
— H_8: iteracje = 10, $\mu = 100$, $p_m = 0{,}000\,000\,1$, $p_c = 0{,}000\,000\,1$
— H_9: iteracje = 2, $\mu = 500$, $p_m = 0{,}001$, $p_c = 0{,}001$

2.2. Funkcja celu

Funkcja celu osiąga swoje maksimum wówczas, gdy rakieta osiąga wysokość 750, lub, jeśli osiągnięcie dokładnie takiej wysokości, następna możliwa wysokość wyższa niż 750. Wysokość końcowa mniejsza niż 750 skutkuje automatycznie wartością funkcji celu równą 0. W pozostałych przypadkach wartość funkcji celu jest zależna od ilości zabranego paliwa, gdyż wówczas wynosi $200 - x$, gdzie x to liczba zabranych jednostek paliwa.

2.3. Wyniki

Z uwagi na wysoką liczebność wyników zerowych, średnia wszystkich wyników nie jest szczególnie miarodajna, bowiem zależy głównie od konkretnej liczby zer w zbiorze wyników dla danego zestawu hiperparametrów. Z tego względu zostanie także przedstawiona średnia wyników niezerowych oraz ich liczba. Każdy zestaw hiperparametrów posłużył do przeprowadzenia 25 pomiarów.

	liczba iteracji	μ	p_m	p_c	średni wynik
H_1	10	10	0,1	0,1	101,12
H_2	10	100	0,1	0,1	110,28
H_3	3	50	0,01	0,01	112,52
H_4	2	50	0,2	0,4	112,8
H_5	2	100	0,2	0,4	115,8
H_6	5	100	0,001	0,001	113,52
H_7	7	70	0,001	0,001	110,92
H_8	10	100	0,000 000 1	0,000 000 1	110,8
H_9	2	500	0,001	0,001	119,52

2.4. Analiza wyników

Algorytm spisywał się najlepiej przy niskiej liczbie generacji. Wyniki są dość porównywalne, przy czym warto zauważyć, że najwyższe średnie zostały uzyskane przy największych populacjach. Wynika to z faktu, że wówczas posiadamy największą pulę, z której można uzyskać rozwiązania. Przy wyższych liczbach iteracji występowało zbyt dużo krzyżowań i mutacji, co skutkowało wysoką losowością otrzymanych wyników. Widoczna jest wartość dodana względem zupełnie losowej populacji, gdyż dla takiej wartość średnia wyniosła 105,84.

2.5. Wnioski

Algorytm genetyczny spisuje się dobrze w takim zadaniu, jednak łatwo wyobrazić sobie, że przy zadaniach „trudniejszych", czyli takich, w których mniejsza liczba kombinacji zapewnia niezerową wartość funkcji celu, algorytm ten może być zupełnie nieskuteczny i „wypluwać" same zera. Wynika to z natury tego algorytmu, która sprawia, że łatwo może dojść do „zgubienia" niezerowych rozwiązań podczas krzyżowania lub mutacji. Taką wadę można jednak łatwo skorygować, na przykład poprzez:

— przechwytywanie każdego niezerowego wyniku i porównywanie wszystkich uzyskanych na koniec działania programu
— wykluczanie najlepszego osobnika w każdej iteracji z krzyżowania i mutacji