

Wydział Elektroniki i Technik Informatycznych
Podstawy Sztucznej Inteligencji
Projekt nr 1 “Przeszukiwanie” - Wykorzystanie algorytmu ewolucyjnego

Wykonawcy:

1. Sebastian Smoliński
2. Oleksandr Drobinin

Spis treści:

1. Opis projektu oraz cele
2. Przedstawienie i realizacja algorytmu
3. Wyniki testów i zmian
4. Podsumowanie pracy

1. Opis projektu i cele

Treść:

Do walizki o ograniczonej pojemności C chcemy załadować przedmioty o jak największej wartości, mając jednak na uwadze, że każdy z nich zajmuje pewną objętość. Mając n przedmiotów wraz z ich wartościami oraz objętościami, znajdź przy użyciu Algorytmu Ewolucyjnego zestaw rzeczy mieszczących się w walizce o największej sumarycznej wartości.

Założenia projektowe i realizacyjne:

1. Celem projektu jest implementacja algorytmu ewolucyjnego, który rozwiązuje problem pomieszczenia jak największej ilości rzeczy N o jak największej wartości w ograniczonej przestrzeni C np. walizki, pudełka.
2. Dane wejściowe:
 - będą wprowadzane z poziomu programu
 - obejmować będą: ilość członków populacji
3. Zostaną wykonane testy algorytmu przy różnych pojemnościach walizki oraz przy różnych ilościach rozpatrywanych przedmiotów
4. Nie korzystamy z gotowych bibliotek zawierających funkcje związane z szeroko pojętą sztuczną inteligencją

Odpowiedzialność za poszczególne części projektu: Sebastian Smoliński: część dokumentacyjna + inicjalizacja zmiennych

Oleksandr Drobinin: pozostała część kodu

2. Przedstawienie i realizacja algorytmu algorytmu

- I. W naszym algorytmie wynikiem ma być znalezienie jak jak najlepszego osobnika z naszej populacji. Jako osobnika w algorytmie traktujemy zbiór przedmiotów zapakowanych do walizki, natomiast gen będzie pojedynczym elementem o wygenerowanej wartości oraz objętości zapakowanym do walizki. Funkcja przystosowania ma na celu znalezienie najlepszego zbioru elementów zapakowanych do plecaka przy obowiązującym ograniczeniu, czyli pojemności.
- II. Generujemy pierwszą populację na podstawie wprowadzonych przez nas wartości: ograniczonej pojemności walizki oraz ilość elementów.
Adnotacja: w algorytmie uznaliśmy, że dla poprawnego działania programu należy wprowadzić dosyć dużą populację początkową oraz odpowiednio dopasowaną wartość pojemności, aby nie doszło do sytuacji, gdzie wygenerowane elementy będą miały zbyt duże wymiary i wartości w stosunku do postawionych wcześniej ograniczeń.
- III. W następnym kroku wykonujemy reprodukcję nowego zbioru osobników z obecnej populacji dobierając nowy zestaw przedmiotów.
- IV. Z otrzymanego zbioru osobników wybieramy nową populację i zwracamy najlepszego osobnika z dostępnych.

3. Wyniki testów i zmian

Celem naszych badań na przygotowanej implementacji było sprawdzenie czasu działania skryptu przy zmiennych danych: pojemności walizki C oraz liczbie przedmiotów N . W tym celu wykonaliśmy kilka testów w celu sprawdzenia czasu działania oraz jakości programu.

Zmienna pojemność walizki C :

Pomiar czasu wykonywania obliczeń					
Pojemność walizki C	26	40	60	80	100
Czas wykonywania	0.76	1.23	3.22	4.56	7.33

Zmienna liczba przedmiotów N :

Pomiar czasu wykonywania obliczeń					
Liczba populacji	10	20	30	40	50
Czas wykonywania	0.09	0.11	0.24	0.47	0.63

4. Podsumowanie pracy

Zaimplementowany algorytm ewolucyjny, mający na celu realizację typowego problemu walizki/pudełka, (w literaturze znany jako „Knapsack problem”), działa zgodnie z początkowymi założeniami oraz treścią zadania.