Sprawozdanie

Autor: Jan Milczarek

1. Polecenie  
   Napisać program umożliwiający znalezienie maksimum funkcji dopasowania jednej zmiennej określonej dla liczb całkowitych w zadanym zakresie przy pomocy elementarnego algorytmu genetycznego (reprodukcja z użyciem nieproporcjonalnej ruletki, krzyżowanie proste, mutacja równomierna). Program powinien umożliwiać użycie różnych funkcji dopasowania, populacji o różnej liczebności oraz różnych parametrów operacji genetycznych (krzyżowania i mutacji). Program powinien zapewnić wizualizację wyników w postaci wykresów średniego, maksymalnego i minimalnego przystosowania dla kolejnych populacji oraz wykresu funkcji w zadanym przedziale.  
   Program przetestować dla funkcji f(x)= -0.2x2+6x+7 dla x= -1, 0, ... 31
2. Instrukcja obsługi  
   Program można uruchomić bezpośrednio z pliku wykonywalnego main.exe w katalogu dist, wszystkie zależności zostały tam uwzględnione. Alternatywnie można uruchomić go z poziomu Microsoft Visual Studio 2022, przebiega to wtedy w następujących krokach:
   1. Rozpakować plik MaksimumFunkcji.zip
   2. Dwukrotnie kliknąć MaksimumFunkcji.sln, spowoduje to uruchomienie programu Microsoft Visual Studio.
   3. Z menu rozwijanego wybrać środowisko „env” dla Pythona 3.9. Jeżeli środowisko nie uruchomi się, należy na dowolnym innym środowiskiem z Pythonem 3.9 uruchomić polecenie pip install, podając plik requirements.txt jako argument (są w nim zawarte wszystkie wymagane zależności).  
      **UWAGA**: **Program nie uruchomi się na środowisku Python 3.11 z uwagi na różnicę w nazewnictwie modułu matplotlib.**
   4. Jeżeli środowisko Pythona jest gotowe, należy w Microsoft Visual Studio wybrać opcję Rozpocznij. Widok powinien być identyczny z tym otrzymanym po uruchomieniu pliku main.exe
   5. Uruchomiony program zawiera cztery pola tekstowe, jedno pole z listą rozwijaną oraz jeden przycisk, wszystkie opisane w języku angielskim. Lista rozwijana pozwala wybrać jedną z trzech funkcji przystosowania (funkcja testowana jest oznaczona numerem 1), w pola opisujące prawdopodobieństwa krzyżowania i mutacji należy wpisać liczby zmiennoprzecinkowe, zaś w pola opisujące liczbę pokoleń i osobników liczby całkowite
   6. Po wpisaniu wszystkich wymaganych parametrów należy kliknąć przycisk „Start algorithm”. Powinno pojawić się wtedy nowe okno prezentujące 4 wykresy: minimalnego przystosowania w zależności od pokolenia, średniego przystosowania, maksymalnego przystosowania i badanej funkcji w przedziale od -1 do 31.
3. Opis wykonanych symulacji  
     
   Pierwsza symulacja została przeprowadzona dla następujących parametrów:  
   - Prawdopodobieństwo mutacji równe 0.001  
   - Prawdopodobieństwo krzyżowania równe 0.8  
   - Liczba pokoleń równa 100  
   - Populacja 50 osobników  
   Wykresy:  
   Obraz zawierający tekst, zrzut ekranu, linia, diagram

   Opis wygenerowany automatycznie  
   Opis: Algorytm przebiegł w sposób prawidłowy, minimalne oraz średnie przystosowanie osobników wykazywały tendencję rosnącą. Maksymalne przystosowanie gwałtownie zmalało w ciągu pierwszych 10 iteracji, prawdopodobnie wyniku niepomyślnego krzyżowania, wzrosło jednak w ostatnich pokoleniach  
     
     
     
     
     
     
     
     
   Druga symulacja została przeprowadzona dla następujących parametrów:  
   - Prawdopodobieństwo mutacji równe 0.001  
   - Prawdopodobieństwo krzyżowania równe 0.5  
   - Liczba pokoleń równa 100  
   - Populacja 50 osobników  
   Wykresy:  
   Obraz zawierający tekst, zrzut ekranu, linia, diagram

   Opis wygenerowany automatycznie  
   Opis: Cała populacja osiągnęła optymalne dopasowanie około 20 pokolenia i utrzymała je aż do 80 pokolenia (z niewielkim spadkiem przy 60 – 70 pokoleniu). Pod koniec wykonywania się programu pojawił się jeden lub więcej suboptymalnych osobników, które zmniejszyły średnie przystosowanie całej populacji. Godnym odnotowania jest fakt że populacja początkowa zawierała osobnika dla którego funkcja przyjmowała swoją wartość maksymalną i osobnik ten nie zginął ani nie zmienił się w ciągu 100 pokoleń (linia prosta na wykresie).  
     
     
     
     
     
     
     
     
     
     
     
     
     
   Trzecia symulacja została przeprowadzona dla następujących parametrów:  
   - Prawdopodobieństwo mutacji równe 0.001  
   - Prawdopodobieństwo krzyżowania równe 0.01  
   - Liczba pokoleń równa 100  
   - Populacja 50 osobników  
   Wykresy:  
   Obraz zawierający tekst, linia, diagram, Wykres

   Opis wygenerowany automatycznie  
   Opis: Algorytm przebiegł poprawnie, średnie przystosowanie populacji było zbliżone do maksimum funkcji, mimo bardzo niskiego prawdopodobieństwa krzyżowania. Jest kilka możliwych przyczyn takiego stanu rzeczy: wąski zakres losowanych osobników w pierwszym pokoleniu (warto zauważyć z wykresu maksymalnego przystosowania, że już w pierwszym pokoleniu pojawił się optymalny osobnik) oraz obecność funkcji reprodukcji (faworyzowała ona optymalnego osobnika z pierwszego pokolenia).  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   Czwarta symulacja została przeprowadzona dla następujących parametrów:  
   - Prawdopodobieństwo mutacji równe 0.01  
   - Prawdopodobieństwo krzyżowania równe 0.8  
   - Liczba pokoleń równa 100  
   - Populacja 50 osobników  
   Wykresy:  
   Obraz zawierający tekst, linia, diagram, Wykres

   Opis wygenerowany automatycznieOpis:  
   Główna różnica w przebiegu algorytmu z wyższym prawdopodobieństwem mutacji jest widoczna na wykresach jako kilkanaście „dolin” w ich przebiegu. Te nagłe spadki w minimalnym i średnim przystosowaniu mogą wynikać właśnie z niekorzystnych mutacji, aczkolwiek mogą być również spowodowane niepomyślnym krzyżowaniem.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   Piąta symulacja została przeprowadzona dla następujących parametrów:  
   - Prawdopodobieństwo mutacji równe 0.1  
   - Prawdopodobieństwo krzyżowania równe 0.8  
   - Liczba pokoleń równa 100  
   Wykresy:  
   Obraz zawierający tekst, pismo odręczne, linia, Czcionka

   Opis wygenerowany automatycznie  
   Opis:  
   Dla tak wysokiego prawdopodobieństwa mutacji algorytm przestaje działać prawidłowo. Podobnie jak w poprzednim przykładzie wykresy posiadają bardzo wiele „dolin”, tym razem jednak wpływają one na działanie dużo mocniej. Krzyżowanie i reprodukcja nie są w stanie zniwelować błędów wynikających z błędnych mutacji.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   Szósta symulacja została przeprowadzona dla następujących parametrów:  
   - Prawdopodobieństwo mutacji równe 0.001  
   - Prawdopodobieństwo krzyżowania równe 0.8  
   - Liczba pokoleń równa 500  
   Wykresy:  
   Obraz zawierający tekst, diagram, linia, zrzut ekranu

   Opis wygenerowany automatycznie  
   Opis: Zwiększenie liczby pokoleń do 500 wpłynęło pozytywnie na przebieg algorytmu, widoczne jest wypłaszczenie wykresów w końcowych 200 pokoleniach. Na wykresach średniego i minimalnego przystosowania, pomiędzy 400 a 500 pokoleniem widać dwie następujące po sobie „doliny”, prawdopodobnie spowodowane wystąpieniem mutacji (wykresy przed pojawieniem się „dolin” były praktycznie wypłaszczone dla wartości maksymalnej funkcji).  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   Siódma symulacja została przeprowadzona dla następujących parametrów:  
   - Prawdopodobieństwo mutacji równe 0.001  
   - Prawdopodobieństwo krzyżowania równe 0.8  
   - Liczba pokoleń równa 10  
   Wykresy:  
   Obraz zawierający tekst, diagram, linia, Wykres

   Opis wygenerowany automatycznie  
   Opis:  
   Na tak małej ilości pokoleń bardzo trudno jest określić skuteczności algorytmu. Wykresy jednakże wykazują tendencję rosnącą, a średnie przystosowanie w ostatnich pokoleniach jest bardzo zbliżone do wartości maksymalnej. Optymalny osobnik pojawił się już w pierwszym pokoleniu i nie zginął (bądź też zginął i w tym samym pokoleniu został zastąpiony identycznym osobnikiem)