# Zasady programowania strukturalnego II – projektowanie

|  |  |  |  |
| --- | --- | --- | --- |
| Imię i Nazwisko | Łukasz Jaśkowski | Grupa | 14 |
| Prowadzący | Mgr inż. Adam Piechna | | |
| Temat projektu | Projekt 7 - Algorytm genetyczny | | |

1. ***Opis projektu***

Program implementujący algorytm genetyczny do rozwiązania problemu optymalizacji – znalezienia minimum wybranej funkcji w określonym przedziale. Program wykorzystuje strukturę dynamiczną danych opartą o ręcznie tworzone listy, a także operacje na plikach(odczyt i zapis).

***1a. Algorytm***

Idea algorytmu opiera się na zjawisku doboru naturalnego i dziedziczenia znanego z nauk przyrodniczych. Zgodnie z tym zjawiskiem osobniki lepiej przystosowane do warunków środowiska mają większą szansę na przetrwanie niż osobniki gorzej przystosowane. Pierwsze z nich przekazują informację genetyczną swoim potomkom. Krzyżowanie informacji genetycznej przodków prowadzi do coraz to lepszego przeciętnego dostosowania się osobników do warunków środowiska. Zachodzi więc tutaj pewien proces optymalizacji.

Przekładając to na temat projektu – pokolenie jest zbiorem osobników o określonej liczebności. Osobnik z kolei reprezentuje pewną liczbę(ew. punkt w przypadku funkcji wielu zmiennych) zawartą w rozpatrywanym w zadaniu przedziale. Genotyp osobnika jest zakodowanym zapisem tej liczby w systemie binarnym. Pierwszym pokoleniem jest pewna ilość osobników, których materiał genetyczny został wylosowany. Funkcja określająca przystosowanie osobnika jest nazywana inaczej funkcją oceny. W tym przypadku jest to funkcja, której minimum szukamy. Podając zdekodowany materiał genetyczny osobnika(czyli de facto liczbę lub punkt, który on reprezentuje) jako argument funkcji celu otrzymujemy wartość funkcji dla tej liczby(punktu). Osobniki dla których funkcja celu przyjmuje niższe wartości przeżywają, natomiast te gorzej przystosowane(czyli z większymi wartościami funkcji oceny) są usuwane. Następnie pomiędzy pozostałymi osobnikami następuje zjawisko krzyżowania(zdefiniowane jako wymiana określonej liczby genów między osobnikami) i powstają nowe osobniki. Dość rzadko, ale tak samo jak w naturze występuje zjawisko mutacji – w projekcie reprezentowane jest poprzez zmianę losowego bitu osobnika na przeciwny. Po przejściu jednego takiego cyklu liczba osobników pozostaje niezmienna, a ich średnie przystosowanie powinno być lepsze niż w pokoleniu poprzedzającym. Następnie pokolenie drugie zostaje poddane takiemu samemu procesowi jak pokolenie pierwsze. Taki cykl zachodzi określoną ilość razy. Na samym końcu z ostatniej populacji wybierany jest osobnik o najlepszym przystosowaniu. Jeżeli wykonaliśmy odpowiednią ilość cykli rozwiązanie to powinno być równe bądź bardzo bliskie szukanemu minimum.

1. ***Opis formatu danych wejściowych/wyjściowych***

Dane wejściowe powinny być zapisane w pliku tekstowym o rozszerzeniu \*.txt. Nazwa pliku jest dowolna – użytkownik programu będzie w odpowiednim momencie poproszony o jej podanie. W pliku powinno znaleźć się 5 liczb całkowitych oddzielonych spacjami, które będą oznaczać kolejno:

* początek przedziału, w którym będziemy szukać minimum,
* koniec przedziału,
* liczbę bitów na których będzie zapisywany materiał genetyczny jednego osobnika; należy pamiętać, że im mniejsza będzie ta liczba tym mniej dokładny będzie wynik(rozdzielczość wyniku okreslona jest wzorem: (koniec\_przedzialu-poczatek\_przedzialu)/(2^liczba\_bitow-1), ale podanie jej zbyt dużej spowoduje dłuższe działanie programu,
* liczbę osobników w jednej populacji,
* liczbę pokoleń;

Dane wyjściowe mają być docelowo zapisywane w pliku tekstowym

„wyniki.txt” i zawierać dwie liczby zmiennoprzecinkowe:

* liczbę określającą współrzędną x-ową szukanego minimum,
* wartość funkcji f(x) dla argumentu z powyższego punktu – czyli szukane minimum funkcji;

1. ***Podział na moduły***

Program(w wersji konsolowej) będzie składał się z czterech modułów:

* Interfejs – moduł odpowiedzialny za komunikację z użytkownikiem,
* Dane – moduł(a jednocześnie nazwa klasy) odpowiedzialny za pobranie(z pliku) i przechowywanie danych wejściowych od użytkownika,
* Osobnik – moduł(a jednocześnie nazwa klasy) odpowiedzialny za operacje na poszczególnych osobnikach populacji,
* Algorytmy – moduł, w którym zawarte są wszelkie inne funkcje(funkcja celu, dekodowanie genotypu, przeprowadzenie cyklu etc.)

.