Głównym polem zastosowań algorytmów genetycznych są zadania optymalizacyjne. Jednym z najpopularniejszych zadań optymalizacyjnych jest poszukiwanie ekstremów funkcji. W ramach tego projektu, tym razem zostanie zastosowana typowa dla klasycznego algorytmu genetycznego reprezentacja chromosomów - reprezentacja binarna.

Podstawowym problemem w implementacji algorytmów genetycznych jest odpowiednia reprezentacja osobników. W niniejszym projekcie populacja będzie zbiorem liczb rzeczywistych z przedziału [-1.5 ; 2.5]. Liczby te możemy zapisać w postaci wektora binarnego. Długość tego wektora zależy od żądanej dokładności rozwiązania (liczby zmiennoprzecinkowej). Przyjmijmy dokładność równą 6 miejsc po przecinku. Dziedzina zmienności x ma długość 4 (długość przedziału [-1.5 ; 2.5] jest równa 4). Żądana dokładność wymaga, aby przedział był podzielony na co najmniej 4 \* 1 000 000 równych podprzedziałów (czyli dziedzina zmienności razy 10^(zadana dokładność)). log2(4000000) jest równy w przybliżeniu 21.93. Oznacza to, że wektor binarny (chromosom) musi mieć 22 bity. Przekształcenie chromosomu w wartość dziesiętną odbywa się poprzez przekształcenie łańcucha binarnego w liczbę rzeczywistą. To przekształcenie jest kluczowe w konstrukcji funkcji oceny (fitness function). Wartość funkcji dopasowania jest równa wartości f(x) zadanej funkcji, gdzie x jest chromosomem w zapisie dziesiętnym. Im wartość funkcji dopasowania niższa, tym rozwiązanie jest bardziej optymalne.

Celem projektu była minimalizacja funkcji x^2+ ( cos(4\*x) )^3 w przedziale [-1.5 ; 2.5] korzystając z algorytmu z ustalonym stanem( ang. steady-state) oraz z krzyżowania dwupunktowego. Użyliśmy następujących stałych konfiguracyjnych:

-wielkość populacji(512) - ilość badanych osobników. Zwiększenie wielkości populacji prowadzi do zwiększenia szans na znalezienie właściwego osobnika, ale jednocześnie zwiększa zapotrzebowanie na zasoby.

-maksymalna liczba iteracji(400) - zwiększenie prowadzi do bardziej dokładnych rozwiązań kosztem czasu wykonywania.

-prawdopodobieństwo mutacji(0.2) – szansa na zmutowanie.

Aby rozwiązać postawiony problem minimalizacji funkcji, należy najpierw zainicjalizować populację. Inicjalizacja populacji polega na losowym przydzieleniu 22 chromosomów każdemu osobnikowi w populacji. Następnie wchodzimy w pętlę, która wykonuje się zadaną liczbę iteracji. W tej pętli na początku obliczamy przystosowanie każdego osobnika w populacji, czyli obliczamy wartość zadanej funkcji dla każdego osobnika, przy uprzedniej zamianie wartości binarnej na wartość rzeczywistą. Po obliczeniu przystosowania należy posortować populację tak, aby na początku były osobniki najlepiej przystosowane, czyli takie, dla których wartość funkcji jest najbliższa minimum. Po sortowaniu korzystamy z algorytmu z ustalonym stanem(steady-state). Polega to na wyborze dwóch losowych osobników z populacji, którzy będą rodzicami dwóch nowych osobników. Nowe osobniki powstają przy użyciu krzyżowania dwupunktowego. Dwa nowo powstałe osobniki zastępują w populacji dwa najmniej przystosowane osobniki. Po wykonaniu zadanej liczby iteracji na ekranie pojawia się najlepiej przystosowany osobnik oraz wartość jego przystosowania.