**Dokumentacja**

**Opis problemu:**

Należy zaimplementować algorytm ewolucyjny lub genetyczny, który stworzy dojeżdżający najdalej model pojazdu o dwóch kołach. Do implementacji należy wykorzystać bibliotekę box2d.

**Dodatkowe założenia:**

Liczba kroków symulacji wynosi 750. Odpowiada ona 12,5 sekundom działania programu przy 60 klatkach na sekundę. Oznacza to, że każda symulacja trwa domyślnie 12,5 sekund i samochody mają tyle czasu na przejazd trasy.

Algorytm działa dla dowolnej trasy.

**Rozwiązanie problemu:**

Genotyp (Reprezentacja samochodu):

Samochód składa się z jednej bryły oraz dwóch kół.

Środek koła znajduje się w jednym z wierzchołków bryły. Koła nie mogą znajdować w tym samym wierzchołku, bo wtedy samochód zachowuje się, jakby miał tylko jedno koło. Nie będzie więc dojeżdżał dalej niż inne pojazdy dwukołowe przez występowanie tarcia.

Bryła składa się z góry ustalonej liczby wierzchołków.

W celu stworzenia bryły reprezentującej nadwozie płaszczyzna w postaci koła jest dzielona na *n* podziałów równej liczbie wierzchołków: <0; 360/n), <360/n; 2\*360/n)… . Następnie każde kolejne położenie wierzchołka jest losowane w jednym z przedziałów.

Koła znajdują się „w dolnej połowie” wierzchołków bryły, to znaczy koła nie mogą się znajdować się gdzie indziej niż na wierzchołkach, które mają pod sobą powierzchnię trasy. Eliminuje to pojazdy, które w zdecydowanej większości przypadków nie mogłyby w ogóle pojechać, bo ich koła nie mają kontaktu z podłożem.

Dodatkowymi parametrami samochodu są promienie, szybkości, siły tarcia oraz momenty obrotowe silnika każdego z kół.

Funkcja celu:

Odległość jaką przejechał pojazd w założonej liczbie kroków symulacji.

Selekcja:

Rodzaje selekcji podlegają badaniom.

Krzyżowanie:

Wybrano krzyżowanie wielopunktowe.

Z dwóch rodziców powstaje dwoje potomków.

Wybiera się losowo z góry założoną liczbę indeksów genotypu. Indeksy te oznaczają w którym momencie należy dokonać zamiany potomka do którego dokonuje się przypisywania genów określonego rodzica.

Przypisywanie genów do potomków następuje w sposób iteracyjny – najpierw gen pierwszego rodzica jest przypisywany do pierwszego potomka, potem gen drugiego rodzica jest przypisywany do drugiego potomka. Gdy numer indeksu genu będzie się zgadzał z numerem znajdującym się w poprzednio wygenerowanej liście, to należy dokonać zamiany potomków do których będzie się przypisywało geny.

Zależy zakończyć przypisywanie, gdy wszystkie geny zostaną przypisane.

Mutacja:

Domyślnie 1/5 nowo powstałych samochodów podlega mutacji.

Polega ona na zastąpieniu dwóch wartości genotypu nowymi losowymi wartościami.

Wybór osobników do następnej populacji:

Poprzednia populacja jest całkowicie zastępowana przez swoich potomków, dlatego też liczba potomków jest równa liczbie rodziców.

Wynika to z tego, że nie znane są wartości funkcji celu potomków, więc wymagane jest przeprowadzenie symulacji, aby wyznaczyć dla nich te wartości.

**Implementacja:**

Kod został napisany w języku Python 3.7.

Zewnętrznymi bibliotekami są PyBox2D oraz PyGame.

PyGame jest wykorzystywany do wizualizacji samochodów.

**Obszary podlegające badaniom:**

Wielkość populacji

Prawdopodobieństwo zajścia mutacji u potomka

Selekcja turniejowa i ruletkowa

**Opis sposobu badania wartości:**

Podczas badań stan początkowy populacji jest zawsze taki sam.

Przeprowadzane są one do 20 generacji po 10 uruchomień.

Jakość danej części algorytmu pod względem szybkości znajdowania co raz lepszego samochodu jest określana na podstawie średniej wartości funkcji celu najlepszego pojazdu z ostatniej generacji każdego uruchomienia programu.

**Domyślne stałe wartości podczas badań:**

Ziarno generatora liczb losowych – 11

Wielkość populacji – badane są wartości 10 i 50

Liczba wierzchołków bryły – 10

Selekcja turniejowa o wielkości turnieju – badane są wartości 5 i 10

Liczba indeksów przy których następuje zamiana podczas krzyżowania - 6

Prawdopodobieństwo zajścia mutacji u potomka – badane są wartości 1/5 i 4/5

Liczba wartości genotypu podlegająca mutacji – 2

Opis trasy znajduje się w pliku track.py.

**Selekcja ruletkowa:**

**Wpływ wielkości populacji w selekcji ruletkowej:**

Z wykresów możemy zaobserwować że algorytm jest mało efektywny pracując na małych populacjach. Spowodowane jest to małą pulą genów populacji początkowej. Również na podstawie wykresów pojedynczego uruchomienia możemy zaobserwować szybko następującą stagnację (już około 10 generacji). Przy dużych populacjach algorytm radzi sobie znacznie lepiej jednak nie eliminuje wcześniej wspomnianej stagnacji. Mimo to w obu przypadkach algorytm znajduje z czasem coraz lepsze rozwiązanie.

**Wpływ prawdopodobieństwa zajścia mutacji:**

Celem zwiększenia współczynnika prawdopodobieństwa zajścia mutacji jest eliminacja wcześnie następującej stagnacji. Z wykresów możemy zaobserwować, że wprowadza ona zniekształcenia jednak nie poprawia efektywności algorytmu.

**Selekcja turniejowa:**

