Informatyka, studia dzienne, mgr jednolite semestr IX

Obliczenia ewolucyjne **2010/2011**

Prowadzący: mgr inż. Łukasz Chomątek wtorek, 14:15

Data oddania: ______________ Ocena: ______________

Cezar Pokorski 138077
Artur Czajka

Zadanie 3: Ewolucyjna optymalizacja kształtu na przykładzie brachistochrony*

1. Opis zadania

Należy pokazać, ze brachistochrona jest łukiem cykloidy stosując np. strategie ewolucyjne. Można wykorzystac funkcje sklejane (punkty kontrolne - zalecane) lub wielomiany aproksymacyjne.

Brachistochrona to krzywa, po której czas staczania się masy punktowej od punktu A do punktu B pod wpływem stałej siły (siły ciężkości) jest najkrótszy.

2. Implementacja

Do rozwiązania tego zadania posłużyliśmy się językiem Python oraz pakietami `matplotlib` i `numpy`.

2.1. Reprezentacja

Jako podstawę rozwiązania utworzyliśmy klasę `Bezier` — jej obiekty reprezentują krzywe Beziéra, stanowiące osobniki naszej populacji. Każda z takich krzywych posiada cztery punkty charakterystyczne, $p_{0..3}$, jednoznacznie opisujące jej kształt. Do rozwiązania problemu przedstawionego w tym zadaniu przyjmujemy wręcz, że punkty p_0 oraz p_3 są ustalone jako początkowy i końcowy punkt pewnego badanego spadku, mają zatem współrzędne

* SVN: `https://aelabnull.googlecode.com/hg/`

$p_0 = A = (0, Y)$, $p_3 = B = (X, 0)$, gdzie X umownie rozumiemy jako szerokość wykresu, Y zaś jako jego wysokość (jest to wysokość, z której stacza się ciało). Drogę pomiędzy nimi wytycza kształt krzywej.

Całość procesu polega zatem na wykorzystaniu strategii ewolucyjnej w celu takiej optymalizacji punktów p_1 i p_2 aby czas staczania się masy punktowej był najkrótszy. Punkty te opisane są za pomocą wektorów współrzędnych rzeczywistych (typu `numpy.array`) zawierających ich współrzędne. Ich początkowe wartości dobierane są losowo z przedziału $x \in (0, X), y \in (-Y, Y)$.

Na populację składa się pewna liczba N (domyślnie 40) osobników, każdy z nich jest pojedynczą krzywą Beziéra. Dla każdego osobnika obliczana jest jego "jakość" jako czas potrzebny do pokonania wyznacoznej przez nią trasy, gdzie najniższy czas oznacza wynik najlepszy (najlepsze przystosowanie).

2.2. Operatory krzyżowania i mutacji

Ponieważ osobniki kodowane są jako zestawienia (czterech) współrzędnych rzeczywistych, operator krzyżowania zdefiniowany jest jako średnia ważona współrzędnych obojga rodziców. Każde krzyżowanie generuje dwóch potomków, z których jeden przyjmuje cechy jednego rodzica z wagą 0.3, a drugiego z wagą 0.7, drugi zaś – odwrotnie. Nic nie stoi na przeszkodzie by zmodyfikować te parametry, jednak w praktyce okazują się dostatecznie dobre (a na pewno bardziej interesujące od średniej arytmetycznej i pojedynczego potomka), a operator krzyżowania oparty o średnią generalnie i tak odgrywa mniejszą rolę, gdyż bardzo ważny staje się operator mutacji.

Mutacja przebiega następująco: wybierana jest losowo jedna z czterech $(p_{1_x}, p_{1_y}, p_{2_x}, p_{2_y})$ współrzędnych, która zostanie poddana modyfikacji, następnie zaś modyfikowana jest o wartość z przedziału odpowiednio $(-\frac{1}{4}X, \frac{1}{4}X)$ lub $(-\frac{1}{4}Y, \frac{1}{4}Y)$, dobraną losowo z rozkładem jednostajnym. Chociaż powoduje to stosunkowo duże modyfikacje, są one tutaj bardzo pożądane i wprowadzają element "eksperymentu" i "świeżości", zapobiegając stagnacji algorytmu.

2.3. Ocena przystosowania

Aby ocenić przystosowanie osobnika, obliczamy przybliżony "czas" staczania masy punktowej wzdłuż wyznaczonej przez niego krzywej. Obliczenia te nie muszą być wysoce precyzyjne, ważne natomiast by pozwalały porównywać osobniki i odzwierciedlały charakter zjawiska, które niejako modelują.

Dla każdego osobnika próbkujemy krzywą dzieląc ją na pewną ilość równych[1] przedziałów.

W każdym z nich zaś obliczamy przyspieszenie a wynikające z działania siły grawitacji na ciało, przy czym każdy taki odcinek traktujemy jak małą równię pochyłą, gdzie $a = g\frac{h}{l}$ (h – wysokość, z której masa opada, l – długość odcinka, jaki przebywa).

Znając a w danym kroku, uaktualniamy prędkość ciała v i sprawdzamy, czy nie doszło do zatrzymania lub wręcz cofania się ciała (gdy $v \leq 0$,

[1] równych tak naprawdę nie w sensie współrzędnej x a raczej parametru t we wzorze $p(t) = \sum_{i=0}^{n} p_i B_i^n(t)$, $t \in [0, 1]$ opisującym wszystkie współrzędne punktu na krzywej. Są to zatem przedziały o równej długości krzywej, a nie równo odległe w x.

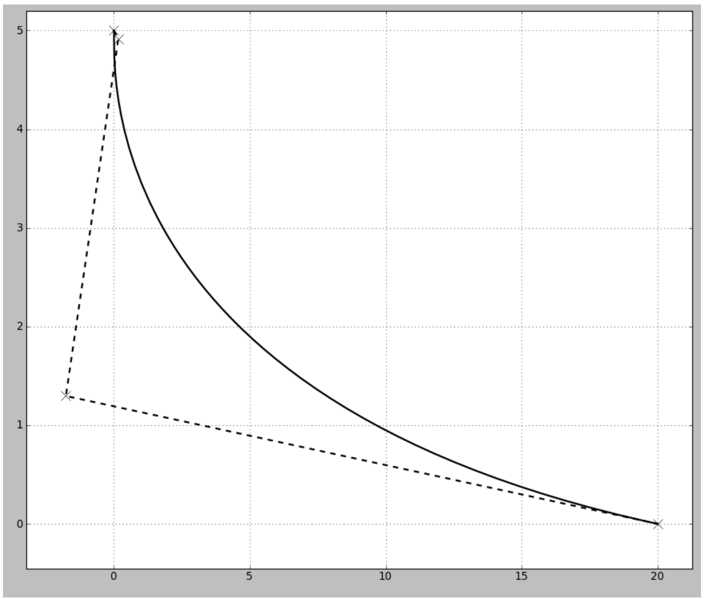
stwierdzamy, że ciało nie stoczy się nigdy i jako czas staczania zwracamy nieskończoność (`numpy.inf`)). Pozwala to wykryć krzywe, które nie miałyby sensu fizycznego.

Po posortowaniu osobników od najlepszego do najgorszego dokonujemy standardowego krzyżowania osobników dopasowanych najlepiej oraz przeprowadzamy ewentualne mutacje, uwzględniając przy tym oczywiście parametry takie, jak prawdopodobieństwo krzyżowania i prawdopodobieństwo mutacji.

Podczas pracy programu można obserwować kształt najlepszej uzyskanej w danej epoce krzywej.

3. Wyniki i wnioski

Końcowym rezultatem działania algorytmu jest zwykle krzywa o poniższym wyglądzie:



Za każdym przebiegiem algorytmu zauważamy, że:
- punkt p_1 praktycznie redukuje się do punktu p_0 w pierwszych iteracjach. Wynika z tego, że taką krzywą można równie skutecznie reprezentować za pomocą jednego tylko punktu charakterystycznego.
- za każdym razem efektem jest krzywa, której początkowy fragment jest "bardziej pionowy", dalej zaś krzywa zbiega do linii prostej. W poczatkowej fazie ruchu nasze ciało nabiera zatem prędkości, która pozwala mu przebyć pozostałą część drogi po trasie bliskiej najkrótszej euklidesowej trasie między tymi punktami.
- w zadaniu tym zwiększenie pradopodobieństwa mutacji pomaga programowi szybciej odnaleźć optymalny kształt krzywej, mimo zaburzeń wprowadzanych przez mutację. Nawet dla prawdopodobieństwa mutacji wynoszącego $P_{mutacji} = 1$ program dąży do podobnego rozwiązania. Zbyt mała mutacja powoduje zaś szybkie uśrednienie wszystkich osobników populacji i zanik różnorodności — a zatem i eksplorowania nowych rozwiązań.