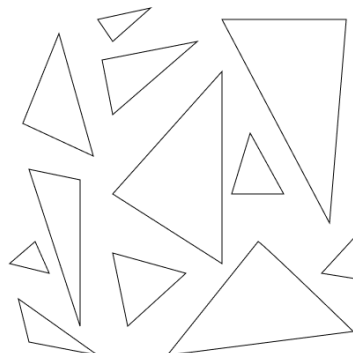


Algorytmy genetyczne 2022/23 – projekt



Opis problemu

Mamy do dyspozycji pewną liczbę prętów stalowych o różnych długościach. Układamy je na taśmie produkcyjnej, jeden za drugim. Robot pobiera z taśmy trzy kolejne pręty i próbuje zespawać z nich trójkąt. Jeśli jest to możliwe, robi to, a jeśli nie, wyrzuca te trzy pręty i bierze kolejne trzy.

Celem projektu jest napisanie algorytmu genetycznego, który ustali taką kolejność prętów na taśmie, aby robot skonstruował jak najwięcej trójkątów i jednocześnie aby odchylenie standardowe ich pól powierzchni od średniego pola powierzchni wszystkich skonstruowanych trójkątów było jak najmniejsze.

Dane wejściowe

Program wczytuje dane wejściowe z pliku `prety.txt`, który w każdej linii zawiera kolejną liczbę całkowitą począwszy od 1, następnie spację i długość jednego pręta. Umawiamy się, że długości prętów podajemy w cm z dokładnością od 0.1. Przykładowy plik `prety.txt` jest załącznikiem do niniejszego opisu.

Dane wyjściowe

Program powinien zapisać rozwiązanie w pliku `output.txt`. W każdej linii powinien się znaleźć numer pręta z pliku `prety.txt`, który należy w danym momencie ułożyć na taśmie. Separatorem dziesiętnym jest kropka. Po każdej liczbie, włącznie z ostatnią, jest znak końca linii. Przykładowy plik `output.txt` jest załącznikiem do niniejszego opisu. Np. ciąg linii:

```
5
3
10
( . . . )
```

oznacza, że na taśmie ma najpierw leżeć pręt o numerze 5, następnie 3, 10 itd.

Założenia wstępne

Liczba prętów w pliku `prety.txt` będzie mniejsza niż 1003 i będzie podzielna przez 3. Długość każdego pręta będzie dodatnią liczbą rzeczywistą mniejszą lub równą 50.

Projekt

Każda osoba pisze własny program, który może wykorzystywać fragmenty programów z laboratoriów. Można także napisać program od zera. Program może wykorzystywać bibliotekę GALib, ale nie musi, ocena nie będzie od tego zależała. Język programowania jest dowolny. Archiwum z projektem powinno się nazywać `AG2223_ImięNazwisko_projekt.zip`. W środku proszę umieścić:

- podkatalog "src" z plikami źródłowymi oraz wszystkimi bibliotekami niestandardowymi (poza GALib), które są potrzebne do skompilowania projektu
- podkatalog "exe", zawierający skompilowaną wersję programu. Ta wersja powinna działać na Windows 10 lub Taurusie bez konieczności instalowania żadnych dodatkowych pakietów, programów itp.
- sprawozdanie z realizacji projektu (PDF, maksimum 5 str. A4) z opisem zastosowanego kodowania, metody reprodukcji, operatorów genetycznych, funkcji dostosowania, testów, którym poddano aplikację i innych elementów, które wydają się Państwu istotne.

Ocenianie

1. Za cały projekt będzie można uzyskać maksymalnie **30 pkt.**
2. Będę oceniał poprawność implementacji, czy algorytm genetyczny działa jak należy. Za poprawnie działający program, napisany bez błędów, będzie można uzyskać maksymalnie **14 pkt.**
3. Za sprawozdanie można będzie uzyskać **7 pkt.**
4. W ocenianiu pojawi się również czynnik rywalizacji. Każdy program zostanie uruchomiony dziesięciokrotnie z różnymi plikami `prety.txt` (dla wszystkich programów jednakowymi). Dla każdego uruchomienia zanotuję ilość skonstruowanych trójkątów i odchylenie standardowe pól powierzchni trójkątów od średniego pola powierzchni, a następnie uśrednię te wielkości dla tych 10 uruchomień. Taki punkt umieszczę na wykresie x-y. Rozwiązania niezdominowane w sensie Pareto (szczegóły na zajęciach) uzyskają **9 pkt.**, te z drugiego frontu Pareto **6 pkt.**, z trzeciego **3 pkt.**, a pozostałe **0 pkt.**
5. Programy z istotnymi błędami implementacji otrzymają w sumie od **0** do **10 pkt.**, niezależnie od generowanych wyników. Autorzy takich programów mogą jeszcze liczyć na punkty za sprawozdanie.

Termin oddania projektów

Gotowe projekty proszę przesłać przez zadanie, które będzie na platformie UPEL w terminie:

- do **25.01.2023 r.** do godz. **23:59** (grupa wtorkowa)

Każdy dzień opóźnienia będzie skutkował obniżeniem punktacji o **4 pkt.**