

Mini-Projekt	Sieć kanałów	Algorytmy genetyczne 2013
--------------	--------------	---------------------------

Opis problemu

Przez kwadratową wyspę o długości boku równej 1 należy poprowadzić sieć prostoliniowych kanałów, tak aby zasilić w wodę jak największą liczbę miast, a jednocześnie aby zachować jak najmniejszą sumaryczną długość sieci kanałów. Kanały mogą się zaczynać i kończyć na dowolnej krawędzi wyspy. Miasta reprezentowane są przez okręgi o zadanym położeniu środka oraz danym promieniu. Przyjmujemy, że kanał zasila miasto jeżeli przecina on okrąg reprezentujący dane miasto.

Dane wejściowe.

Program jako dane wejściowe przyjmuje:

N – liczbę miast

x_i, y_i, r_i – położenie środka oraz promień dla i -tego miasta

Format danych wejściowych.

Dane wejściowe podawane będą w pliku tekstowym o nazwie „param.ini”. W pierwszej linijce pliku znajdują się wartości N .

Następne N linijek zawiera trójki liczb oddzielonych spacją. Pierwsza liczba w trójce to x_i , druga y_i a trzecia to r_i . Wartości współrzędnych liczone są od lewego górnego narożnika wyspy, który ma współrzędne (0,0).

Format danych wyjściowych.

Rozwiązanie powinno zostać zapisane w pliku „nazwisko.txt”.

Pierwsza linijka powinna zawierać liczbę utworzonych kanałów K .

W następnych K linijkach powinny się znaleźć cztery liczby w każdej linijce oddzielone spacjami:

- x_0, y_0 – położenie punktu początkowego kanału
- x_1, y_1 – położenie punktu końcowego kanału

W ostatniej linijce pliku powinny się znaleźć dwie liczby oddzielone spacją: liczba miast, do których doprowadzono wodę oraz suma długości wszystkich kanałów.

Założenia wstępne.

Zakładamy, że N będzie się mogło zmieniać pomiędzy 10 a 30, natomiast K może się zmieniać między 1 a 10.

Kanały mogą się przecinać. Dopuszcza się istnienie kanałów, które nie zasilają żadnego miasta.

Projekt.

Każda osoba pisze własny programik (można się oprzeć na przykładach, które już ćwiczyliśmy), który ma rozwiązywać postawione zadanie. Każdy sam musi zdecydować (zgadnąć, przetestować, poczytać czy poczekać na natchnienie) jakich metod użyć, jakich parametrów krzyżowania i mutacji oraz jakiej liczebności populacji.

Katalog z projektem należy nazwać własnym nazwiskiem i zzipować do pojedynczego pliku o nazwie „nazwisko.zip”.

W katalogu projektu należy utworzyć podkatalog o nazwie EXE, a w nim umieścić skompilowaną wersję programu nazwaną „nazwisko.exe”.

Ocena projektu

1. Każdy program zostanie uruchomiony dziesięciokrotnie z różnymi parametrami (dla wszystkich programów jednakowymi).
2. Dla każdego uruchomienia każdego programu policzony zostanie stosunek liczby zasilonych miast do liczby wszystkich miast dla danych parametrów wejściowych (parametr A).
3. Dla każdego uruchomienia programu policzona zostanie całkowita długość sieci kanałów (parametr B).
4. Wynik każdego uruchomienia programu reprezentować będzie pojedynczy punkt w układzie o osiach A,B.
5. Dla każdego programu policzony zostanie środek ciężkości punktów.

6. Środki ciężkości wszystkich programów zostaną umieszczone na wykresie A,B.
7. Rozwiązania niezdominowane (w sensie Pareto) zdobędą 25 punktów.
8. Rozwiązania odległe od nich o nie więcej niż 5% w każdym kierunku uzyskają 20 punktów.
9. Rozwiązania z drugiego frontu Pareto zdobędą 15 punktów.
10. Rozwiązania odległe od nich o nie więcej niż 5% w każdym kierunku uzyskają 10 punktów.
11. Pozostałe rozwiązanie o ile będą poprawne otrzymają 5 punktów.
12. Programy z istotnymi błędami implementacji uzyskają poniżej 5 punktów (niezależnie od wyniku).
13. Ponadto 5 punktów będzie można zdobyć za dokumentację do mini-projektu. Dokumentacja nie powinna przekraczać 4 stron A4. W dokumentacji należy opisać:
 - a. jak poradzono sobie z optymalizacją dwukryterialną,
 - b. rodzaj zastosowanego kodowania,
 - c. rodzaj użytej mutacji (i dlaczego?),
 - d. rodzaj użytego krzyżowania (i dlaczego?),
 - e. rodzaj użytej selekcji (i dlaczego?),
 - f. sposób wyznaczania funkcji dostosowania,
 - g. inne rozwiązania, jeżeli były stosowane.

Przykładowy plik „param.ini”:

```
4
0.25 0.25 0.1
0.5 0.75 0.2
0.55 0.15 0.1
0.78 0.36 0.24
```

Dodatkowe uwagi.

1. Program nie może korzystać z żadnej dodatkowej wiedzy. Nie można więc np. w ostatecznym rozwiązaniu sprawdzać, które kanały nie przecinają miast i eliminować je z rozwiązania. Rozwiązanie w całości musi zostać znalezione przez algorytm genetyczny.
2. Pojedyncze uruchomienie programu nie powinno trwać dłużej niż 3 minuty na komputerze prowadzącego. Program powinien sam mierzyć czas, po którego upływie powinien zatrzymać ewolucję i zapisać najlepsze rozwiązanie.
3. W trakcie działania program powinien wyświetlać numer aktualnego pokolenia.

Dodatek geometryczny.

Miasto definiujemy jako koło o środku w punkcie $P_0=(x_0,y_0)$ i promieniu R . Kanał to prosta przechodząca przez punkty $P_1=(x_1,y_1)$ i $P_2=(x_2,y_2)$. Miasto ma dostęp do kanału, jeśli prosta przechodzi przez koło. Aby przekonać się czy tak jest wystarczy więc sprawdzić czy odległość P_0 od prostej P_1P_2 jest mniejsza lub równa R

Odległość prostej zdefiniowanej równaniem $ax+by+c=0$ od punktu $P_0=(x_0,y_0)$ wynosi:

$$D = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \quad (*)$$

Prosta przechodzi przez P_1 i P_2 , więc:

$$\begin{cases} ax_1 + by_1 + c = 0 \\ ax_2 + by_2 + c = 0 \end{cases}$$

stąd:

$$a = \frac{y_2 - y_1}{x_1 - x_2} b \quad \text{ i } \quad c = \frac{y_1 x_2 - y_2 x_1}{x_1 - x_2} b$$

Podstawiając do (*) otrzymujemy:

$$D = \frac{\left| \frac{y_2 - y_1}{x_1 - x_2} x_0 + y_0 + \frac{y_1 x_2 - y_2 x_1}{x_1 - x_2} \right| \cdot |b|}{\sqrt{\left(\frac{y_1 - y_2}{x_1 - x_2} \right)^2 + 1} \cdot |b|}$$

Jeżeli $x_1 \neq x_2$ (co jest równoważne warunkowi $b \neq 0$):

$$D = \frac{|x_0(y_2 - y_1) + (x_1 - x_2)y_0 + x_2y_1 - x_1y_2|}{\sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2}} \quad (**)$$

Ale, gdy $x_1 = x_2$ wzór (**) sprowadza się do postaci:

$$D = |x_0 - x_1|$$

a więc też daje dobry wynik. Podsumowując, wzór (**) jest prawdziwy dla dowolnych punktów P_0, P_1, P_2 , jeżeli tylko $P_1 \neq P_2$ (w przeciwnym wypadku prosta nie jest zdefiniowana). Miasto ma dostęp do kanału, jeżeli $D \leq R$.