

Algorytm Mrówkowy - Instrukcja obsługi

Imię i Nazwisko:	Adam Brodacki	Grupa:	MT-123
Temat projektu:	Algorytm mrówkowy szukania optymalnych dróg w grafie		
Opis projektu:	Implementacja algorytmu mrówkowego z dziedziny sztucznej inteligencji do znalezienia najlepszej drogi w zadanym grafie. Graf wczytywany z pliku tekstowego.		

Zasada działania programu

Program służy do wyszukiwania najkrótszej drogi w nieskierowanym grafie. Przeszukiwanie grafu przebiega losowo, przez wiele niezależnych *agentów* -mrówek. Prawdopodobieństwo przejścia mrówki z węzła i do j określa wzór:

$$p(i,j) = \frac{f^A(i,j)/d^B(i,j)}{\sum_{k \in \{\text{nieodwiedz. sąsiednie węzły}\}} f^A(i,k)/d^B(i,k)},$$

gdzie:

$f(i,j)$ - poziom feromonu na krawędzi łączącej węzły i, j ; na początku równy dla wszystkich krawędzi

$d(i,j)$ - odległość węzłów i, j

A -stały wykładnik określający wpływ feromonu na decyzje mrówek

B -stały wykładnik określający wpływ odległości na decyzję mrówek

Po każdym odnalezieniu przez mrówkę węzła końcowego, aktualizowana jest ilość feromonu. Na właśnie zakończonej trasie pojawia się ilość odwrotnie proporcjonalna do jej długości, a z wszystkich krawędzi "odparowuje" pewna jego część. Dzieje się to zgodnie ze wzorami:

$$f(i,j) = R f(i,j), (*)$$

gdzie:

R -współczynnik odparowania

(i,j) -wszystkie krawędzie grafu

$$f(i,j) += 1/L, (**)$$

gdzie:

L -długość trasy znalezionej przez mrówkę

(i,j) -krawędzie należące do trasy o długości L

Rozwiązanie zadania zawarte jest w układzie feromonu. Na jego podstawie, wybierając zawsze najbardziej uczęszczane krawędzie, można wyznaczyć optymalną trasę. Jako warunek zakończenia programu wybrano przekroczenie progu *rozmycia feromonu*. Parametr ten zdefiniowano jako maksymalną na trasie wartość, stosunku ilości feromonu na ścieżkach wychodzących z węzła, ale nienależących do trasy, do sumy feromonu na możliwych do wyboru krawędziach:

$$\max_{(i,j)} \frac{1-f(i,j)}{\sum_{k \in \{\text{sąsiedzi } i\}} f(i,k)}$$

gdzie:

(i,j) -krawędzie należące do optymalnej trasy

Algorytm mrówkowy

```
stworzenie albo wczytanie grafu o W wierzchołkach
stworzenie populacji M mrówek
DOPÓKI rozmycie < ROZMYCIE_MAX I czas symulacji < T_MAX:
    NIECH KAŻDA mrówka:
        JEŚLI ilość krawędzi w jej trasie przekracza W:
            idzie na start
            wybierze losowo kolejny węzeł
            skoczy tam
            obliczy długość swojej trasy
        JEŚLI jest na starcie:
            wyczyści pamięć
        JEŚLI jest na mecie:
            doda feromon na trasie (**)
            wraca na start
    odparowanie feromonu (*)
    znalezienie optymalnej trasy i obliczenie rozmycia
zapisanie optymalnej trasy
```

Tworzenie grafu

Program zawiera moduł odpowiedzialny za stworzenie spójnego grafu o zadanej liczbie wierzchołków, średnią i maksymalną liczbą krawędzi przypadających na wierzchołek. Tworzenie takiego grafu przebiega dwuetapowo. Najpierw łączone są ze sobą wszystkie elementy, tak by zapewnić spójność. Następnie dodawane są połączenia pomiędzy losowymi wierzchołkami, które mają mniej niż maksymalna możliwa ilość połączeń. Dzieje się to dopóki nie zostanie osiągnięta zadana średnia liczba połączeń na wierzchołek. Wylosowane grafy stanowią dane testowe dla programu.

Zapis i odczyt grafu

Program wczytuje i zapisuje graf do plików tekstowych o następującej strukturze:

1. Pierwsza linia zawiera liczbę wierzchołków
2. Każda kolejna składa się z trójki liczb oddzielonych znakami tabulacji lub spacjami. Pierwsze dwie to numery wierzchołków, w numeracji od zera; muszą być mniejsze niż liczba w pierwszej linii. Trzecia zmiennoprzecinkowa liczba określa długość krawędzi.

Zapisywanie trasy do pliku

Wynik działania programu zapisywany jest jako plik tekstowy, którego pierwsza linia zawiera długość trasy, a kolejne numery węzłów ją definiujących.

Obsługa programu

Program umożliwia ustawienie parametrów symulacji, opisanych w pierwszej części instrukcji, wczytanie lub wygenerowanie grafu, określenie początkowego i końcowego węzła.

Obsługa programu przebiega z poziomu linii poleceń. Po wywołaniu z argumentem `--help`, program wyświetla następujące informacje:

AMrowkowy -- Adam Brodacki

Program szuka najkrótszej ścieżki w grafie, pomiędzy dwoma węzłami.

Dostępne opcje:

```
--help          wyświetla tę informację
```

Parametry symulacji:

```
--start arg (=1)          nr węzła startowego
```

```
--meta arg (=2)          nr węzła końcowego
```

```
--m arg (=50)          liczba mrówek
```

--A arg (=4) wykładnik odpowiadający feromonowi

--B arg (=1) wykładnik odpowiadający odległości

```
--R arg (=0.875)      współczynnik odparowania
```

```
--dt arg (=5)          co tyle kroków sprawdza warunek zakończenia i zapisuje
                        stan grafu
```

```
--t arg (=10000)      maksymalna liczba kroków symulacji
```

[illegible]

Wczytywanie, tworzenie i zapisywanie grafu:

```
--if arg      wejściowy plik tekstowy z grafem (użycie tej
              spowoduje pominięcie losowania nowego grafu)
```

Plik zawiera w pierwszej linii liczbę wierzchołków grafu, następne linie zawierają trójki liczb, dwie pierwsze reprezentują nry wierzchołków w numeracji od zera, trzecia odległość tych wierzchołków.

```
--of arg          wyjściowy plik tekstowy z grafem (format pliku i.w.)
```

```
--wg arg (=15)      liczba wierzchołków grafu do wylosowania
```

```
--kga arg (=5)          średnia liczba krawędzi
```

```
--kgm arg (=10)          maksymalna liczba krawedzi
```

Dane wyjściowe:

```
--p-trasa arg      zapisuje listę wierzchołków trasy do pliku
                   tekstowego o podanej nazwie (kolejne linie zawierają
                   kolejne wierzchołki, z pominięciem startu i mety)
```

```
--q-trasa arg zapisuje graf z wyznaczoną trasą w formacie dot
```

```
--g-feromon arg      zapisuje graf z końcowym stanem feromonu w formacie
                        dot
```

```
--g-dt arg          szkielet nazwy pliku na stany pośrednie, zapisywane co
                    dt kroków, formacie dot
```

Dokumentacja kodu źródłowego

Dokumentację kodu źródłowego stworzono w Doxygen.