

GIS 18Z – zadania projektowe

doc. dr inż. Dariusz Bursztynowski

darek@tele.pw.edu.pl

Konsultacje:

poniedziałek, godz. 12-14 lub w innym uzgodnionym terminie.

pok. 348, tel. 508276739

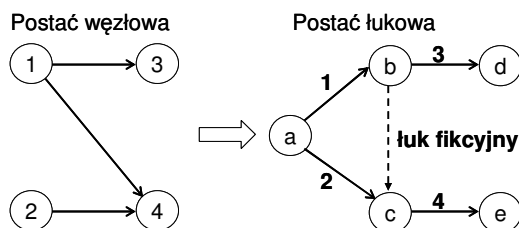
UWAGA: w temacie emaili proszę umieszczać: (GIS)

DB1. Łukowy graf czynności dla zadanego grafu węzłowego.

Wyznaczyć łukowy graf czynności na podstawie węzłowego grafu czynności.

Podczas konsultacji szczegóły zadania mogą zostać skorygowane.

Wyjaśnienie: Planem czynności jest graf określający kolejność wykonywania czynności tworzących zadany zbiór. Węzłowym planem czynności jest graf, w którym węzłami są opisywane czynności (z podaniem ich parametrów, np. czasu realizacji), a skierowane łuki określają (częściowy) porządek realizacji zadań. Łuk skierowany z węzła i do węzła j oznacza, że czynność j można rozpocząć dopiero po zakończeniu czynności i. Łukowym planem czynności jest natomiast graf, w którym czynności są opisywane przez skierowane łuki, a węzły w takim grafie odpowiadają zdarzeniom. Zdarzenie takie oznacza, że zakończono wszystkie czynności reprezentowane przez łuki wchodzące i można rozpocząć czynności odpowiadające łukom wychodzącym. Uwaga: w przypadku, gdy dana czynność ma więcej niż jednego następnika (poprzedza wiele innych czynności), wyznaczenie grafu łukowego wymaga stosowania łuków fikcyjnych – o zerowym czasie realizacji. Przykład użycia łuku fikcyjnego – por. rysunek Plan_czynnosci.pdf w materiałach.



DB2. Wyznaczenie sumarycznie najkrótszych ścieżek rozłącznych węzłowo z użyciem algorytmu Surballe’a.

Dla dowolnej pary węzłów i, j w skierowanym grafie G wyznaczyć dwie sumarycznie najkrótsze ścieżki rozłączne węzłowo. W tym przypadku ścieżki rozłączne węzłowo to takie ścieżki, które nie zawierają wspólnych węzłów (oczywiście z pominięciem i, j); ścieżki takie są jednocześnie rozłączne krawędziowo. Należy zaadaptować algorytm Suurballe (oraz klasyczny algorytm Dijkstry).

Podczas konsultacji szczegóły zadania mogą zostać skorygowane.

J.W. Surballe, R.E. Trajan, A quick method for finding shortest pairs of disjoint paths, Networks, vol. 14, 1984.

Algorytm – patrz plik suurballe_alg.pdf w materiałach.

również dostępny na: http://www.cse.yorku.ca/course_archive/2007-08/F/6590/Notes/surballe_alg.pdf

DB3. Znajdowanie ścieżek o największej i najmniejszej wolnej przepustowości.

Niech w grafie skierowanym G wagi łączy oznaczają dostępną przepustowość (pasmo) sieci transmisyjnej. Pomiedzy parą dowolnych węzłów należy wyznaczyć ścieżki o najmniejszej i największej dostępnej (wolnej) przepustowości. Przepustowość takich ścieżek jest określona przez najmniejszą dostępną przepustowość łączy na ścieżce.

Podczas konsultacji szczegóły zadania mogą zostać skorygowane.

dr hab. inż. Krzysztof Pieńkosz

K.Pienkosz@ia.pw.edu.pl

Konsultacje:

czwartki: godz. 18:10 - 19:00

wtorki: godz. 13:00-14:00

pok. 560A

KP1. Wyznaczanie najkrótszych ścieżek połączeń

Dana jest sieć acykliczna z wierzchołkiem początkowym s i końcowym t . Każdy łuk sieci ma określoną przepustowość. Trzeba przesłać F jednostek z wierzchołka s do t ścieżkami połączeń, tak aby nie przekroczyć przepustowości łuków i liczba łuków najdłuższej z tych ścieżek była jak najmniejsza.

Literatura:

Wojciechowski J, Pieńkosz K., Grafy i Sieci, PWN Warszawa.

KP2. Wyznaczanie najmniejszej liczby ścieżek połączeń

Dana jest sieć acykliczna z wierzchołkiem początkowym s i końcowym t . Każdy łuk sieci ma określoną przepustowość. Trzeba przesłać F jednostek z wierzchołka s do t ścieżkami połączeń, tak aby nie przekroczyć przepustowości łuków i liczba ścieżek połączeń była jak najmniejsza.

Literatura: Wojciechowski J, Pieńkosz K., Grafy i Sieci, PWN Warszawa.

KP3. Wyznaczenie najmniejszego zbioru dominującego w grafie

Zbiorem dominującym grafu $G=(V,E)$ nazywamy taki podzbiór jego wierzchołków U , że każdy wierzchołek, który nie należy do U ma w tym zbiorze sąsiada. W ramach projektu należy opracować algorytm i program do wyznaczania najmniejszego zbioru dominującego w zadanym grafie.

Literatura:

Alina Sampan A., Traian Marius Truta T.M., Beckerach M., Fast Dominating Set Algorithms for Social Networks,

Johan M.M. van Rooij J.M.M, Bodlaender H.L., Exact algorithms for dominating set, Discrete Applied Mathematics 159 (2011) 2147–2164

dr inż. Sebastian Kozłowski

S.Kozlowski@ire.pw.edu.pl

Konsultacje:

piątek 12.15, pok. 444

SK1. Generator sieci bezskalowej I

Należy zaimplementować generator sieci opisanej w podanej publikacji. Aplikacja powinna być przenośna. Wygenerowane grafy powinny być zapisywane do pliku w uzgodnionym formacie. Sprawdzić, czy otrzymywane rozkłady stopni wierzchołków zgadzają się z przewidywaniami teoretycznymi.

Literatura:

E.R.Colman, G.J.Rodgers, „Complex scale-free networks with tunable power-law exponent and clustering”, Physica A: Statistical Mechanics and its Applications, Vol. 392, Issue 21, 1 Nov. 2013, pp. 5501-5510

SK2. Generator sieci bezskalowej II

Należy zaimplementować szybki i przenośny generator sieci Barabasi-Alberta o dużych rozmiarach. Wygenerowane grafy powinny być zapisywane do pliku w uzgodnionym formacie. Sprawdzić, czy otrzymywane rozkłady stopni wierzchołków zgadzają się z przewidywaniami teoretycznymi.

SK3. Narzędzie do masowej analizy i przetwarzania grafów

Należy zaimplementować przenośną aplikację do analizy i przetwarzania dużej liczby grafów, podanych w uzgodniony z prowadzącym sposób. Analiza powinna obejmować co najmniej wyznaczenie rozkładu stopni wierzchołków, rozkładu współczynnika klasteryzacji i rozkładu długości dróg pomiędzy wierzchołkami. Przetwarzanie powinno obejmować zapis grafów w różnych formatach.

SK4. Wizualizacja grafu w przestrzeni 2D

Celem projektu jest opracowanie i implementacja w Matlabie oprogramowania do wizualizacji grafu w przestrzeni 2D metodą Fruchtermana-Reingolda.

Literatura:

T. M. J. Fruchterman, E. M. Reingold, „Graph Drawing by Force-directed Placement”, Software – Practice and Experience, vol. 21(11), Nov. 1991, pp. 1129-1164.

SK5. Problem chińskiego listonosza w obecności ulic jednokierunkowych

Należy zaimplementować (nietrywialny) algorytm rozwiązujący problem chińskiego listonosza, a następnie zbadać, czy można modelować ulice jednokierunkowe w ten sposób, że połączeniu w kierunku $A \rightarrow B$ (A, B - wybrane skrzyżowania) nadaje się wagę równą odległości między

skrzyżowaniami, a połączeniu $B \rightarrow A$ nadaje się wagę o bardzo dużej wartości. Głównym celem zadania jest identyfikacja warunków, w których algorytm wskazuje drogę „pod prąd” pomimo istnienia drogi „legalnej”.

SK6. Badanie właściwości grafów euklidesowych

Należy zaimplementować generator sieci euklidesowych, a następnie zbadać prawdopodobieństwo spójności sieci i określić rozmiar największej składowej spójnej w zależności od liczby i zasięgu wierzchołków.

Literatura:

J. Wojciechowski, K. Pieńkosz, „Grafy i sieci”, Wydawnictwo Naukowe PWN, 2013.

SK7. Izomorfizm drzew

Należy zaimplementować i przebadać dowolny algorytm rozwiązujący problem izomorfizmu drzew nieukorzenionych.

SK8. Atak na sieć

Dane są dwie sieci: bezskalowa (model B-A) i małego świata (model W-S) o mniej więcej takiej samej liczbie wierzchołków i krawędzi. Porównać prawdopodobieństwa powodzenia ataku na losowe wierzchołki tych sieci (udany atak to taki, który prowadzi do rozspójnienia sieci).

SK9. Przydział kanałów w sieci - kolorowanie ścieżek

Dana jest sieć do transmisji danych, składająca się z węzłów i połączeń między nimi. Dana jest również lista par węzłów (tzw. zgłoszeń), pomiędzy którymi w danej chwili należy zestawić połączenia, polegające na wyborze odpowiednich ścieżek pomiędzy węzłami tworzącymi zgłoszenie. Każde fizyczne połączenie pomiędzy dwoma węzłami umożliwia transmisję w kilku kanałach równocześnie. Należy obsłużyć zgłoszenia w taki sposób, żeby transmisja wzdłuż całej ścieżki odbywała się w tym samym kanale, zajmując przy tym jak najmniejszą liczbę kanałów.

Literatura:

M. Kubale, Optymalizacja dyskretna, modele i metody kolorowania grafów, WNT, Warszawa 2002

SK10. Porównanie implementacji wybranego algorytmu grafowego w różnych językach programowania

Wybrany algorytm grafowy (do ustalenia z prowadzącym) należy zaimplementować w C/C++ oraz w Javie i zbadać czas trwania obliczeń dla tych dwóch implementacji przy takich samych parametrach wejściowych.