Konspekt

Piotr Chołda

17 października 2017

1 Podstawowe informacje nt. przedmiotu

1.1 Dane nt. prowadzących

- Prowadzący przedmiot (wykład, egzamin):
  - dr hab. inż. Piotr Chołda;
  - pok. 113 (wejście przez sekretariat), pawilon D-5, piętro
  - konsultacje: czwartki, 14.45-16.16 + można się umówić mailowo;
  - tel.: (617–)26–16;
  - e-mail: piotr.cholda@agh.edu.pl;
  - materiały, informacje, kalendarz zajęć oraz dostępności prowadzącego: strona WWW przedmiotu.
- Ćwiczenia komputerowe oraz projekty:
  - mgr inż. Andrzej Kamisiński;
  - e-mail: kamisinski@kt.agh.edu.pl.

1.2 Konspekty

- Konspekty będą udostępniane na stronie WWW przedmiotu przed wykładem, w miarę możliwości najpóźniej dzień wcześniej.
- Konspekty czasem będą poprawiane po wykładzie (np. korekta błędów, literówek, ewentualne uzupełnienia), dlatego proszę zwracać uwagę na nowe wersje — oznaczane przez słowo **modyfikacja** na stronie WWW przedmiotu.
- Proszę sobie drukować konspekty i przynosić na wykłady.
- Niekiedy po wykładzie są również udostępniane zwięzłe podsumowania materiału z wykładu.

1.3 Cele kształcenia

Szczegółowe cele kształcenia są dostępne w syllabusie przedmiotu.

W praktyce w ramach przedmiotu będą nauczane i wymagane przede wszystkim następujące umiejętności:

- przekształcania zadanego werbalnie problemu inżynierskiego w zadanie optymalizacji oparte na programowaniu matematycznym (umiejętność formułowania),
- interpretacji znaczenia zadania programowania matematycznego (umiejętność rozumienia),
- implementacji zadania optymalizacji do postaci zrozumiałej dla odpowiedniego oprogramowania (umiejętność rozwiązywania — głównie ćwiczenia komputerowe),
- korzystania z tekstów naukowych oraz wyciągania z nich wniosków (umiejętność kreatywnego zdobywania wiedzy — głównie projekt).

1.4 Zasady związane z zaliczaniem przedmiotu

1. Harmonogram zajęć podano na stronie WWW przedmiotu.
2. Przedmiot obejmuje:
   - wykład (uczęszczanie nie jest obowiązkowe — ale prowadzący oczekuje, że na każdym wykładzie zjawią się co najmniej dwie osoby!),
   - ćwiczenia komputerowe (obowiązkowe),
   - projekt (obowiązkowy).
3. Zaliczenie całego przedmiotu: uzyskanie co najmniej oceny *dst* z ćwiczeń komputerowych, projektu i egzaminu.
4. Warunkiem dopuszczenia do egzaminu jest uzyskanie zaliczenia z ćwiczeń komputerowych oraz projektu (zdobycie z obu typów zajęć co najmniej *dst* — ocenę trzeba mieć wystawioną najpóźniej na dzień przed egzaminem).
5. Ocena końcowa z przedmiotu, uzyskiwana po skutecznym zdaniu egzaminu (tj. po zdobyciu co najmniej *dst*) — średnia arytmetyczna ocen z wagami 35%: laboratorium, 35%: projekt, 30%: egzamin (z uwzględnieniem wszystkich ocen *ndst*).
6. Ćwiczenia komputerowe:
   (a) Dokładne zasady podano na stronie WWW przedmiotu w dokumencie pt. *Informacje ogólne nt. ćwiczeń komputerowych*. Należy się z nim zapoznać, ponieważ stanowi integralną część zasad związanych z zaliczeniem całego przedmiotu.
   (b) Składniki ćwiczeń komputerowych:
      - siedem spotkań (po dwie godziny każde);
      - punktowana aktywność;
      - testy pisane na początku niektórych zajęć;

- sprawdzian praktyczny z użyciem komputera i dowolnych materiałów (ostatnie zajęcia).

(c) Pierwsze ćwiczenia: 15/16 listopada. Na stronie WWW przedmiotu będą umieszczone odpowiednie wytyczne ws. przygotowania się przed zajęciami.

7. Projekt:

   (a) Dokładne zasady podano na stronie WWW przedmiotu w dokumencie pt. *Wytyczne nt. sposobu wykonania projektów*. Należy się z nim zapoznać, ponieważ stanowi integralną część zasad związanych z zaliczeniem całego przedmiotu.
   (b) Możliwa praca grupowa.
   (c) Ze względu na wcześniejsze złe doświadczenia z wykonywaniem projektu, wprowadzono obowiązek spotykania się z prowadzącym w celu sprawnego wykonywania projektu.
   (d) Sprawozdanie z projektu: do PIĘCIU stron standardowego maszynopisu 12 pt (opracowania dłuższe nie będą przyjmowane), oddawane w wersji drukowanej, z załącznikiem elektronicznym.
   (e) Zaliczenie projektu obejmuje publiczną prezentację wyników.

8. Egzamin: ustny.

1.5 Bibliografia do przedmiotu

Treść wykładów jest w dużym stopniu oparta na następujących książkach:

1. Michał Pióro and Deepankar Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers—Elsevier, San Francisco, CA, 2004: książka, na której jest oparta optymalizacyjna część niniejszego kursu.

2. **Poompat Saengudomlert. *Optimization for Communications and Networks.* CRC Press/Science Publishers, Boca Raton, FL, 2012**: metody rozwiązywania różnych problemów optymalizacyjnych w kontekście projektowania sieci.

3. **Alexander A. Stepanov and Daniel E. Rose. *From Mathematics to Generic Programming.* Addison-Wesley, Upper Saddle River, NJ, 2015**: podstawy algebraiczne dla zastosowań teleinformatycznych.

4. Robin J. Wilson. *Wprowadzenie do teorii grafów.* Wydawnictwo Naukowe PWN, Warszawa, 2000: zwięzłe wprowadzenie do teorii grafów.

5. Wykłady z programowania liniowego prof. A. P. Wojdy (Wydział Matematyki Stosowanej AGH).

Do każdego wykładu dołączana jest literatura uzupełniająca, zazwyczaj zawierająca wykładany materiał lub jego rozszerzenie. Można zgłosić się po wymieniane pozycje do prowadzącego, gdyby były problemy z ich zdobyciem.

2 Podstawowe pojęcia teorii grafów

2.1 Podstawowe informacje nt. teorii grafów: powtórka

1. Graf jako abstrakcja sieci telekomunikacyjnej lub komputerowej.

2. Matematyczny opis **grafu prostego**:

   - $G = (V, E)$ (graf jest zbiorem, ale typowo używamy reprezentacji graficznej — uwaga na związane z nią pułapki!; możliwe też inne podejście — algebraiczne),
   - V: wierzchołki (*vertex, pl. vertices*), w praktyce sieciowej nazywane węzłami (*nodes*); rząd grafu (*order*),
   - E: krawędzie (*edge*), $e = \{v, w\} \in E$, $v, w \in V$, $v \neq w$, w praktyce sieciowej nazywane łączami (*links*); rozmiar grafu (*size*),
   - wierzchołki mogą być połączone/sąsiednie (*adjacent*, *neighboring*, *connected*), natomiast wierzchołki i krawędzie — incydentne (*incident*).

3. Stopień węzła (*degree*) $deg(v)$:

   - $|E| = \frac{1}{2}\sum_{v\in V} deg(v)$[1],
   - średni stopień węzła: $\mathbb{E}[deg] = \frac{1}{|V|}\sum_{v\in V} deg(v_i) = 2\frac{|E|}{|V|}$,
   - lemat o uściskach dłoni (o podawaniu rąk, *handshaking theorem*).

4. Macierze podstawowe dla tzw. algebraicznej teorii grafów: macierz sąsiedztwa (*adjacency matrix*) $\mathbf{A} = [a_{ij}]$, $deg(v_i) = \sum_{j=1}^{|V|} a_{ij}$; macierz incydencji (*incidence matrix*).

5. Trasa/marszruta/łańcuch w grafie (*walk*); liczba tras (o określonej długości) między węzłami w grafie może być wyliczona z macierzy $\mathbf{A}$.

6. Ścieżka/droga w grafie (*path*):

   - $< v, w >$ jest n-tką wierzchołków $< v = v_1, v_2, \ldots, v_{n-1}, v_n = w >$, takich że $\{v_i, v_{i+1}\}$ jest krawędzią w grafie i krawędzie nie powtarzają się,
   - długość ścieżki jako liczba krawędzi związanych z taką n-tką $(n-1)$.

7. Cykl (*cycle*) w grafie.

8. Odległość wierzchołków w grafie (*distance*), $dist(i, j)$ (długość najkrótszej ścieżki między wierzchołkami, *shortest path*).

9. Średnica grafu (*diameter*), $d(G)$ (największa odległość).

10. Podgraf (*subgraph*).

11. Spójność grafu (*connectivity*):

    - graf spójny (*connected graph*),

[1] Czasami tę zależność utożsamia się z lematem o uściskach dłoni.

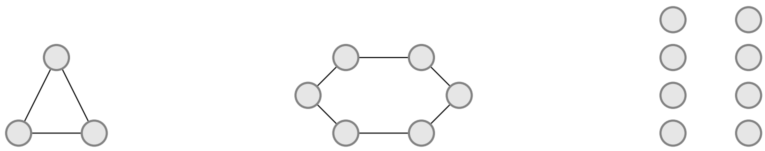
- składowa spójności/komponent/klaster ((*connected*) *component*, *cluster*),
- zbiór rozspajający (*edge partition set*), rozcięcie/przekrój rozdzielający (*edge cut*), most (*bridge*),
- spójność (liczba spójności) krawędziowa $\lambda(G)$ (*edge connectivity*),
- zbiór separujący (*vertex partition set*), separator (*vertex cut*), przegub tj. wierzchołek rozspajający (*articulation point*, *cut vertex*),
- spójność (liczba spójności) wierzchołkowa $\kappa(G)$ (*vertex connectivity*),
- zależność ogólna: $\kappa(G) \leq \lambda(G) \leq deg_{\min} \leq \frac{2|E|}{|V|}$,
- twierdzenia Mengera.

12. Wyróżnione typy grafów:
    - pełny (*full mesh*, *complete graph*) K_n,
    - cykliczny (*cyclic*) C_n,
    - liniowy (*linear*) P_n,
    - dwudzielny (*bipartite*) $G(V_1 \cup V_2, E)$, graf pełny dwudzielny $K_{n,m}$,
    - gwiazda (*star*) $K_{1,n-1}$,
    - hiperkostka/hipersześcian (*hypercube*) Q_n.
13. Drzewo (*tree*) T_n:
    - pojęcie liścia (*leaf*, pl. *leaves*; *pendant*),
    - las (*forest*),
    - $n-1$: liczba krawędzi drzewa o n wierzchołkach,
    - drzewo binarne (*binary tree*), ograniczenie na średnicę drzewa binarnego.
14. Drzewo rozpinające (spinające, *spanning tree*).
15. Graf ważony (*weighted graph*), $G = (V, E, f)$, $f : E \to \mathbb{R}$. Siła (*strength*) węzła.
16. Cykl Eulera i grafy eulerowskie, twierdzenie Eulera, mosty królewieckie.
17. Cykl Hamiltona i grafy hamiltonowskie.
18. Problem chińskiego listonosza (*Chinese postman/route inspection problem*). Problem komiwojażera (*travelling salesman problem*).
19. Graf skierowany (digraf), pojęcie łuku (*arc*), $G = (V, A)$, $A \subseteq \{(x, y) : x, y \in V, x \neq y\}$.
20. Sieć jako digraf ważony, przepływność/przepustowość[2] (*capacity*), przepływ (*flow*), źródło przepływu (*source*), ujście/odpływ przepływu (*sink*), pojęcie łącza nasyconego (*saturated edge/link*).

[2] W języku polskim specjaliści od teorii grafów i optymalizacji często używają tych określeń zamiennie, ale należy pamiętać, że w języku angielskim *capacity* jest właściwością interfejsu (np. przepływność bitowa), natomiast *throughput* (przepustowość) to miara chwilowej jakości działania sieci.

2.2 Zadania

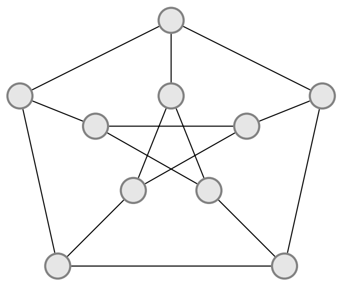
1. Które z grafów pokazanych na rysunku 1 są podgrafami grafu Petersena pokazanego na rysunku 2? Odpowiedź proszę uzasadnić na przykład umieszczając odpowiednie etykiety.



(a) (b) (c)

Rysunek 1: Grafy związane z zadaniem 1.

2. Jaka jest spójność krawędziowa grafu pełnego o 45 krawędziach?
3. Proszę narysować taki graf G, w którym minimalny stopień wierchołka wynosi k i spójności spełniają następujące relacje: $\kappa(G) < \lambda(G) < k$.
4. Proszę narysować graf, w którym istnieje dokładnie sześć cykli. Proszę podać jego średnicę.
5. Czy hiperkostka Q_3 jest grafem dwudzielnym?
6. Czy hiperkostka Q_4 jest grafem eulerowskim?
7. Czy hiperkostka Q_3 jest grafem hamiltonowskim? Jeśli tak, to proszę podać sposób konstruowania cyklu hamiltonowskiego w takim grafie.
8. Jaka jest różnica wartości między spójnością wierzchołkową i spójnością krawędziową hiperkostki o 32 wierzchołkach?
9. Niech n będzie liczbą uzyskaną przez dodanie 10 do liczby reprezentowanej przez ostatnią cyfrą numeru własnego indeksu (np. dla numeru indeksu $xxxxx5$: $n = 15$). Proszę narysować graf prosty, który ma n wierzchołków, jest grafem spójnym i nie zawiera cykli. Proszę znaleźć gęstość tego grafu. Proszę nadać etykiety $1, \ldots, n$ poszczególnym wierzchołkom. Następnie proszę zmodyfikować ten graf, dodając do niego nowe krawędzie w taki sposób, żeby macierz sąsiedztwa $\mathbf{A}$ grafu po modyfikacji charakteryzowała



Rysunek 2: Graf Petersena.

się następującą właściwością: $\forall_{j=1,\ldots,n} \left(\mathbf{A}^2\right)_{jj} = X$ (wartość X proszę sobie wybrać według własnego uznania) i żeby graf zmodyfikowany też był grafem prostym.

2.3 Lektury

2.3.1 Materiał wykładu

Zagadnienia omówione w ramach tego wykładu są w dużym stopniu opisane w następującej pozycji (i to chyba wyczerpuje lektury po polsku do tego przedmiotu ☺):

- Robin J. Wilson. *Wprowadzenie do teorii grafów.* Wydawnictwo Naukowe PWN, Warszawa, 2000 (§ 1-3, § 5-7, § 9).

2.3.2 Bibliografia uzupełniająca

- LiYing Cui, Soundar Kumara, and Réka Albert. Complex Networks: An Engineering View. *IEEE Circuits and Systems Magazine*, 10(3):10–25, third quarter 2010: **bardzo dobry** tekst szkoleniowy nt. podstaw teorii grafów, zagadnienia grafów losowych oraz zastosowania teorii grafów w zoptymalizowanym projektowaniu sieci.
- Robin J. Wilson. *Wprowadzenie do teorii grafów.* Wydawnictwo Naukowe PWN, Warszawa, 2000: zwięzłe wprowadzenie do teorii grafów.
- Piet Van Mieghem. *Graph Spectra for Complex Networks.* Cambridge University Press, Cambridge, UK, 2011: rozdziały początkowe zawierają wiele ciekawych zależności opartych na tzw. algebraicznej teorii grafów.