Matematyka Dyskretna

Wykład 11

Przepływy w sieciach

Wojciech Kordecki
wojciech.kordecki@collegiumwitelona.pl

Collegium Witelona
Wydział Nauk Technicznych i Ekonomicznych
Zakład Informatyki

Semestr letni 2023/24

Wejścia i wyjścia

Niech $G = (V, E)$ będzie spójnym grafem skierowanym bez cykli, $u \in$. Oznaczmy:

- $V^{+}(u)$ – zbiór końców łuków wychodzących z wierzchołka u
- $V^{-}(u)$ – zbiór początków łuków wchodzących do wierzchołka u

Wejścia i wyjścia

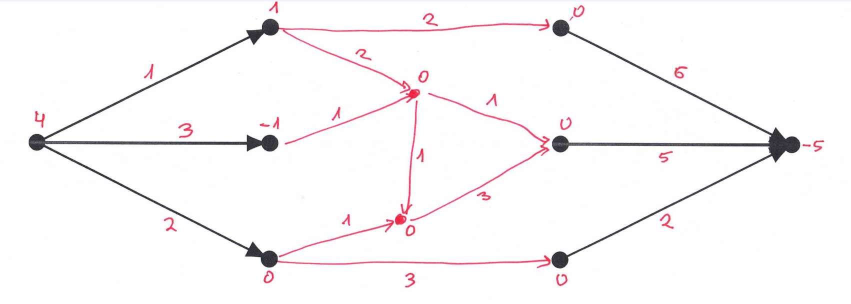
Niech $G = (V, E)$ będzie spójnym grafem skierowanym bez cykli, $u \in$. Oznaczmy:

- $V^+(u)$ – zbiór końców łuków wychodzących z wierzchołka u
- $V^-(u)$ – zbiór początków łuków wchodzących do wierzchołka u

- $V^-(u) = \emptyset$: u jest wejściem w grafie G.
- $V^+(u) = \emptyset$: u jest wyjściem w grafie G.

Wejście i wyjście – ilustracja

Sieć – przykład



Przepływy

Niech $G = (V', E')$ będzie spójnym grafem skierowanym, acyklicznym, bez wielokrotnych łuków. Siecią przepływową nazywamy czwórkę $S' = (V', E', f, g)$, w której $f : V \to \mathbb{Z}$, $g : E \to \mathbb{Z}^+$. Zakładamy, że:

$$f(v) > 0, \text{ gdy } V^-(v) = \emptyset,$$

czyli v jest *wejściem* lub *źródłem* oraz

$$f(v) < 0, \text{ gdy } V^+(v) = \emptyset,$$

czyli v jest *wyjściem*, ściekiem lub *drenem*.

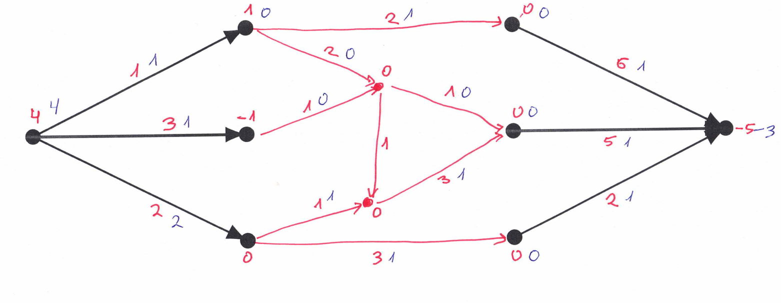
Strumień – definicja

Strumień to dowolna para funkcji (φ, ψ), gdzie $\psi : V' \to \mathbb{Z}$, $\varphi : E' \to \mathbb{Z}^+$, taka, że:

- $\psi(v) f(v) \geq 0$ dla dowolnego $v \in V'$,
- $|\psi(v)| \leq |f(v)|$ dla dowolnego $v \in V'$,
- $\varphi(e) \leq g(e)$ dla dowolnego $e \in E'$,
- $$\sum_{u \in V^-(v)} \varphi((u, v)) + \psi(v) = \sum_{w \in V^+(v)} \varphi((v, w))$$

  dla dowolnego $v \in V'$.

Strumień – przykład



Strumień – interpretacja

Funkcje f, g, ψ oraz φ mają następującą interpretację:

- $f(v)$ jest wydajnością wypływu (gdy $f(v) > 0$) lub wpływu (gdy $f(v) < 0$) wierzchołka v,
- $g(e)$ jest przepustowością łuku e,
- ψ jest wypływem lub wpływem do v,
- φ jest przepływem przez łuk e.

Uproszczenie

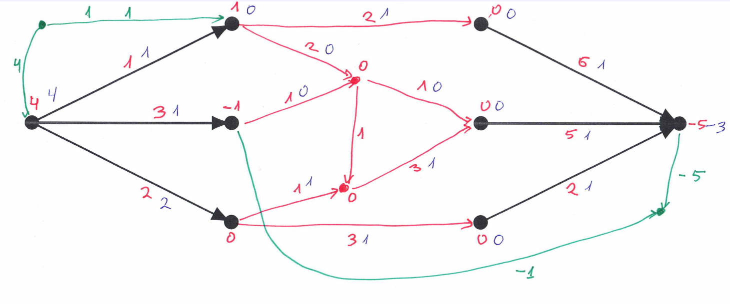
Sieć S' upraszczamy do sieci $S = (V, E, c)$, przyjmując $V = V' \cup \{s, t\}$, gdzie $s \notin V'$ jest wejściem, natomiast $t \notin V'$ jest wyjściem.
Zbiór łuków E' uzupełniamy o łuki:

$$E_t = \{e = (s, v) : f(v) > 0\}, \quad E_s = \{e = (v, t) : f(v) < 0\},$$

a więc $E = E' \cup E_s \cup E_t$.
Funkcję c określamy wzorem:

$$c(e) = \begin{cases} g(u, w) & \text{dla } e = (u, w) \text{ oraz } u, w \in V', \\ f(v) & \text{dla } e = (s, v) \text{ oraz } v \in V', \\ -f(v) & \text{dla } e = (v, t) \text{ oraz } v \in V'. \end{cases}$$

Uproszczenie – przykład



Sieć uproszczona

Sieć S można bezpośrednio określić jako trójkę $S = (V, E, c)$, w której $G = (V, E)$ jest spójnym grafem skierowanym, acyklicznym, bez wielokrotnych łuków oraz o dokładnie jednym wejściu s, to znaczy $V^{-}(s) = \emptyset$ i dokładnie jednym wyjściu t, $V^{+}(t) = \emptyset$.
Ponadto $c : E \to \mathbb{Z}^{+}$. Funkcja c jest *przepustowością* sieci S.

Strumień

W sieci $S = (V, E, c)$ *strumień* określa się jako dowolną funkcję $\varphi : E \to \mathbb{Z}^+$ taką, że:

(A) $\varphi(e) \leq c(e)$ dla każdego łuku $e \in E$,

(B) dla dowolnego wierzchołka $v \in V$ takiego, że $v \neq s$ oraz $v \neq t$, zachodzi równość:

$$\sum_{u \in V^-(v)} \varphi((u, v)) = \sum_{w \in V^+(v)} \varphi((v, w)).$$

Strumień o największej przepustowości

Twierdzenie

Jeśli s jest wejściem, natomiast t jest wyjściem sieci S, to:

$$\sum_{v \in V^{+}(s)} \varphi((s,v)) = \sum_{v \in V^{-}(t)} \varphi((v,t)) = \Phi_{\varphi} .$$

Strumień o największej przepustowości

Twierdzenie

Jeśli s jest wejściem, natomiast t jest wyjściem sieci S, to:

$$\sum_{v \in V^+(s)} \varphi((s,v)) = \sum_{v \in V^-(t)} \varphi((v,t)) = \Phi_\varphi .$$

Liczba Φ_φ występująca po prawej stronie wzoru tego wzoru nazywa się *przepustowością* strumienia φ.

Strumień o największej przepustowości

Twierdzenie

Jeśli s jest wejściem, natomiast t jest wyjściem sieci S, to:

$$\sum_{v \in V^+(s)} \varphi((s,v)) = \sum_{v \in V^-(t)} \varphi((v,t)) = \Phi_\varphi \,.$$

Liczba Φ_φ występująca po prawej stronie wzoru tego wzoru nazywa się *przepustowością* strumienia φ.

Strumień maksymalny definiuje się jako strumień o największej przepustowości.

Przekrój minimalny

Przekrój $C = (U, W)$ określa się jako parę zbiorów taką, że $U \cap W = \emptyset$, $U \cup W = V$, $s \in U$, $t \in W$.

Przekrój jest *normalny*, gdy podgrafy o zbiorach wierzchołków U oraz W są spójne.

Liczbę

$$P(U, W) = \sum_{\substack{u \in U \\ w \in W}} c((u, w)) = P_C$$

nazywa się przepustowością przekroju (U, W).

Przekrój minimalny to przekrój o najmniejszej przepustowości.

Twierdzenie Forda-Fulkersona

W 1955 roku L. R. Ford i D. R. Fulkerson udowodnili twierdzenie wiążące wartości minimalnego przekroju i maksymalnego przepływu.

Twierdzenie Forda-Fulkersona

W 1955 roku L. R. Ford i D. R. Fulkerson udowodnili twierdzenie wiążące wartości minimalnego przekroju i maksymalnego przepływu.

Twierdzenie

W dowolnej sieci przepływowej przepustowość strumienia maksymalnego jest równa przepustowości przekroju minimalnego, to znaczy:

$$\max_{\varphi} \Phi_{\varphi} = \min_{C} P_C \,.$$

Łańcuchy nienasycone

Niech $l = (e_1, e_2, \ldots, e_{m-1})$ będzie łańcuchem o ciągu wierzchołków $(s = v_1, v_2, \ldots, v_m = t)$, gdzie albo $e_i = (v_i, v_{i+1})$, albo $e_i = (v_{i+1}, v_i)$. Jeśli $e_i = (v_i, v_{i+1})$, to łuk e_i jest *zgodnie skierowany* z łańcuchem l, natomiast jeśli $e_i = (v_{i+1}, v_i)$, wtedy e_i jest *przeciwnie skierowany* do łańcucha l.

Łańcuchy nienasycone

Niech $l = (e_1, e_2, \ldots, e_{m-1})$ będzie łańcuchem o ciągu wierzchołków $(s = v_1, v_2, \ldots, v_m = t)$, gdzie albo $e_i = (v_i, v_{i+1})$, albo $e_i = (v_{i+1}, v_i)$. Jeśli $e_i = (v_i, v_{i+1})$, to łuk e_i jest *zgodnie skierowany* z łańcuchem l, natomiast jeśli $e_i = (v_{i+1}, v_i)$, wtedy e_i jest *przeciwnie skierowany* do łańcucha l.

Łańcuch l jest *nienasycony* przez strumień φ, jeśli dla każdego łuku $e \in l$ zgodnie skierowanego z l mamy $\varphi(e) < c(e)$, a dla łuku $e \in l$ przeciwnie skierowanego do l jest $\varphi(e) > 0$.

Warunek konieczny

Twierdzenie

Jeżeli istnieje łańcuch l nienasycony przez strumień φ, to φ nie jest maksymalny.

Modyfikacja stumienia

Niech:

$$\theta = \min_{i,j} \left(c\left(e_i\right) - \varphi\left(e_i\right), \varphi\left(e_j\right) \right),$$

gdzie minimum bierzemy po łukach $e_i \in I$ zgodnie skierowanych i łukach $e_j \in I$ przeciwnie skierowanych.
Określamy nowy strumień:

$$\varphi'(e) = \begin{cases} \varphi(e) & \text{dla } e \notin I, \\ \varphi(e) + \theta & \text{dla } e \text{ skierowanych zgodnie z } I, \\ \varphi(e) - \theta & \text{dla } e \text{ skierowanych przeciwnie do } I. \end{cases}$$

Zauważmy, że φ' jest strumieniem oraz:

$$\Phi_{\varphi'} = \Phi_{\varphi} + \theta > \Phi_{\varphi},$$

więc φ nie jest strumieniem maksymalnym.

Warunek dostateczny

Twierdzenie

Jeżeli sieć przepływowa nie zawiera żadnego łańcucha nienasyconego przez strumień φ, to φ jest maksymalny.

Nasycamy drogi (1)

Twierdzenie

Warunkiem koniecznym maksymalności strumienia φ jest nasycenie co najmniej jednego łuku e (czyli $\varphi(e) = c(e)$) dla każdej drogi łączącej s oraz t.

Niech D będzie zbiorem wszystkich dróg łączących wierzchołek s z wierzchołkiem t. Niech $d_1 = \left(e_1^1, \ldots, e_{m_1}^1\right) \in D$.
Określamy wartości $\varphi_1\left(e_i^1\right) = \min_i c\left(e_i^1\right)$ dla $e_i^1 \in d_1$ oraz $\varphi(e) = 0$ dla $e \notin d_1$.

Nasycamy drogi (2)

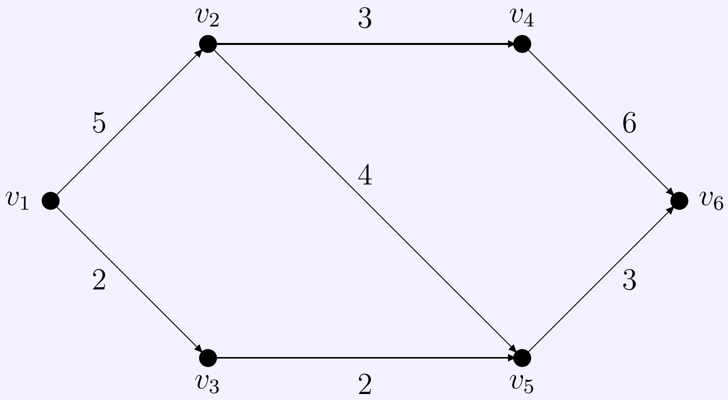
Niech $d_j = \left(e_1^j, \ldots, e_{m_1}^j\right) \in D$ oraz d_j nie zawiera łuków nasyconych przez φ_{j-1}.
Określamy

$$\varphi_j\left(e_i^j\right) = \varphi_{j-1}\left(e_i^j\right) + \min_i\left(c\left(e_i^j\right) - \varphi_{j-1}\left(e_i^j\right)\right)$$

dla $e_i^j \in d_j$ oraz $\varphi_j(e) = \varphi_{j-1}(e)$ dla $e \notin d_j$.
Powtarzamy ten krok dla wszystkich dróg z D.

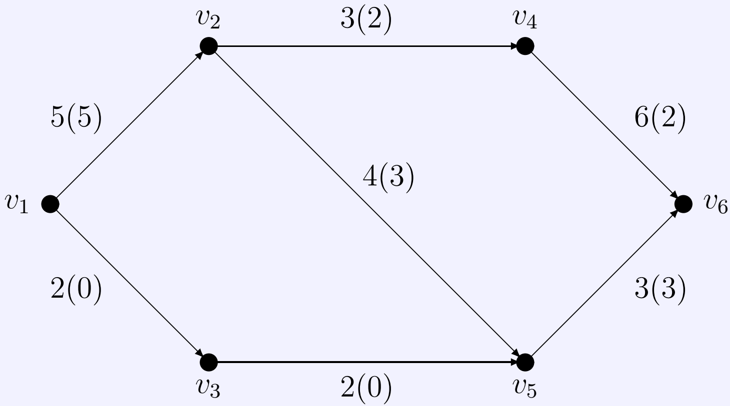
Przykład

Na łukach są podane przepustowości.

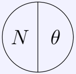


Nasycamy drogi

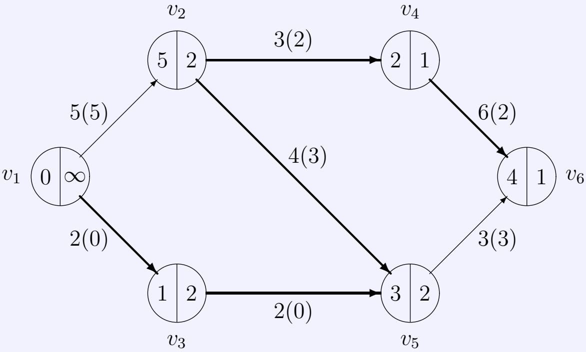
Wykonujemy algorytm dla kolejnych dróg (v_1, v_2, v_4, v_6), (v_1, v_2, v_5, v_6), (v_1, v_3, v_5, v_6).



Szukamy łańcuchów nienasyconych

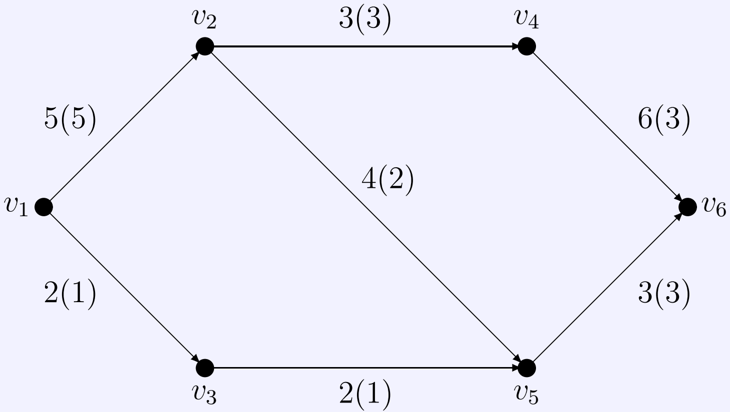


(WK-AŁH, MD, str. 161, alg. 8.3.2., notatki do wykładu).



$\theta = 1$.

Modyfikujemy przepływ



Artykuły (1)

Agata Surówka, *ALGORYTM FORDA – FULKERSONA I JEGO ZNACZENIE W ROZWIĄZYWANIU PROBLEMÓW TRANSPORTOWYCH:*

https://www.logistyka.net.pl/bank-wiedzy/item/download/76827_532c787cdf6a135626a0c306826d66cc

Artykuły (2)

Sezon nadchodzi!

Andrzej Marczuk, *LOGISTYCZNE ZARZĄDZANIE TRANSPORTEM TRUSKAWEK:*

https://www.up.lublin.pl/files/wydawnictwo-czasopisma/acta/technica_agraria/2002/2/acta_tech_1(2)_art_01.pdf

Ciekawe

Chmury są niezwykle ciekawe, ale nieobowiązkowe.

Tylko dla zainteresowanych!

Ciekawe

Chmury są niezwykle ciekawe, ale nieobowiązkowe.

Tylko dla zainteresowanych!

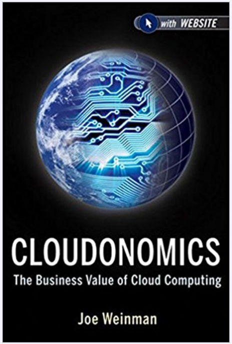
Zasada działania chmury obliczeniowej polega na przeniesieniu całego ciężaru świadczenia usług IT (danych, oprogramowania lub mocy obliczeniowej) na serwer i umożliwienie stałego dostępu poprzez komputery klienckie.

Źródło: Wikipedia

Joe Weinman

Joe Weinman. Axiomatic Cloud Theory. Working Paper, July 29, 2011.

Źródło:
http://www.asecib.ase.ro/cc/articole/Axiomatic%20Cloud%20Theory.pdf



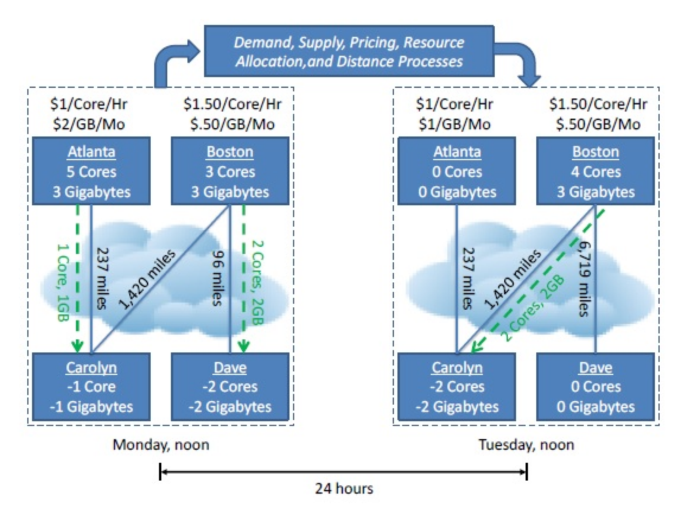
Definicja

Peter Mell and Tim Grance, "The NIST Definition of Cloud Computing," v. 15, NIST Special Publication 800-145 (Draft), January 2011.

U.S. National Institute of Standards and Technology

U.S. National Institute of Standards and Technology
Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Przykład


Demand, Supply, Pricing, Resource Allocation,and Distance Processes
$1/Core/Hr
$2/GB/Mo
$1.50/Core/Hr
$.50/GB/Mo
Atlanta
5 Cores
3 Gigabytes
Boston
3 Cores
3 Gigabytes
1 Core, 1GB
237 miles
1,420 miles
96 miles
2 Cores, 2GB
Carolyn
-1 Core
-1 Gigabytes
Dave
-2 Cores
-2 Gigabytes
$1/Core/Hr
$1/GB/Mo
$1.50/Core/Hr
$.50/GB/Mo
Atlanta
0 Cores
0 Gigabytes
Boston
4 Cores
3 Gigabytes
237 miles
1,420 miles
2 Cores, 2GB
6,719 miles
Carolyn
-2 Cores
-2 Gigabytes
Dave
0 Cores
0 Gigabytes
Monday, noon
Tuesday, noon
24 hours

Objaśnienia do przykładu

- Zbiór czterech węzłów lub wierzchołków: (*Atlanta*, *Boston*, *Carolyn*, *Dave*).
- Węzły są połączone w sieć przez trzy krawędzie: (A–C, B–C, B–D) wraz z przypisanymi im odległościami.
- Zasoby w poniedziałek w południe, A ma 5 rdzeni i 3 GB: $\langle 5, 3 \rangle$
- C potrzebuje 1 rdzeń i 1 GB: $\langle -1, -1 \rangle$. 1 rdzeń i 1 GB są alokowane dla C z A przez proces alokowania.
- Ceny: 1$ za rdzeń przez godzinę i 2$ za 1 GB przez miesiąc.
- Dobę później: C potrzebuje więcej, D – mniej. Ceny się zmieniają, zasoby też.

Chmura formalnie

Chmura jest strukturą

$$(\mathbb{S}, \mathbb{T}, G, Q, \delta, q_0)$$

gdzie:

- $\mathbb{S}$ – przestrzeń
- $\mathbb{T}$ – czas
- G – graf skierowany
- Q – zbiór stanów
- δ – funkcja przejścia
- q_0 – stan początkowy

Symbol $\mathbb{S}$

Przestrzeń metryczna $\mathbb{S} = (M, \lambda)$, gdzie

- M – zbiór lokacji,
- λ – odległość.

Symbol $\mathbb{T}$

Czas $\mathbb{T} = (T, \Sigma, \tau, <)$, gdzie

- T – zbiór chwil (zbiór punktów, odcinek),
- Σ – σ-algebra nad T (rodzina zbiorów),
- $\tau : \Sigma \to \mathbb{R}_0^+$ – miara na Σ,
- $<$ – porządek liniowy na T.

Symbol G

Graf zorientowany $G = (V, E)$ prosty, tzn. bez pętli i wielkrotnych krawędzi (łuków, gałęzi), gdzie

- V – zbiór wierzchołków,
- E – zbiór krawędzi, $e \subseteq V \times V$ oraz

$$(u, v) \in E \implies (v, u) \notin E.$$

Symbol Q

Zbiór stanów $Q = \{q_0, q_1, q_2, \dots\}$, gdzie $q_j = (R_{q_j}, A_{q_j}, L_{q_j}, P_{q_j})$ oraz

- $R_{q_j} : V \to \mathbb{R}^r$ – funkcja zasobów,
- $A_{q_j} : V \to \mathbb{R}^r$ – funkcja alokacji,
- $L_{q_j} : V \to M$ – funkcja lokacji,
- $P_{q_j} : V \to F$ – funkcja kosztów.

Symbol δ

Funkcja przejścia $\delta : T \rightarrow Q$.

Pojęcie zaczerpnięte z maszyny Turinga lub sieci Petri.

Sieci Petri:
http://jedrzej.ulasiewicz.staff.iiar.pwr.wroc.pl/ProgramowanieWspolbiezne/wyklad/Sieci-Petriego15.pdf

Symbol q_0

$q_0 \in Q$ – stan początkowy, gdzie $q_0 = \delta(t_0)$.