

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Wiktor Zuba

Nr albumu: 320501

Efektywne algorytmy generacji obiektów kombinatorycznych???

Praca magisterska
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
prof. Wojciech Rytter
Instytut Informatyki

??? 2017

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

Streszczenie

???

Słowa kluczowe

???

Dziedzina pracy (kody wg programu Socrates-Erasmus)

???

Klasyfikacja tematyczna

???

Tytuł pracy w języku angielskim

Effective algorithms of combinatorial objects generation???

Spis treści

Wprowadzenie	5
1. Własności hiperkostki	7
1.1. Podstawowe definicje	7
1.2. Podstawy kombinatoryczne	9
1.3. Własność ekspansji	9
2. Problemy na wadliwej hiperkostce	13
2.1. Graf z wadami	13
2.2. Spójność wadliwej hiperkostki	13
2.2.1. Podejście ekspansywne	13
2.2.2. Redukcja przy pomocy transformacji ścieżek	16
2.3. Długie ścieżki i cykle w grafie	21
2.3.1. Definicje	21
2.3.2. Długie ścieżki	22
2.3.3. Długie cykle	23
2.3.4. Długie pary ścieżek	23
2.4. Inne problemy	24
Bibliografia	25

Wprowadzenie

Rozdział 1

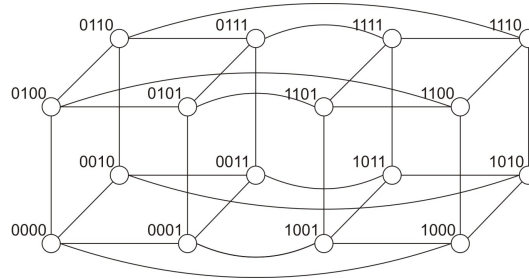
Własności hiperkostki

1.1. Podstawowe definicje

Definicja 1.1.1. Dla $n \in \mathbb{N}$ $[n] = \{0, \dots, n-1\}$ (zbiór pierwszych n liczb naturalnych).

Definicja 1.1.2. Hiperkostką wymiaru n (Q_n) nazwiemy graf, w którym każdy wierzchołek odpowiada ciągowi binarnemu długości n , zaś krawędzią połączone są te wierzchołki, których ciągi binarne różnią się na dokładnie jednej pozycji.

$$V(Q_n) = \{(v_0, \dots, v_{n-1}) : v_i \in \{0, 1\}\}, E(Q_n) = \{(u, v) : \sum_i |u_i - v_i| = 1\}$$



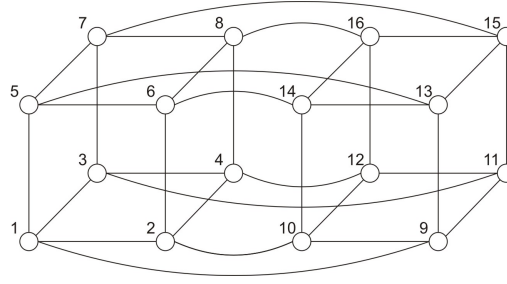
W przypadku pełnej hiperkostki bardzo łatwo jest określić długość najkrótszej ścieżki pomiędzy wierzchołkami – jest ona równa ilości pozycji na których różnią się ciągi tych wierzchołków.

Hiperkostka jest grafem dwudzielnym, w którym jedną składową jest zbiór wierzchołków o ciągach z parzystą liczbą jedynek, zaś drugą tych o ich nieparzystej liczbie.

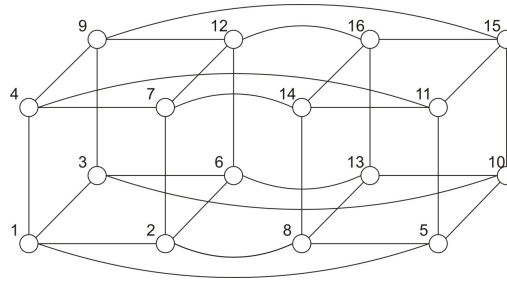
Cowięcej przy badaniu hiperkostek często dzieli się je na $n+1$ warstw, gdzie dla $i \in [n+1]$ i -tą warstwę stanowią te wierzchołki, których ciągi binarne mają dokładnie i jedynek (warstwa zawiera zatem wierzchołki oddalone o i od wierzchołka zerowego $(\bar{0})$).

Definicja 1.1.3. Dla dwóch wierzchołków hiperkostki definiujemy: $u\Delta v = \{i : u_i \neq v_i\}$, gdzie (u_0, \dots, u_{n-1}) i (v_0, \dots, v_{n-1}) to ciągi binarne wierzchołków u i v odpowiednio ($|u\Delta v|$ wyznacza odległość wierzchołków w hiperkostce).

Definicja 1.1.4. Numerowaniem klasycznym (naturalnym) hiperkostki nazwiemy takie numerowanie $\varphi : V(Q_n) \rightarrow \{1, \dots, |V(Q_n)|\}$ jej wierzchołków, że $\varphi(v) = 1 + \sum_i v_i \cdot 2^i$



Definicja 1.1.5. Numerowaniem warstwowym hiperkostki nazwiemy jej numerowanie w kolejności przeszukiwania grafu wszerz zaczynając od wierzchołka $\bar{0}$ z wybieraniem sąsiadów w kolejności leksykograficznej.



Uwaga 1. Jest to takie numerowanie $\varphi : V(Q_n) \rightarrow \{1, \dots, |V(Q_n)|\}$ jej wierzchołków, że wierzchołki z i -tej warstwy otrzymują numery od $\sum_{j=0}^{i-1} \binom{n}{j} + 1$ do $\sum_{j=0}^i \binom{n}{j}$. W obrębie jednej warstwy numery przyznawane są przeciwnie do kolejności leksykograficznej na odwróconych słowach. $\varphi(v) > \varphi(u) \Leftrightarrow (\sum_{i=0}^n v_i > \sum_{i=0}^n u_i) \vee ((\sum_{i=0}^n v_i = \sum_{i=0}^n u_i) \wedge (\sum_{i=0}^n 2^{n-i} v_i < \sum_{i=0}^n 2^{n-i} u_i))$

Dowód. Indukcyjnie po warstwach.

Dla warstwy 0 oczywiste.

Zakładając, że i -ta warstwa jest ponumerowana w tym porządku weźmy dwa wierzchołki u, v z warstwy $i + 1$: $u = (\bar{y}_1, 1, \bar{x}), v = (\bar{y}_2, 0, \bar{x})$.

Jeśli \bar{y}_1 zawiera same 0, to \bar{y}_2 zawiera dokładnie jedną 1, sąsiedzi tych wierzchołków z poprzedniej warstwy o najmniejszych numerach to odpowiednio $(\bar{0}, 0, \bar{x}), (\bar{0}, 0, \bar{x})$, tak więc zostaną ponumerowane jako sąsiedzi tego samego wierzchołka, jednak u otrzyma mniejszy numer jako sąsiad mniejszy leksykograficznie.

Jeśli \bar{y}_1 zawiera 1, to \bar{y}_2 też, więc sąsiedzi tych wierzchołków z poprzedniej warstwy o najmniejszych numerach to odpowiednio $(\bar{y}'_1, 1, \bar{x}), (\bar{y}'_2, 0, \bar{x})$, gdzie \bar{y}'_1 i \bar{y}'_2 , to odpowiednio \bar{y}_1 i \bar{y}_2 z pierwszymi 1 zamienionymi na 0. Z założenia indukcyjnego sąsiad u ma mniejszy numer niż sąsiad v , więc u ma mniejszy numer niż v . \square

Definicja 1.1.6. Dla grafu G oraz wierzchołka $v \in V(G)$ definiujemy sąsiedztwo wierzchołka jako zbiór wierzchołków połączonych z nim krawędzią: $N(v) = \{u \in V(G) : (u, v) \in E(G)\}$.

Definicja 1.1.7. Dla grafu G oraz zbioru wierzchołków $S \subseteq V(G)$ definiujemy sąsiedztwo zbioru wierzchołków jako zbiór tych sąsiadów wierzchołków ze zbioru, które same do tego zbioru nie należą: $N(S) = (\bigcup_{v \in S} N(v)) \setminus S$

Definicja 1.1.8. Dla grafu G oraz zbioru wierzchołków $S \subseteq V(G)$ definiujemy wnętrze zbioru wierzchołków jako zbiór tych wierzchołków z S , których wszyscy sąsiedzi również należą do tego zbioru: $In(S) = \{v \in S : N(v) \subseteq S\}$

Definicja 1.1.9. Dla danego $V \subseteq V(G)$ $G[V] = (V, \{uv \in E(G) : u, v \in V\})$ oznacza podgraf indukowany przez podzbiór wierzchołków V .

Definicja 1.1.10. Dla danego $V \subseteq V(G)$ $G - V = G[V(G) \setminus V]$ oznacza graf G z usuniętymi wierzchołkami V .

Definicja 1.1.11. Dla danego grafu G

$G^2 = (V(G), E(G) \cup \{uv : \exists w \in V(G) uw \in E(G) \cap vw \in E(G)\})$ oznacza kwadrat grafu, czyli graf z dodanymi krawędziami między wierzchołkami oddalonymi o co najwyżej 2.

1.2. Podstawy kombinatoryczne

$$\binom{2n}{n} = \frac{2^{2n} \Gamma(n + \frac{1}{2})}{\sqrt{\pi} \Gamma(n+1)}, \quad \binom{2n+1}{n} = \binom{2n+1}{n+1} = \frac{2^{2n+1} \Gamma(n + \frac{3}{2})}{\sqrt{\pi} \Gamma(n+2)}$$

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx \quad \text{dla } n \in \mathbb{N} \quad \Gamma(n) = (n-1)!,$$

$$\text{ogólniej dla } x \in \mathbb{R}, x > 1 \quad \frac{\Gamma(x+1)}{\Gamma(x)} = x, \quad \frac{\Gamma(x+\frac{1}{2})}{\Gamma(x)} < \frac{\Gamma(x+1)}{\Gamma(x+\frac{1}{2})} \Rightarrow \sqrt{x - \frac{1}{2}} < \frac{\Gamma(x+\frac{1}{2})}{\Gamma(x)} < \sqrt{x}$$

$$\text{Daje to ograniczenia: } \frac{2^{2n}}{\sqrt{\pi(n+\frac{1}{2})}} < \binom{2n}{n} < \frac{2^{2n}}{\sqrt{\pi n}}, \quad \frac{2^{2n+1}}{\sqrt{\pi(n+\frac{3}{2})}} < \binom{2n+1}{n} = \binom{2n+1}{n+1} < \frac{2^{2n+1}}{\sqrt{\pi(n+1)}}$$

$$\text{Lub równoważnie: } \frac{2^n}{\sqrt{\pi(\lfloor \frac{n}{2} \rfloor + \frac{1}{2})}} < \binom{n}{\lfloor \frac{n}{2} \rfloor} = \binom{n}{\lceil \frac{n}{2} \rceil} < \frac{2^n}{\sqrt{\pi \lceil \frac{n}{2} \rceil}}$$

Lemat 2. Dla $k \leq \lfloor \frac{n+1}{2} \rfloor$ zachodzi ograniczenie $\sum_{i=0}^{k-1} \binom{n}{i} \leq 2^{n-1} \frac{\binom{n}{k}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}}$

Dowód. (Uogólnienie dowodu z podobnego lematu dla $n = 2m, k < m$ z [1])

Założmy najpierw, że $k < \lfloor \frac{n}{2} \rfloor$

$$\text{Zdefiniujmy } c = \frac{\binom{n}{k}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} < 1, t = \lfloor \frac{n}{2} \rfloor - k, A = \sum_{i=0}^{k-1} \binom{n}{i}, B = \sum_{i=k}^{\lfloor \frac{n}{2} \rfloor - 1} \binom{n}{i}$$

$$\forall 1 \leq i \leq k \quad \frac{\binom{n}{k-i}}{\binom{n}{\lfloor \frac{n}{2} \rfloor - i}} < \frac{\binom{n}{k-i+1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor - i+1}} \Leftrightarrow \frac{k-c+1}{n-k+c} < \frac{\lfloor \frac{n}{2} \rfloor - c+1}{\lfloor \frac{n}{2} \rfloor + c},$$

$$\text{co wynika z szeregu prostych nierówności } \frac{k-c+1}{n-k+c} \leq \frac{k-c+1}{k+c+1} < \frac{\lfloor \frac{n}{2} \rfloor - c+1}{\lfloor \frac{n}{2} \rfloor + c+1} \leq \frac{\lfloor \frac{n}{2} \rfloor - c+1}{\lfloor \frac{n}{2} \rfloor + c}$$

$$\text{Daje nam to ograniczenia } \forall 1 \leq i \leq k \quad \frac{\binom{n}{k-i}}{\binom{n}{\lfloor \frac{n}{2} \rfloor - i}} < c.$$

Suma ostatnich t wyrazów szeregu A jest majoryzowana przez $c \cdot B$, wcześniejszych t przez c razy suma ostatnich t (a więc przez $c^2 \cdot B$). Daje nam to oszacowanie $A < (c + c^2 + c^3 + \dots + c^{\lfloor \frac{k}{t} \rfloor}) \cdot B < (c + c^2 + c^3 + \dots) \cdot B = \frac{c}{1-c} \cdot B$. Jednocześnie $A + B = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - 1} \binom{n}{i} < 2^{n-1}$.

$$A = c \cdot A + (1-c) \cdot A = c(A + \frac{1-c}{c} A) < c \cdot (A + B) < c \cdot 2^{n-1}.$$

Pozostaje udowodnić przypadki większych k :

$$\text{Dla } n = 2m, k = m \quad \sum_{i=0}^{m-1} \binom{2m}{i} = 2^{2m-1} - \frac{1}{2} \binom{2m}{m} < 2^{2m-1} = 2^{n-1} \cdot \frac{\binom{n}{k}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}}$$

$$\text{Dla } n = 2m+1, k = m \quad \sum_{i=0}^{m-1} \binom{2m+1}{i} = 2^{2m} - \binom{2m+1}{m} < 2^{2m} = 2^{n-1} \cdot \frac{\binom{n}{k}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}}$$

$$\text{Dla } n = 2m+1, k = m+1 \quad \sum_{i=0}^m \binom{2m+1}{i} = 2^{2m} = 2^{n-1} \cdot \frac{\binom{n}{k}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} \quad (\text{jedyna nieostra nierówność})$$

□

1.3. Własność ekspansji

Definicja 1.3.1. Graf G posiada własność ε -ekspansji wierzchołkowej, jeżeli dla każdego zbioru wierzchołków $S \subseteq V(G)$ takiego, że $|S| \leq \frac{|V(G)|}{2}$ zachodzi $|N(S)| \geq \varepsilon \cdot |S|$

Lemat 3. Zbiór pierwszych l wierzchołków hiperkostki według numerowania warstwowego posiada maksymalne wnętrze wśród zbiorów wielkości l .

Jest to jeden z lematów dowodzonych w pracy [2].

Lemat 4. Dla hiperkostki do udowodnienia własności ε_n -ekspansji wierzchołkowej wystarczy rozważyć zbiory S postaci $S_k, k \leq 2^{n-1}$.

Dowód. Weźmy dowolne $S \subseteq V(G), l = |S| + |N(S)|$ z Lematu 1.3 wynika, że $\frac{|N(S)|}{|S|} = \frac{|N(S)| + |S|}{|S|} - 1 \geq \frac{|S_l|}{|In(S_l)|} - 1 = \frac{|S_l \setminus In(S_l)|}{|In(S_l)|} \geq \frac{|N(In(S_l))|}{|In(S_l)|}$. Z definicji S_l wynika, że $In(S_l) = S_k$ dla $k = In(S_l)$.

Pozostaje udowodnić, że wystarczy rozważyć te S_k , że $k \leq 2^{n-1}$

Dla $l = |N(S)| + |S| \geq (\varepsilon_n + 1) \cdot 2^{n-1}$ mamy $|S| > 2^{n-1}$ lub $|N(S)| \geq \varepsilon_n |S|$, wystarczy więc rozważyć przypadek $l < (\varepsilon_n + 1) \cdot 2^{n-1}$.

Dla $n = 2m + 1$ weźmy $k = 2^{n-1} = \sum_{i=0}^m \binom{2m+1}{i}$, wtedy S_k = pełne $m + 1$ pierwszych warstw i $N(S_k)$ = warstwa $m + 1$. Przykład ten pokazuje, że $\varepsilon_n \leq \frac{\binom{2m+1}{m+1}}{2^{2m}}$, więc $l < 2^{2m} + \binom{2m+1}{m+1} \Rightarrow S_l$ mieści się w pierwszych $m + 2$ warstwach $\Rightarrow S_k = In(S_l)$ mieści się w pierwszych $m + 1$ warstwach $\Rightarrow k \leq 2^{2m} = 2^{n-1}$.

Dla $n = 2m$ weźmy $k = 2^{n-1} = \sum_{i=0}^{m-1} \binom{2m}{i} + \frac{1}{2} \binom{2m}{m}$, wtedy S_k = pełne m pierwszych warstw + połowa środkowej. W środkowej warstwie pierwsze $\binom{2m-1}{m-1} = \frac{1}{2} \binom{2m}{m}$ wierzchołków to dokładnie te, których ciągi binarne kończą się na 1. Wtedy też $S_k \cup N(S_k)$ to dokładnie pełne $m + 1$ pierwszych warstw plus te wierzchołki z warstwy $m + 2$, które kończą się na 1 $\Rightarrow |N(S_k)| = \binom{2m-1}{m} + \binom{2m-1}{m-1} = \binom{2m-1}{m} + \binom{2m-1}{m-1} = \binom{2m}{m}$. Przykład ten pokazuje, że $\varepsilon_n \leq \frac{\binom{2m}{m}}{2^{2m-1}}$, więc $l < 2^{2m-1} + \binom{2m}{m} \Rightarrow S_l$ mieści się w pierwszych $m + 1$ warstwach plus tych wierzchołkach z warstwy $m + 2$, które kończą się na 1 $\Rightarrow k \leq 2^{n-1}$. \square

Wniosek 5. Hiperkostka wymiaru n nie posiada własności $\frac{2\sqrt{2}}{\sqrt{\pi n}}$ -ekspansji wierzchołkowej.

Dowód. Dla $n = 2m + 1$

$$\frac{|N(S_{2^{2m}})|}{|S_{2^{2m}}|} = \frac{\binom{2m+1}{m+1}}{2^{2m}} = \frac{2^{2m+1}}{2^{2m} \sqrt{\pi(m+1)}} = \frac{2}{\sqrt{\pi(m+1)}} = \frac{2}{\sqrt{\pi(\frac{n}{2} + \frac{1}{2})}} = \frac{2\sqrt{2}}{\sqrt{\pi(n+1)}} < \frac{2\sqrt{2}}{\sqrt{\pi n}}.$$

$$\text{Dla } n = 2m \quad \frac{|N(S_{2^{2m-1}})|}{|S_{2^{2m-1}}|} = \frac{\binom{2m}{m}}{2^{2m-1}} < \frac{2^{2m}}{2^{2m-1} \sqrt{\pi(m+1)}} = \frac{2}{\sqrt{\pi m}} = \frac{2}{\sqrt{\pi \cdot \frac{n}{2}}} = \frac{2\sqrt{2}}{\sqrt{\pi n}}. \quad \square$$

Twierdzenie 6. Hiperkostka Q_n posiada własność $\frac{1}{\sqrt{\pi n}}$ -ekspansji wierzchołkowej.

Dowód. Jeśli $k = \sum_{i=0}^r \binom{n}{i}$ (pełne $r + 1 \leq \lfloor \frac{n}{2} \rfloor + 1$ warstw), to $\frac{|N(S_k)|}{|S_k|} = \frac{\binom{n}{r+1}}{\sum_{i=0}^r \binom{n}{i}} > \frac{\binom{n}{r+1} \binom{\lfloor \frac{n}{2} \rfloor}{r+1}}{2^{n-1} \binom{\lfloor \frac{n}{2} \rfloor}{r+1}} = \frac{\binom{\lfloor \frac{n}{2} \rfloor}{r+1}}{2^{n-1}} > \frac{2^n}{2^{n-1} \sqrt{\pi(\lceil \frac{n}{2} \rceil + \frac{1}{2})}} = \frac{2}{\sqrt{\pi(\lceil \frac{n}{2} \rceil + \frac{1}{2})}} \geq \frac{2\sqrt{2}}{\sqrt{\pi(n + \frac{3}{2})}} \geq \frac{2}{\sqrt{\pi n}}$ (dla $n \geq 2$).

($n = 2m + 1, r = m$ rozważone w 4)

Jeśli $k = \sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}$ (pełne $r + 1 \leq \lfloor \frac{n}{2} \rfloor + 1$ warstw plus te wierzchołki z warstwy $r + 2$, których ciągi binarne kończą się na 1).

$$|N(S_k) \cup S_k| = \sum_{i=0}^{r+1} \binom{n}{i} + \binom{n-1}{r} \Rightarrow |N(S_k)| = 2 \cdot \binom{n-1}{r+1}$$

$$\begin{aligned} \frac{|N(S_k)|}{|S_k|} &= \frac{2 \cdot \binom{n-1}{r+1}}{\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}} > \frac{2 \cdot \binom{n-1}{r+1} \binom{\lfloor \frac{n}{2} \rfloor}{r+1}}{2^{n-1} \cdot (\binom{n}{r+1} + \binom{n-1}{r}) \cdot \binom{\lfloor \frac{n}{2} \rfloor}{r+1}} = \frac{2 \cdot \binom{n-1}{r+1} \binom{\lfloor \frac{n}{2} \rfloor}{r+1}}{2^{n-1} \cdot ((\binom{n-1}{r} + \binom{n-1}{r+1}) + \binom{n-1}{r}) \cdot \binom{\lfloor \frac{n}{2} \rfloor}{r+1}} > \\ &= \frac{2 \cdot \binom{n-1}{r+1} \binom{\lfloor \frac{n}{2} \rfloor}{r+1}}{2^{n-1} \cdot \binom{n-1}{r+1} \cdot \binom{\lfloor \frac{n}{2} \rfloor}{r+1}} = \left(\frac{2^{n-1}}{\binom{\lfloor \frac{n}{2} \rfloor}{r+1}} + \frac{\binom{n-1}{r+1}}{2 \cdot \binom{n-1}{r+1}} \right)^{-1} > \left(\frac{\sqrt{\pi(n + \frac{3}{2})}}{2\sqrt{2}} + \frac{1}{2} \right)^{-1} = \frac{2\sqrt{2}}{\sqrt{\pi(n + \frac{3}{2})} + \sqrt{2}} > \frac{2}{\sqrt{\pi n}} \quad (\text{dla } n \geq 7). \end{aligned}$$

W pozostałych przypadkach można otrzymać ograniczenie choć dużo gorsze wiedząc, że dodanie wierzchołka do S zmniejszy $N(S)$ o co najwyżej 1.

Weźmy teraz $\sum_{i=0}^r \binom{n}{i} < k < \sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}$

$$\frac{|N(S_k)|}{|S_k|} > \frac{\binom{n}{r+1} - \binom{n-1}{r}}{\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}} = \frac{\binom{n-1}{r+1}}{\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}} \geq \frac{\frac{1}{2} \binom{n}{r+1}}{\sum_{i=0}^r \binom{n}{i} + \frac{1}{2} \binom{n}{r+1}} \left(\frac{\sqrt{\pi(n+\frac{3}{2})}}{\sqrt{2}} + 1 \right)^{-1} > \frac{1}{\sqrt{\pi n}} \quad (\text{dla } n \geq 7).$$

Analogicznie dla $\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r} < k < \sum_{i=0}^{r+1} \binom{n}{i}$

$$\frac{|N(S_k)|}{|S_k|} > \frac{2\binom{n-1}{r+1} - \binom{n-1}{r}}{\sum_{i=0}^{r+1} \binom{n}{i}} = \frac{\binom{n-1}{r+1}}{\sum_{i=0}^{r+1} \binom{n}{i}} > \frac{\binom{n-1}{r+1}}{2^{n-1} \frac{\binom{n}{r+1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} + \binom{n}{r+1}} \geq \frac{\binom{n}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{(2^{n-1} + \binom{n}{\lfloor \frac{n}{2} \rfloor}) \binom{n}{r+1}} = \frac{\binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1} + \binom{n}{\lfloor \frac{n}{2} \rfloor}} >$$

$$\left(\frac{\sqrt{\pi(n+\frac{3}{2})}}{\sqrt{2}} + 1 \right)^{-1} > \frac{1}{\sqrt{\pi n}} \quad (\text{dla } n \geq 7).$$

Dla przypadków $n \leq 6$ można ręcznie sprawdzić wszystkie 2^{n-1} przypadków, aby również otrzymać oszacowanie $\frac{1}{\sqrt{\pi n}}$. \square

Rozdział 2

Problemy na wadliwej hiperkostce

2.1. Graf z wadami

Definicja 2.1.1. W grafie G możemy wyróżnić niektóre wierzchołki (czasem również krawędzie) i oznaczyć jako wadliwe. Graf z niepustym takim wyróżnionym zbiorem wierzchołków wadliwych $F \subseteq V(G)$ nazywamy grafem z wadami (lub grafem wadliwym)

Wadliwe wierzchołki (i/lub krawędzie) najczęściej traktowane są jako usunięte z grafu – mówimy w tym przypadku o grafie $G - F$. Wyróżnianie wadliwych wierzchołków w grafie zamiast definiowania nowego grafu jest umotywowane głównie w przypadkach, gdy pełny graf łatwo zdefiniować i zapisać w pamięci małej względem jego rozmiaru (np. klika, hiperkostka, graf de Bruijna), a zbiór wadliwych wierzchołków jest również mały.

2.2. Spójność wadliwej hiperkostki

Ten podrozdział jest napisany w większości na podstawie [3].

Uwaga 7. Aby zbadać spójność grafu $G - F$ dla spójnego grafu G wystarczy sprawdzić czy wciąż istnieje ścieżka pomiędzy dowolnymi dwoma wierzchołkami, które oryginalnym grafie sąsiadowały z jakimś spośród usuniętych wierzchołków (wszystkie takie wierzchołki należą do jednej spójnej składowej).

Dowód. Aby udowodnić spójność trzeba pokazać, że istnieje ścieżka pomiędzy dowolnymi dwoma wierzchołkami, jednak skoro w oryginalnym grafie taka ścieżka istniała, to w nowym grafie jedyną przeszkodą jest to, że na tej ścieżce mogły występować wierzchołki, które zostały usunięte. Taką ścieżkę można naprawić wstawiając w miejsca od pierwszego do ostatniego wystąpienia wierzchołka usuniętego ścieżkę pomiędzy odpowiednimi ich sąsiadami istniejącą w pomniejszonym grafie. \square

2.2.1. Podejście ekspansywne

Twierdzenie 8. Niech graf G posiada własność ε -ekspansji wierzchołkowej z $\varepsilon > 0$ i maksymalny stopień wierzchołka Δ , oraz dana jest wyrocznia zwracająca dla danego wierzchołka listę jego sąsiadów. Wtedy istnieje algorytm, który otrzymuje na wejściu zbiór $F \subseteq V(G)$ oraz ε i testuje spójność $G - F$ w czasie $O\left(\frac{|F|^2 \cdot \Delta^2 \cdot \log(|V(G)|)}{\varepsilon}\right)$

Lemat 9. Spójna składowa $S \subseteq V(G) \setminus F$ grafu $G - F$ jest jednego z dwóch typów:

- *główna* – $|S| > \frac{|V(G)|}{2}$
- *mała* – $|S| \leq \frac{|F|}{\varepsilon}$

Uwaga 10. *Co prawda dla dużego $|F|$ i małego ε może być tak, że składowa jest jednocześnie główna i mała, jednak po pierwsze jest to przypadek mało interesujący, gdyż wtedy zwykle przeszukiwanie grafu spełnia tezę twierdzenia, a po drugie przypadek ten nie psuje w żaden sposób otrzymywanego algorytmu. W lemacie istotne jest to, że w grafie nie ma składowych średnich wielkości.*

Fakt 11. *Może być tylko jedna składowa główna.*

Dowód. (Lematu)

Weźmy spójną składową S grafu $G - F$ ($N_{G-F}(S) = 0$), jeżeli $S \leq \frac{|V(G)|}{2}$, to z własności ε -ekspansji wierzchołkowej grafu G $|N_G(S)| \geq \varepsilon \cdot |S|$ (gdzie S jest teraz traktowane jako podzbiór wierzchołków grafu G). Gdyby zachodziło $|S| > \frac{|F|}{\varepsilon}$, to mielibyśmy $|N_G(S)| > \frac{\varepsilon \cdot |F|}{\varepsilon} = |F|$, co daje sprzeczność ponieważ aby w grafie $G - F$ to sąsiedztwo było puste z grafu G trzeba usunąć co najmniej $N_G(S)$ wierzchołków. \square

Dowód. (Twierdzenia)

Chcemy sprawdzić, czy wszyscy sąsiedzi wierzchołków usuniętych należą do tej samej spójnej składowej. Na podstawie lematu 9, jeśli składowa zawierająca taki wierzchołek jest większa niż $\frac{|F|}{\varepsilon}$, to jest to składowa główna. Jeżeli wszystkie takie wierzchołki spełniają ten warunek, to $G - F$ jest spójny. Jeżeli natomiast, któraś z tych składowych okaże się mała, to $G - F$ nie jest spójny.

Wystarczy więc uruchomić liniowe przeszukiwanie grafowe w każdym wierzchołku sąsiadującym z wierzchołkiem wadliwym i przerywać po przejrzaniu $\frac{|F|}{\varepsilon}$ wierzchołków.

Algorytm liniowego przeszukiwania grafowego uruchamiany jest co najwyżej $|F| \cdot \Delta$ razy. Za każdym razem przeglądamy co najwyżej $\frac{|F|}{\varepsilon}$ wierzchołków. Dla każdego przeglądanego wierzchołka sprawdzamy co najwyżej Δ sąsiadów, a odpowiedź wyroczni zajmuje $O(\log(|V(G)|))$ czasu. Daje to złożoność z tezy twierdzenia. \square

Wniosek 12. *Ponieważ zgodnie z twierdzeniem 6 hiperkostka Q_n posiada własność $\frac{1}{\sqrt{\pi n}}$ -ekspansji wierzchołkowej, oraz można znaleźć wszystkich sąsiadów wierzchołka w czasie liniowym od ich ilości powyższy algorytm testuje spójność wadliwej hiperkostki w czasie $O(|F|^2 \cdot n^{3.5})$ (wyrażonego w ilości operacji arytmetycznych).*

Uwaga 13. *Ze względu na długość zapisu identyfikatora wierzchołka liniową od wymiaru hiperkostki nie da się przeprowadzać operacji na wierzchołkach w czasie szybszym niż n . To dolne ograniczenie jest osiągalne przy przechowywaniu przejrzanych wierzchołków w hashmapie (czas oczekiwany operacji $O(n)$, złożoność pamięciowa całej struktury $O(n^{0.5}|F|)$), lub w drzewie prefiksowym (czas pesymistyczny operacji $O(n)$, złożoność pamięciowa całej struktury $O(n^{1.5}|F|)$). Pozwala to w łatwy sposób uzyskać efektywną wyrocznię, a więc i algorytm o złożoności z wniosku.*

pseudokod i uwagi

W algorytmie wykorzystywana jest struktura T z operacjami

- $Insert(v, T)$ wstawiającą wierzchołek v do struktury T

- *Retrieve*(v, T) zwracająca binarną informację o obecności wierzchołka v w strukturze T

które wymagają $O(n)$ czasu na wykonanie (jak w uwadze 13).

Przeszukiwanie grafowe odbywa się przy pomocy funkcji o pseudokodzie:

```
DFS(v){
    counter ++;
    Insert(v, T);
    if(counter ≥ size) return(TRUE);
    foreach(u ∈ N(v)){
        if(Retrieve(u, T) == FALSE){
            if(DFS(u)) return(TRUE);
        }
    }
    return(FALSE);
}
```

Spójność sprawdzana jest przy pomocy funkcji głównej o pseudokodzie:

```
Connectivity(n, F){
    T2 = empty_structureT();
    counter = 0;
    foreach(f ∈ F){
        Insert(f, T2);
        counter ++;
    }
    size = sqrt(π * n) * counter;
    foreach(f ∈ F){
        foreach(v ∈ N(f)){
            if(Retrieve(v, T2) == FALSE){
                counter = 0;
                T = T2;
                if(DFS(v) == FALSE) return(FALSE);
            }
        }
    }
    return(TRUE);
}
```

Uwaga 14. Aby przeiterować po $N(v)$ wystarczy przeiterować się po współrzędnych uzyskując sąsiada poprzez zanegowanie tej współrzędnej w zapisie binarnym v .

Uwaga 15. Można użyć dodatkowej struktury T w której przechowywane są wszystkie wierzchołki z poprzednich wywołań $DFS(v)$ z funkcji głównej. Wtedy przy kolejnych użyciach $DFS(v)$ można sprawdzać, czy wierzchołek nie był już wcześniej w jakiejś składowej (można wtedy od razu zwrócić $TRUE$). Teoretycznie może to zwiększyć złożoność dwukrotnie, jednak w praktyce będzie to dużo szybsze (już nawet z tego względu, że albo inni sąsiedzi tego samego f są oddaleni o 2, albo na drodze staje inny wierzchołek z F), w szczególności przy użyciu bardziej wyszukanych kolejności przeszukiwania (np. próba dojścia do wierzchołka $\bar{0}$).

2.2.2. Redukcja przy pomocy transformacji ścieżek

Algorytm przedstawiony w poprzednim podrozdziale jest dowodem na to, że testowanie spójności wadliwej hiperkostki może być zrobione wielomianowo ze względu na ilość wad i wymiar hiperkostki. Algorytm ten wykorzystuje jednak bardzo płytko potencjał tak regularnego grafu. W tym podrozdziale przedstawię algorytm, który dzięki głębszemu wykorzystaniu własności hiperkostki otrzymuje lepsze rezultaty złożonościowe.

Definicja 2.2.1. Na potrzeby tego podrozdziału definiuję ze pracą [3] dla $F \subseteq V(Q_n)$ podgraf $G(F) = (A \cup B \cup F, E)$ grafu Q_n , gdzie $A = N(F)$, $B = N(A) \setminus F$, $E = \{uv \in E(Q_n) : u \in A \cup F\}$ (podgraf zawierający wierzchołki w odległości ≤ 2 od wierzchołków wadliwych, plus krawędzie w których jeden z końców jest wadliwy lub z takim sąsiaduje).

Twierdzenie 16. Dla $F \subseteq V(Q_n)$ graf $Q_n - F$ jest spójny wtedy i tylko wtedy gdy dla każdej C – spójnej składowej $Q_n^2[F]$ spójny jest graf $G(C) - C$.

Transformacje ścieżek w hiperkostce

Definicja 2.2.2. Dla ścieżki $W = (v_0, v_1, \dots, v_n)$ (z możliwymi powtórzeniami) w hiperkostce sekwencją tranzycji nazywamy ciąg $\tau = (d_1, d_2, \dots, d_n)$, gdzie d_i jest współrzędną na której różni się ciągi binarne wierzchołków v_{i-1} i v_i .

Fakt 17. τ jest sekwencją tranzycji pewnej uv -ścieżki w Q_n wtedy i tylko gdy $u\Delta v = \{i \in [n] : \#(\tau, i) \text{ nieparzyste}\}$, gdzie $\#(\tau, i)$ to ilość wystąpień i w sekwencji τ .

Dla τ – sekwencji tranzycji uv -ścieżki W definiujemy trzy operacje:

- $\text{swap}(\tau_1, i, j, \tau_2) = (\tau_1, j, i, \tau_2)$ dla $\tau = (\tau_1, i, j, \tau_2)$
- $\text{insert}_i(\tau_1, \tau_2) = (\tau_1, i, i, \tau_2)$ dla $\tau = (\tau_1, \tau_2), i \in [n]$
- $\text{delete}(\tau_1, i, i, \tau_2) = (\tau_1, \tau_2)$ dla $\tau = (\tau_1, i, i, \tau_2)$

Wszystkie te operacje nie zmieniają parzystości wystąpień współrzędnych, dlatego też dowolnie w ten sposób zmodyfikowana sekwencja wciąż jest sekwencją tranzycji pewnej uv -ścieżki w Q_n .

Definicja 2.2.3. Dwie ścieżki, których sekwencje tranzycji τ, ρ spełniają $\forall_{i \in [n]} \#(\tau, i) = \#(\rho, i)$ nazywamy równoważnymi.

Uwaga 18. Dla dowolnych dwóch uv -ścieżek w Q_n istnieje sekwencja operacji $\text{swap}, \text{insert}, \text{delete}$ (w tej kolejności bez przepłatów), która przemienia sekwencję tranzycji pierwszej w sekwencję tranzycji drugiej.

Dowód. Jeśli dwie ścieżki są równoważne, to można jedną przekształcić w drugą przy pomocy samych operacji swap .

W przypadku gdy sekwencje mają różne liczności wystąpień współrzędnych, to można je doprowadzić do takich τ', ρ' , że $\forall_{i \in [n]} \#(\tau', i) = \#(\rho', i)$ przy pomocy samych operacji insert (używanych w dowolnie wybranych wierzchołkach). \square

Definicja 2.2.4. Na potrzeby dowodu twierdzenia 16 dla uv -ścieżki $W = (w_0, w_1, \dots, w_k)$ (gdzie $w_0 = u, w_k = v$) wierzchołek w_i nazywamy portem, jeśli nie jest wierzchołkiem wadliwym, ale dokładnie jeden z jego sąsiadów w ścieżce należy do F (port musi więc należeć do A).

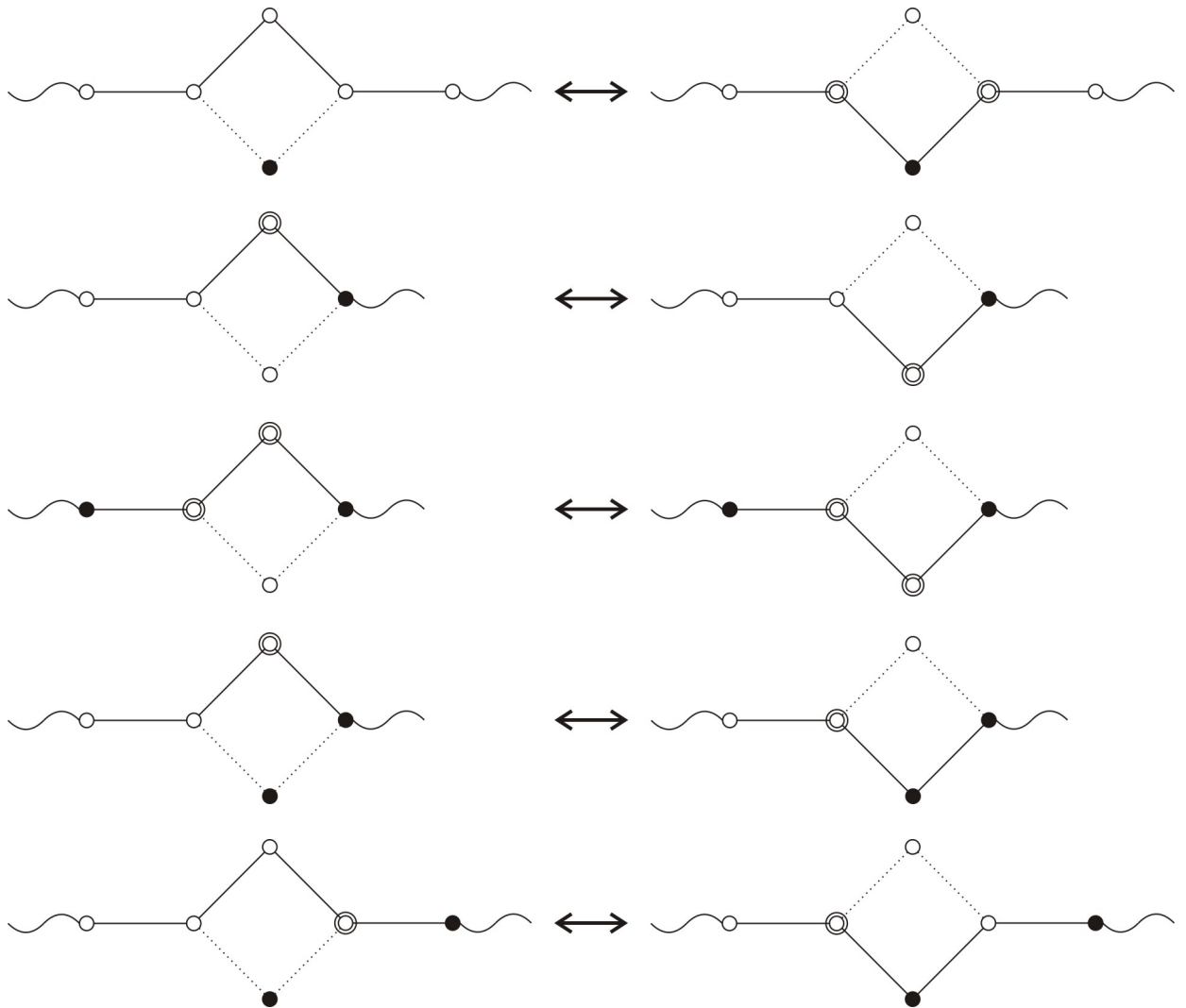
W przypadku tej definicji należy rozróżnić przeplatające się pojęcia wierzchołka grafu i jego wystąpienia na ścieżce – portem nazywane jest konkretne wystąpienie na ścieżce, inne jego wystąpienia nie muszą być portami.

Dla C spójnej składowej $G(F) - F$ przez $p(C, W)$ oznaczamy ilość portów w części W należącej do C .

Lemat 19. *Operacja swap zachowuje parzystość $p(C, W)$.*

Dowód. Dowód stanowi rysunkowe rozpatrzenie wszystkich możliwych przypadków w których w wyniku operacji *swap* powstaje i/lub znika pewien port (przypadki przy końcach ścieżki można "dopełnić" zwykłymi wierzchołkami do przypadków ze środka ponieważ wierzchołki końcowe nie są wadliwe).

○ – wierzchołek zwykły ● – wierzchołek wadliwy ◎ – port
 ——— – krawędź ścieżki – krawędź spoza ścieżki ~~~~~ – nieistotna reszta ścieżki



Dowód. Dla dowolnie wybranych dwóch wierzchołków $u, v \in V(Q_n - F)$ weźmy W – ścieżkę między nimi w pełnym Q_n . Jeśli W nie zawiera wadliwego wierzchołka, to jest poprawną ścieżką w $Q_n - F$. W przeciwnym przypadku znajdujemy na tej ścieżce pierwsze wystąpienie wierzchołka wadliwego. Poprzedni wierzchołek na ścieżce oraz pierwszy kolejny z poza zbioru F są dwoma niewadliwymi wierzchołkami należącymi do $G(C)$, gdzie C jest spójną składową $Q_n^2[F]$ (oddalone o 1 od wadliwych wierzchołków, które są połączone ścieżką samych wadliwych wierzchołków). W $G(C)$ nie ma wadliwych wierzchołków spoza C , ponieważ oznaczałoby to, że taki wierzchołek jest oddalony o ≤ 2 od pewnego wierzchołka z C , a więc byłby z nim połączony w Q_n^2 , dlatego też ścieżka ze spójnego z założenia $G(C) - C$ nie zawiera wadliwego wierzchołka. Wystarczy więc wadliwą część ścieżki W zastąpić odpowiednią ścieżką z $G(C) - C$ aby otrzymać poprawną ścieżkę w $Q_n - F$. \square

Dowód. (Twierdzenia 16)

Lemat 22 jest implikacją w jedną stronę. W drugą stronę dla spójnego $Q_n^2[F]$ jest dana wnioskiem 21. Wystarczy udowodnić, że nic nie psuje się w przypadku gdy $Q_n^2[F]$ ma więcej niż jedną spójną składową. Dla C – spójnej składowej $Q_n^2[F]$ jeśli $Q_n - F$ jest spójne, to jest także również $Q_n - C$ (dla wierzchołków spoza $Q_n - F$ te same ścieżki są dobre, dla tych z $F \setminus C$ dowolny sąsiad należy do $Q_n - F$, więc również łatwo zbudować ścieżkę), a więc spójne jest również $G(C) - C$ co kończy dowód. \square

Algorytm

Stosując powyższe twierdzenie można uzyskać wielomianowy algorytm używając jedynie przeszukiwania grafowego podobnie jak w podrozdziale 2.2.1. Można jednak uzyskać lepsze rezultaty używając dodatkowo struktury *Find-Union* i sprawdzając spójności już w trakcie budowania podgrafów $G(C) - C$.

W algorytmie używane są:

- struktura *Find-Union* D z operacjami:
 - $Make(v, D)$ tworzącą singleton $\{v\}$
 - $Find(v, D)$ zwracającą wskaźnik na zbiór zawierający v
 - $Union(u, v, D)$ łączącą zbiór zawierający u ze zbiorem zawierającym v

których zamortyzowany czas można ograniczyć przez $O(\log m)$ (a da się nawet uzyskać $O(\log^* m)$), gdzie m jest ilością użytych operacji $Make(v, D)$. Dodatkowo struktura zapewnia możliwość sprawdzenia, czy zawiera więcej niż jeden zbiór (wystarczy pojedynczy licznik inkrementowany przy $Make(v, D)$ i dekrementowany przy $Union(u, v, D)$).

- strukturę T do przechowywania informacji o niektórych wierzchołkach jak binarne drzewo prefiksowe lub hashmapa, przechowywującą dla wierzchołka v_T informacje:
 - wskaźnik do wierzchołka v w strukturze D
 - informacje o wadliwości/braku wadliwości wierzchołka
 - binarną informację o tym czy wierzchołek był odwiedzony i należy do $F \cup N(F)$

wspierającą operacje:

- $Insert(v, T)$ wstawiającą wierzchołek v do struktury T i zwracającą wskaźnik na v_T
- $Retrieve(v, T)$ zwracającą v_T lub $NULL$ w przypadku gdy v nie ma w strukturze

które wymagają $O(n)$ czasu na wykonanie.

Definiuję pomocniczą funkcję uzyskiwania wierzchołków ze struktury T i inicjalizowania w razie nieobecności:

```
Retrieve2(v, T){
  v_T = Retrieve(v, T);
  if(v_T == NULL){
    v_T = Insert(v, T);
    v_T.healthy = TRUE;
    v_T.visited = FALSE;
    Make(v, D);
  }
  return(v_T);
}
```

Najistotniejszą częścią algorytmu jest procedura (czasami dla wielu wierzchołków z F) $DFS(f)$ znajdująca spójne składowe $G(C) - C$, dla C spójnej składowej $Q_n^2[F]$ zawierającej wadliwy wierzchołek f , o następującym pseudokodzie:

```
DFS(f){
  foreach(u ∈ N(f)){
    u_T = Retrieve2(u, T);
    if(u_T.visited == FALSE){
      u_T.visited = TRUE;
      if(u_T.healthy){
        foreach(v ∈ N(u)){
          v_T = Retrieve2(v, T);
          if(v_T.healthy){
            if(Find(u, D) ≠ Find(v, D))  \\ krawędź uv należy do G(C) - C
          }else if(v_T.visited == FALSE){
            v_T.visited = TRUE;
            DFS(v);  \\ wadliwy wierzchołek należący do C
          }
        }
      }
    }else DFS(u);  \\ wadliwy wierzchołek należący do C
  }
}
```

Powyższa prodedura uruchamiana jest z funkcji głównej:

```
Conectivity(n, F){
  T = empty_structureT();
  foreach(f ∈ F){
    f_T = Insert(f, T);
    f_T.healthy = FALSE;
    f_T.visited = FALSE;
  }
}
```

```

foreach( $f \in F$ ){
   $f_T = \text{Retrieve}(f, T)$ ;
  if( $f_T.visited == FALSE$ ){
     $f_T.visited = TRUE$ ;
     $D = \text{empty\_structure}D()$ ;
     $DFS(f)$ ;
    if( $D.counter > 1$ ) return( $FALSE$ );
  }
}
return( $TRUE$ );
}

```

Analiza złożoności

Wniosek 23. Algorytm ma pesymistyczną złożoność czasową i pamięciową $O(|F| \cdot n^3)$.

Dowód. Dla każdego wierzchołka z F każdy sąsiad jest przeglądany po jeden raz, dla każdego wierzchołka należącego do $N(F)$ również przeglądani są wszyscy sąsiedzi po razie. Przeglądnięcie jednego wierzchołka (znalezienie odpowiedniego wierzchołka w T i D) zajmuje $O(n)$, ustawienie właściwości w D zajmuje stały czas po posiadaniu dowiązania do odpowiedniego wierzchołka – daje to złożoność tej części $O(|F| \cdot n^3)$.

Operacja $Make(v, D)$ używana jest dla każdego wierzchołka z $G(C) - C$ po razie dla każdego C (może być użyta więcej niż raz dla wierzchołków oddalonych o 2 od F i występujących w różnych $G(C)$). W przypadku $Find(v, D)$ i $Union(u, v, D)$ uruchamiane są one maksymalnie odpowiednio dwa i jeden raz dla każdego z sąsiadów wierzchołków $N(F)$ – daje to złożoność $O(|F| \cdot n^2 \log(n))$.

Preprocessing i Postprocessing (tworzenie i usuwanie struktur T i D) może być zrobione w czasie liniowym od ich wielkości (w przypadku D i hashmapy można trzymać dodatkowo nieuporządkowaną listę dowiązań do wszystkich elementów). W przypadku struktury T wielkość tą można ograniczyć przez $O(|F| \cdot n^3)$ przy użyciu drzewa prefiksowego (lub $O(|F| \cdot n^2)$ przy użyciu hashmapy, która nie pozwala jednak uzyskać odpowiedniej złożoności przy pesymistycznym scenariuszu), zaś w przypadku struktur D łącznie $O(|F| \cdot n^2)$. \square

2.3. Długie ścieżki i cykle w grafie

W poprzednim podrozdziale przedstawiony był przykład problemu na wadliwej hiperkostce, dla którego można było znaleźć rozwiązanie wielomianowe od n i $|F|$. W tym rozdziale przedstawimy kilka problemów, dla których samo przedstawienie wyników wymagało by wykładniczej pamięci, jednak samo rozstrzygnięcie czy rozwiązanie istnieje (sprawdzenie warunków twierdzenia) jest możliwe w czasie $O(|F| \cdot n)$ dla odpowiednio małych $|F|$ (wartości podane w sformułowaniach twierdzeń). W przypadku podwójnych ścieżek twierdzenie daje jedynie warunek wystarczający, dlatego algorytm otrzymany dzięki niemu nawet dla tych małych $|F|$ potrafi jedynie rozstrzygnąć pomiędzy "istnieją długie ścieżki" i "kryterium nie rozstrzyga".

2.3.1. Definicje

Definicja 2.3.1. Wolną od wad (nieprzechodzącą przez wierzchołki wadliwe) ścieżkę bez powtórzeń (drogę) w hiperkostce Q_n z wadami ze zbioru $F \subseteq V(Q_n)$ nazwiemy długą, jeśli ma długość co najmniej $2^n - 2|F| - 2$.

Definicja 2.3.2. Wolny od wad cykl bez powtórzeń w hiperkostce Q_n z wadami ze zbioru $F \subseteq V(Q_n)$ nazwiemy długim, jeśli ma długość co najmniej $2^n - 2|F|$.

Uwaga 24. Dla $F \cup \{u, v\}$ należącego do jednej dwudzielnej części Q_n nie da się skonstruować uv -ścieżki wolnej od wad o długości większej niż $2^n - 2|F| - 2$ (stąd długość w definicji).

Dowód. Skoro Q_n jest dwudzielna, to każda ścieżka musi odwiedzić tyle samo wierzchołków w obu częściach (plus jeden koniec), ponieważ w części z u i v odwiedza co najwyżej $2^{n-1} - |F|$, to w drugiej co najwyżej $2^{n-1} - |F| - 1$ – daje to długość $2^n - 2|F| - 2$. \square

Definicja 2.3.3. Wierzchołek $v \in V(Q_n)$ jest otoczony przez $F \subseteq V(Q_n)$ gdy $N(v) \subseteq F$ (F zawiera wszystkich sąsiadów v).

Definicja 2.3.4. Dla $u, v \in V(Q_n), F \subseteq V(Q_n)$ trójka (u, v, F) jest zablokowana w Q_n gdy u jest otoczony przez $\{v\} \cup F$ lub v jest otoczony przez $\{u\} \cup F$.

2.3.2. Długie ścieżki

Twierdzenie 25. Dla Q_n i $F \subseteq V(Q_n)$, takich, że $2 \leq n \geq 5$ i $|F| \leq 2n - 4$ dla $u, v \in V(Q_n) \setminus F$ takich, że trójka (u, v, F) nie jest zablokowana w Q_n długa uv -ścieżka bez wad nie istnieje tylko wtedy, gdy $n = 4$ oraz istnieją takie $a, b \in V(Q_n)$, że $d(a, b) = 4$ i $F \cup \{u, v, a, b\}$ jest dwudzielną częścią Q_n .

Dowód. Łatwo rozpatrzyć wszystkie przypadki. \square

Twierdzenie 26. Dla Q_n i $F \subseteq V(Q_n)$, takich że $n \geq 6$ i $|F| \leq 2n - 4$ dla każdych $u, v \in V(Q_n) \setminus F$ jeśli (u, v, F) nie jest zablokowane w Q_n , to istnieje długa uv -ścieżka bez wad.

Dowód. (krótki szkic dowodu z pracy [4])

Dowód oparty jest na indukcji po wymiarze. Podstawę indukcji stanowi twierdzenie 25. Dla $n \geq 6$, $|F| \leq 2n - 4$ można podzielić Q_n na dwie kostki Q_{n-1} wybierając jedną z n współrzędnych i definiując podkostki Q_{n-1}^0 i Q_{n-1}^1 jako rozpięte przez wierzchołki mające na tej współrzędnej odpowiednio 0 i 1. Dla $|F| \leq 2n - 5$ łatwo jest dobrać współrzędną tak, żeby każda z podkostek miała co najwyżej $2n - 6 = 2(n - 1) - 4$ wadliwych wierzchołków (wystarczy wybrać dowolne $f_1, f_2 \in F$ i podzielić według jednej ze współrzędnych różniących ich ciągi binarne). Dla $|F| = 2n - 4$ można rozpatrzyć macierz $|F| \times n$, w której w wierszach wypisane są ciągi binarne wszystkich wierzchołków wadliwych. Trzeba wybrać taką kolumnę, w której zarówno 0 jak i 1 jest co najmniej po 2. Gdyby nie dało się dokonać takiego wyboru oznaczałoby to, że w każdej kolumnie jest albo co najwyżej jedno 0 albo co najwyżej jedna 1, przez proste zanegowanie jednej współrzędnej w całej kostce (ta operacja nie zmienia nic poza numerowaniem) można uzyskać przypadek, że w każdej kolumnie jest co najwyżej jedna 1. Ponieważ kolumn jest tylko n , zaś każda zawiera co najwyżej jedną 1, to oznaczałoby to, że może mieć tylko $n + 1$ różnych wierszy $\Rightarrow 2n - 4 = |F| \leq n + 1 \Rightarrow n \leq 5$ (a więc ponieważ $n \geq 6$, to zawsze istnieje wybór współrzędnej).

Dalej przy użyciu lematów:

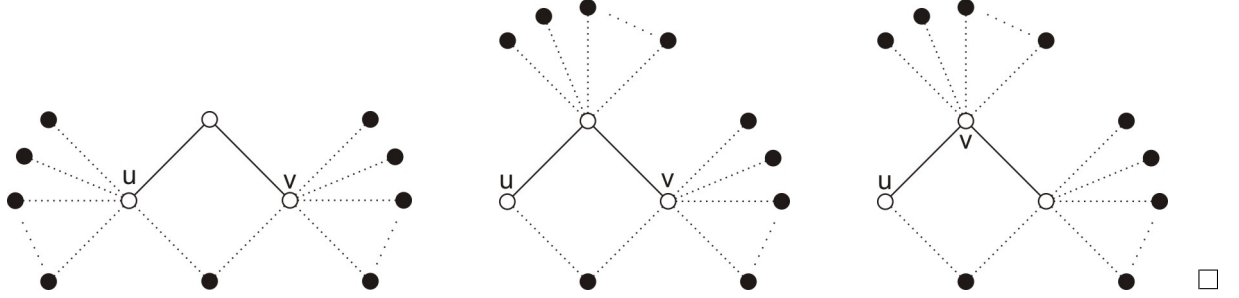
- Dla $|F| \leq 2n - 3$ co najwyżej jeden z wierzchołków jest otoczony przez F .
- Dla $|F| \leq 2n - 4$ i ustalonego nieotoczonego wierzchołka u istnieje co najwyżej jeden wierzchołek v taki, że (u, v, F) jest zablokowana.

- Dla $|F| \leq 2n - 5$ tylko jedna trójka (u, v, F) może być zablokowana, i to taka, że $uv \in E(Q_n - F)$.

i wykorzystując fakt, że w podkostkach poza kilkoma przypadkami istnieją odpowiednie długie kostki rozważa się dużą liczbę przypadków (rozbicie ze względu na należenie u i v do tej samej/różnej podkostki, bycia otoczonym/zablokowanym/wolnym w podkostce). Dla każdego z tych przypadków da się pokazać metodę łączenia długich ścieżek z podkostek. \square

Uwaga 27. Dla Q_n , $F \subseteq V(Q_n)$, $|F| = 2n - 3$ teza twierdzenia 26 przestaje być prawdziwa.

Dowód. Dla każdego $n \geq 4$ istnieje po kilka przypadków w których $|F| = 2n - 3$, (u, v, F) nie jest zablokowane, ale nie ma długiej uv -ścieżki bez wad. 3 przykłady :



2.3.3. Długie cykle

Definicja 2.3.5. Dla zbioru $D \subseteq [n]$, $d = |D|$ oraz $u \in \{0, 1\}^{n-d}$ definiujemy kostkę $Q_D(u)$ jako d wymiarową podkostkę Q_n , której współrzędne spoza D są ustalone przez wektor u . Definiujemy również $V_D(u) = \{(u, v)_D : v \in \{0, 1\}^d\}$ (wierzchołki z oryginalnej kostki wzięte do $Q_D(u)$), oraz $F_D(u) = F \cap V_D(u)$.

Lemat 28. Niech $F \subseteq V(Q_n)$ takie, że $|F| \geq 2n$ i niech $d = \lceil \frac{n^2}{2|F|-n-2} \rceil$. Wtedy istnieje zbiór $D \subseteq [n]$, $|D| = d$, taki że $|F_D(u)| \leq d + 1$ dla każdego $u \in \{0, 1\}^{n-d}$.

Lemat pochodzi z pracy [5] i został zmodyfikowany do tej postaci w pracy [4] aby lepiej pasować do dowodu poniższego twierdzenia.

Twierdzenie 29. Dla $n \geq 15$ i $F \subseteq V(Q_n)$, takiego że $|F| \leq \frac{n^2}{10} + \frac{n}{2} + 1$ istnieje długi cykl bez wad.

Dowód. (krótki szkic dowodu z pracy [4])

Na podstawie lematu 28 znajdujemy zbiór $D \subseteq [n]$, taki że $|F_D(u)| \leq 2d - 4$ dla każdego $u \in \{0, 1\}^{n-d}$. Dla dowolnego cyklu Hamiltona $(u_0, u_1, \dots, u_{2^n-d} = u_0)$ w Q_{n-d} dobieramy w kostce $Q_D u_i$ dwa nie wadliwe wierzchołki a_i oraz b_i , takie że $a_i b_{i+1} \in E(Q_n)$ dla każdego $i \in [2^{n-d}]$ (modulo 2^{n-d}), oraz $(a_i, b_i, F_D(u^i))$ nie zablokowane (choć jest to nietrywialne to da się takie dobrać). Na podstawie twierdzenia 26 wierzchołki a_i i b_i są łączone długimi ścieżkami dając cykl długości $\geq 2^n - 2|F|$. Ograniczenie $|F| \leq \frac{n^2}{10} + \frac{n}{2} + 1$ potrzebne jest po to, aby $\lceil \frac{n^2}{2|F|-n-2} \rceil \geq 5$ omijając złe przypadki z twierdzenia 25. \square

2.3.4. Długie pary ścieżek

Lemat 30. Dla $n \geq 2$, $F \subseteq V(Q_n)$, $|F| \leq n - 2$ dla każdych dwóch $u, v \in V(Q_n - F)$ istnieje długa uv -ścieżka bez wad.

Dowód. Bezpośrednio z 26, gdzie ze względu na rozmiar F trójka (u, v, F) nie może być zablokowana. \square

Twierdzenie 31. *Dla $F \subseteq V(Q_n)$, $F \leq n - 3$ niech A i B będą różnymi dwuelementowymi podzbiorami $V(Q_n) - F$, takimi że $A \cup B$ nie należy do jednej części dwudzielnej kostki. Wtedy istnieje para wierzchołkowo rozłącznych ścieżek o łącznej długości $\geq 2^n - 2|F| - 3$ zaczynających się w wierzchołku z A i kończących na wierzchołku z B .*

Dowód. (krótki szkic dowodu z pracy [6])

Dowód podobnie jak inne przebiega indukcyjnie – małe przypadki ($n \leq 5$) można sprawdzić ręcznie (tutaj trochę więcej sprawdzania niż w poprzednich dowodach), dla większych łatwo jest podzielić Q_n na dwie Q_{n-1} tak, żeby każda z nich miała nie więcej niż $n - 4$ wierzchołków wadliwych. Dalej rozpatrywane jest dużo przypadków w zależności od podziału wierzchołków z A i B na dwie podkostki i w każdym z możliwych przypadków łączy się podwójne i pojedyncze ścieżki istniejące na mocy indukcji i twierdzenia 26. \square

Uwaga 32. *Jeśli $A = \{u, w\}$, $B = \{v, w\}$, to jedna ze ścieżek musi mieć długość 0 i być zaczepiona w wierzchołku w . Twierdzenie 31 daje wtedy uv -ścieżkę wolną od wad długości $2^n - 2|F| - 3 = 2^n - 2|F \cup \{w\}| - 1$, a więc o jeden dłuższą niż w twierdzeniu 26 (możliwe jest to tylko dlatego, że u i v nie należą do jednej części dwudzielnej Q_n).*

2.4. Inne problemy

Twierdzenie 33. *Dla $F \subseteq V(Q_n)$ można rozstrzygnąć, czy w $Q_n - F$ jest cykl Eulera w czasie $O(|F| \cdot n^3)$*

Dowód. Kryterium istnienia cyklu Eulera jest to, że po pierwsze graf jest spójny, a po drugie z każdego wierzchołka wychodzi parzyste wiele krawędzi. Spójność można sprawdzić w czasie $O(|F| \cdot n^3)$ przy pomocy algorytmu z podrozdziału 2.2.2. Wierzchołek nie mający wadliwego sąsiada ma stopień n , wystarczy więc policzyć tylko parzystość dla tych którzy takiego sąsiada mają. W czasie i pamięci $O(|F| \cdot n^2)$ można wstawić wszystkich niewadliwych sąsiadów wierzchołków wadliwych do drzewa prefiksowego zapamiętując w liściach krotność. Po wszystkim wystarczy dla $2|n|$ sprawdzić czy wszystkie wstawione wierzchołki mają krotność parzystą, zaś dla $2 \nmid n$ trzeba po pierwsze sprawdzić, że wszystkie wstawione wierzchołki mają krotność nieparzystą, a po drugie że jest ich dokładnie $2^n - |F|$. \square

Bibliografia

- [1] L. Lovasz, J. Pelikan and K. Vesztergombi. "Discrete Mathematics, Elementary and Beyond." *Undergraduate Texts in Mathematics. New York: Springer, first edition, 2003*
- [2] L. H. HARPER, "Optimal Numberings and Isoperimetric Problems on Graphs" *JOURNAL OF COMBINATORIAL THEORY 1, 385-393 (1966)*
- [3] Tomas Dvorak, Jiri Fink, Petr Gregor, Vaclav Koubek and Tomasz Radzik, "Efficient connectivity testing of hypercubic networks with faults"
- [4] Jiri Fink and Petr Gregor, "Long paths and cycles in hypercubes with faulty vertices"
- [5] G. Wiener, "Edge multiplicity and other trace functions." *In Proceedings of European Conferenc on Combinatorics, Graph Theory and Applications (EuroComb 2007), volume 29 of Electronic Notes in Discrete Mathematics, pages 491-495, 2007.*
- [6] Jiri Fink and Petr Gregor, "Long pairs of paths in faulty hypercubes"
- [7] Frank Harary, John P. Hayes and Horng-Jyh Wu, "A survey of theory of hypercube graphs" *Comput. Math. Applic. Vol. 15, No 4, pp. 277-289, 1988*
- [8] Frank Harary, Marilynn Livingston, "Independent domination in hypercubes" *Appl. Math. Lett. Vol. 6, No 3, pp. 27-28, 1993*
- [9] Wojciech Rytter & Bartosz Szreder, "Wprowadzenie do kombinatoryki algorytmicznej"
- [10] DONALD E. KNUTH, "Generating All Tuples and Permutations" *THE ART OF COMPUTER PROGRAMMING VOLUME 4, FASCICLE 2*
- [11] FRANK RUSKEY, "Combinatorial Generation" *Working Version, October 1, 2003*
- [12] Tibor Szabo, Emo Wertz, "Unique Sink Orientations of Cubes"
- [13] Chi Him Wong, "Novel universal cycle constructions for a variety of combinatorial objects" *Guelph, Ontario, Canada, April, 2015*