

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Wiktor Zuba

Nr albumu: 320501

**Efektywne algorytmy
kombinatoryczne na hiperkostce z
wadliwymi wierzchołkami**

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 combinatorical algorithms on a hypercube with faulty vertices

Spis treści

Wprowadzenie	5
1. Własności hiperkostki	7
1.1. Podstawowe definicje	7
1.1.1. Hiperkostka	7
1.1.2. Sąsiedztwo	8
1.1.3. Podgrafy	9
1.1.4. Graf z wadami	9
1.2. Podstawy kombinatoryczne	9
1.3. Własność podziałowa	10
1.4. Własność ekspansji	10
1.5. Inne własności	12
1.5.1. Automorfizmy	12
1.5.2. Kolorowanie	12
1.5.3. Ilość drzew rozpinających	12
1.5.4. Rysowanie na płaszczyźnie	13
2. Spójność wadliwej hiperkostki	15
2.1. Podejście ekspansywne	15
2.1.1. pseudokod i uwagi	16
2.2. Redukcja do lokalnej spójności przy pomocy transformacji ścieżek	17
2.2.1. Transformacje ścieżek w hiperkostce	18
2.2.2. Dowód twierdzenia o lokalnej spójności	20
2.2.3. Algorytm	21
2.2.4. Analiza złożoności	23
2.3. Wnioski i zastosowania	23
2.3.1. Cykl Eulera	23
2.3.2. Istnienie ścieżki między dwoma punktami	23
2.3.3. Długość ścieżki między dwoma punktami	24
3. Długie ścieżki i cykle w grafie	25
3.1. Definicje	25
3.2. Długie ścieżki	25
3.3. Długie cykle	27
3.4. Długie pary ścieżek	28

4. Ścieżka i cykl Hamiltona	29
4.1. Kostka bez wadliwych wierzchołków	29
4.2. Ogólne podejście do problemu	29
4.3. Szkice dowodów dla szczególnych przypadków	30
4.3.1. podkostka	30
4.3.2. Ścieżka izometryczna	31
4.3.3. Cykl izometryczny	32
4.3.4. Drzewo izometryczne	33
5. Maksymalna ścieżka indukowana	35
5.1. Ścieżka Fibonacciego	35
5.2. Wyniki eksperymentalne	36
5.2.1. NMCS	37
5.2.2. Uwagi praktyczne	38
5.2.3. Uzyskane wartości	39
5.3. Znane ograniczenia długości węży	40
5.3.1. Ograniczenia dolne	40
5.3.2. Ograniczenia górne	41
Bibliografia	43

Wprowadzenie

Jedną z najbardziej popularnych i regularnych klas grafów, są hiperkostki – grafy wierzchołków kostek n -wymiarowych, które połączone są krawędziami odpowiadającymi krawędziom jednowymiarowym kostki. Graf taki jest często wykorzystywany jako struktura sieci komputerowym, dzięki połączeniu cech względnie niskiej gęstości i dużej odporności na usterki w dowolnych węzłach i połączeniach. Badanie własności tych sieci z możliwymi usterekami jest jedną z motywacji do badania nowej klasy grafów – hiperkostek z wadami, która obejmuje hiperkostki, w których część wierzchołków i krawędzi jest oznaczona jako wadliwe lub usunięta (zazwyczaj jest to ilość mała względem wielkości grafu w szczególności wielomianowa od wymiaru kostki). Pomimo, że grafy te są wciąż wykładniczo wielkie w stosunku do wymiaru, to ze względu na wyjątkowe własności hiperkostki wiele standardowych problemów grafowych można rozwiązać w czasie wielomianowym od tego wymiaru i ilości wad.

Grafy te reprezentować można również jako zbiór wszystkich ciągów binarnych długości n , z relacją krawędzi pomiędzy dwoma ciągami różniącymi się na jednym bicie. Dzięki tej interpretacji chodzenie po grafie przydaje się do wielu algorytmów tekstowych. Dowolny cykl Hamiltona hiperkostki daje kod Greya – przyiterowanie po wszystkich możliwych słowach binarnych określonej długości jednocześnie zmieniając za każdym razem zaledwie jeden bit.

Niniejsza praca ma na celu przedstawienie ważniejszych własności wadliwych hiperkostek, części z najważniejszych problemów grafowych oraz znalezionych w literaturze algorytmów rozwiązujących je w złożoności lepszej niż ta dla grafów bez specjalnej struktury.

Praca składa się z pięciu rozdziałów. W pierwszym przedstawiam definicje obiektów używanych w pracy, oraz ważniejsze własności hiperkostki z sporej części korzystam w późniejszych rozważaniach nad hiperkostkami z wadami.

W rozdziale 2 rozpatrywany jest problem spójności w hiperkostce z usuniętymi wierzchołkami, który jest niezwykle istotny dla sieci komputerowych z tą strukturą, w której pewne komputery mogą ulec awarii. Przedstawione są tam dwa algorytmy rozwiązujące problem w czasie wielomianowym od wymiaru kostki n i liczności zbioru wierzchołków wadliwych F – algorytm korzystający z własności ϵ -ekspansji wierzchołkowej w czasie $O(|F|^2 \cdot n^{3.5})$, oraz drugi korzystający z lokalnej spójności w czasie $O(|F| \cdot n^3)$. Wymieniam tam również kilka problemów pobocznych rozwiązywanych przez te algorytmy.

W kolejnym rozdziale wprowadzam problem bardzo długich (wielkości powyżej $2^n - 2|F| - 4$) ścieżek oraz cykli oraz par ścieżek występujących w wadliwych hiperkostkach. Przedstawiam tam kilka twierdzeń dających proste warunki wystarczające na ich istnienie w grafie. Posiadanie tych właściwości pozwala na użycie algorytmów wykorzystujących chodzenie po hiperkostce na prawie całym grafie.

W czwartym rozdziale przedstawiam kilka wyników teoretycznych na temat istnienia cykli Hamiltona i hamiltonowskiej wiązalności w hiperkostkach z usuniętymi wierzchołkami. Następnie naszkicowuje dowody hamiltonowskiej wiązalności dla wadliwych hiperkostek dla zbiorów wad indukujących w hiperkostce podgrafy o specjalnych strukturach (hiperkostka,

ścieżka, cykl, drzewo).

W ostatnim rozdziale rozpatruje problem możliwie najdłuższej ścieżki indukowanej w hiperkostce (hiperkostka z usuniętymi wierzchołkami, która jest ścieżką) nazywany w literaturze węzem w pudełku. Przedstawiam tam własną konstrukcję ścieżek, które dla hiperkostki wymiaru n mają rozmiar równy $n+1$ wyrazowi ciągu Fibonacciego oraz wyniki eksperymentalne dla $n \leq 34$ (czyli dla większej ilości niż w większości prac). Pod koniec rozdziału wymieniam też znaną w literaturze, trudniejszą lecz dającą lepsze rezultaty konstrukcję długiego węża oraz ograniczenia górne.

Rozdział 1

Własności hiperkostki

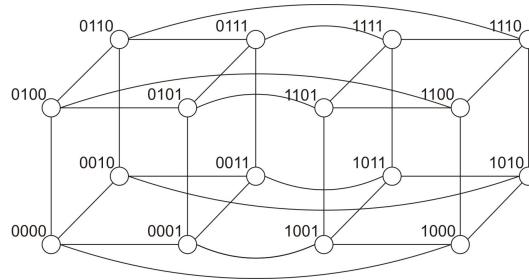
1.1. Podstawowe definicje

Definicja 1.1. Dla $n \in \mathbb{N}$ $[n] = \{0, \dots, n-1\}$ (zbiór pierwszych n liczb naturalnych).

1.1.1. Hiperkostka

Definicja 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.

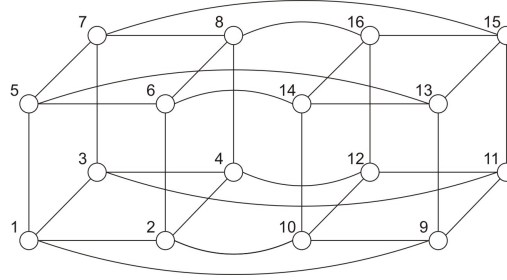
Hiperkostka jest grafem dwudzielnym, w którym jedną częścią jest zbiór wierzchołków o ciągach z parzystą liczbą jedynek, zaś drugą tych o ich nieparzystej liczbie.

Co więcej 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 ($\bar{0}$)). W tym ujęciu części dwudzielne składają się odpowiednio z parzystych i z nieparzystych warstw.

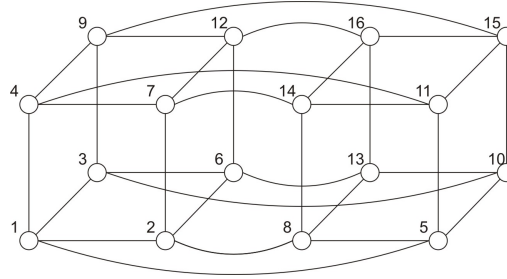
Definicja 1.3. Ze względu na oczywistą interpretację hiperkostki jako punkty i odcinki w przestrzeni n -wymiarowej możemy mówić o kierunku krawędzi wychodzącej z wierzchołka, definiowanym jako numer pozycji (lub równoważnie współrzędnej), która ulega zmianie po przejściu przez tę krawędź.

Definicja 1.4. 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 1.5. 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 1.6. Numerowaniem warstwowym hiperkostki nazwiemy jej numerowanie w kolejności przeszukiwania grafu wszerz zaczynając od wierzchołka $\bar{0}$ z wybieraniem sąsiadów w kolejności leksykograficznej.



Uwaga 1.7. 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

1.1.2. Sąsiedztwo

Definicja 1.8. 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 1.9. Dla grafu G oraz zbioru wierzchołków $S \subseteq V(G)$ definiujemy otoczenie 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 1.10. 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: $In(S) = \{v \in S : N(v) \subseteq S\}$

1.1.3. Podgrafy

Definicja 1.11. 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 1.12. Dla danego $V \subseteq V(G)$ $G - V = G[V(G) \setminus V]$ oznacza graf G z usuniętymi wierzchołkami V .

Definicja 1.13. 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.

1.1.4. Graf z wadami

Definicja 1.14. 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.

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

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

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

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

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

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

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 :

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

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

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

□

Lemat 1.16. $2^n(\log_2(n) - 2) \leq \sum_{i=1}^n \binom{n}{i} \log_2(i) \leq 2^n(\log_2(n) - \frac{1}{2})$ (dla dostatecznie dużych n).

Dowód. Dla $n = 2k + 1$ dzielimy sumę na $\sum_{i=1}^k \binom{n}{i} \log_2(i) + \sum_{i=k+1}^n \binom{n}{i} \log_2(i)$.

$$\sum_{i=k+1}^n \binom{n}{i} = 2^{n-1} \Rightarrow 2^{n-1}(\log_2 n - 1) \leq \sum_{i=k+1}^n \binom{n}{i} \log_2(i) \leq 2^{n-1} \log_2 n.$$

$$\sum_{i=1}^k \log_2(i) = \log_2(k!) \text{ (ze wzoru Stirlinga)} \geq k \log_2(k) - k \log_2(e) - \frac{1}{2}(\log_2(2\pi) + \log_2(k)) \geq k(\log_2(k + \frac{1}{2}) - 2) \text{ (dla } k > 7) = k(\log_2(n) - 3)$$

$$\text{Dla } i \leq k \binom{n}{i} \leq \binom{n}{i+1}, \log_2(i) < \log_2(i+1), \text{ więc } 2^{n-1}(\log_2(n) - 2) \leq (\sum_{i=1}^k \binom{n}{i}) \cdot (\frac{1}{k} \sum_{i=1}^k \log_2(i)) \leq$$

$$\sum_{i=1}^k \binom{n}{i} \log_2(i) \leq \log_2(k + \frac{1}{2}) \cdot (\sum_{i=1}^k \binom{n}{i}) = 2^{n-1}(\log_2(n) - 1). \text{ Po połączeniu daje to:}$$

$$2^n(\log_2(n) - 2) \leq \sum_{i=1}^n \binom{n}{i} \log_2(i) \leq 2^n(\log_2(n) - \frac{1}{2}).$$

Dla parzystego n po wyłączeniu wyrazu środkowego można analogicznie szacować części przed i po tym wyrazie (po podstawieniu $2^n - \binom{n}{\frac{n}{2}}$ zamiast 2^n), zaś środkowy wyraz jest równy $\binom{n}{\frac{n}{2}}(\log_2(n) - 1)$, a więc pomiędzy $\binom{n}{\frac{n}{2}}(\log_2(n) - 2)$ i $\binom{n}{\frac{n}{2}}(\log_2(n) - \frac{1}{2})$, co kończy dowód. □

1.3. Własność podziałowa

Twierdzenie 1.17. Dla dowolnego podziału wierzchołków na dwa zbiory $S \in V(G)$ i $V(Q_n) \setminus S$ liczba krawędzi między tymi zbiorami jest co najmniej tak duża jak liczność mniejszego ze zbiorów: $|\{(u, v) \in E(Q_n) : u \in S, v \in V(Q_n) \setminus S\}| \geq \min(|S|, |V(Q_n) \setminus S|)$.

Dowód. Załóżmy, że $m = |S| \leq 2^{n-1}$.

Dla każdej pary (u, v) wierzchołków hiperkostki wybieramy jedną uv -ścieżkę taką, że jest najkrótsza (krawędź każdego kierunku jest używana co najwyżej raz) oraz idąc od u współrzędne kierunkowe krawędzi rosną. Każda krawędź w hiperkostce występuje na dokładnie 2^{n-1} takich ścieżkach (dla każdego innego kierunku możemy wybrać czy występuje na ścieżce i zawsze otrzymamy poprawną ścieżkę należącą do tego zbioru). Na każdej ścieżce pomiędzy wierzchołkami $u \in S$ i $v \in V(Q_n) \setminus S$ występuje co najmniej jedna krawędź z interesującego nas zbioru, zaś wszystkich takich ścieżek jest $m \cdot (2^n - m)$. Pomiedzy S i $V(Q_n) \setminus S$ jest więc co najmniej $\frac{m \cdot (2^n - m)}{2^{n-1}} = 2m - \frac{m^2}{2^{n-1}} \geq m$ krawędzi. □

1.4. Własność ekspansji

Hiperkostka posiada także mocniejszą własność zdefiniowaną i udowodnioną w tym podrozdziale.

Definicja 1.18. 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 1.19. Zbiór S_l – pierwszych l wierzchołków hiperkostki według numerowania warstwowego posiada maksymalne wnętrze wśród zbiorów wielkości l .

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

Lemat 1.20. 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 1.19 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 $\Rightarrow |N(S_k)| = \binom{2m-1}{m} + \binom{2m-1}{m-1} = \binom{2m-1}{m-1} + \binom{2m-1}{m} = \binom{2m}{m}$. 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

Twierdzenie 1.21. 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 $\frac{|N(S_k)|}{|S_k|} = \frac{\binom{n}{r+1}}{\sum_{i=0}^r \binom{n}{i}} > \frac{\binom{n}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1} \binom{n}{r+1}} = \frac{\binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1}} > \frac{2^n}{2^{n-1} \sqrt{\pi(\lfloor \frac{n}{2} \rfloor + \frac{1}{2})}} = \frac{2}{\sqrt{\pi(\lfloor \frac{n}{2} \rfloor + \frac{1}{2})}} \geq \frac{2\sqrt{2}}{\sqrt{\pi(n + \frac{3}{2})}} \geq \frac{2}{\sqrt{\pi n}}$ (dla $n \geq 2$).

($n = 2m + 1, r = m$ rozważone w 1.20)

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).

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

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

$$\frac{2 \cdot \binom{n-1}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^n \cdot \binom{n-1}{r+1} + \binom{n-1}{r} \cdot \binom{n}{\lfloor \frac{n}{2} \rfloor}} = \left(\frac{2^{n-1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} + \frac{\binom{n-1}{r}}{2 \cdot \binom{n-1}{r+1}} \right)^{-1} > \left(\frac{\sqrt{\pi(n + \frac{3}{2})}}{2\sqrt{2}} + \frac{1}{2} \right)^{-1} = \frac{2\sqrt{2}}{\sqrt{\pi(n + \frac{3}{2})} + \sqrt{2}} > \frac{2}{\sqrt{\pi n}} \text{ (dla } n \geq 7).$$

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

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

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

$n \geq 7$).

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

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

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

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

Wniosek 1.22. *Hiperkostka wymiaru n nie posiada własności $\frac{2\sqrt{2}}{\sqrt{\pi n}}$ -ekspansji wierzchołkowej.*

Dowód. Dla $n = 2m + 1$

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

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

1.5. Inne własności

1.5.1. Automorfizmy

Fakt 1.23. *Hiperkostka Q_n ma dokładnie $2^n \cdot n!$ automorfizmów – na $n!$ sposobów można dokonać permutacji współrzędnych, zaś na 2^n sposobów zanegować część współrzędnych lub równoważnie wybrać który wierzchołek otrzyma współrzędne $(0, \dots, 0)$.*

Oznacza to, że przy dowolnych rozważaniach w których wyróżniamy dwa wierzchołki odległe o k , możemy założyć że pierwszy z nich jest wierzchołkiem $\bar{0}$ zaś drugi ma współrzędne $(0, \dots, 0, 1, \dots, 1)$ (gdzie jedynek jest k), jeżeli tylko zamienimy odpowiednio współrzędne innych wierzchołków – w tym wadliwych.

1.5.2. Kolorowanie

Dla hiperkostki problemy kolorowania są trywialne:

Jako graf dwudzielny hiperkostka jest dwukolorowalna wierzchołkowo. Po usunięciu części wierzchołków ilość potrzebnych kolorów może się tylko zmniejszyć, jedynym grafem jednokolorowalnym jest graf bez krawędzi.

n kolorowanie krawędziowe Q_n można uzyskać kolorując krawędzie według numeru kierunku, jaki przyjmują, jest to jednocześnie jedyne takie kolorowanie. Z twierdzenia Vizinga dla grafu dwudzielnego ilość potrzebnych kolorów jest równa stopniowi grafu, a to można sprawdzić w czasie liniowym od ilości usuniętych wierzchołków (musi być ich co najmniej $\frac{2^n}{n}$, aby każdy wierzchołek stracił sąsiada).

1.5.3. Ilość drzew rozpinających

Definicja 1.24. *Macierz Laplace'a grafu G , o n wierzchołkach, to macierz $[a_{i,j}]_{i,j \in [n]}$, taka że*

$$a_{i,i} = \deg(v_i), \quad a_{i,j} = -1 \cdot [(v_i, v_j) \in E(G)] \text{ (dla } i \neq j \text{)}.$$

Twierdzenie 1.25. *(Kirchhoffa)*

Dla spójnego grafu o n wierzchołkach ilość drzew rozpinających jest równa $\frac{1}{n} \cdot \lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_{n-1}$, gdzie $\lambda_1, \lambda_2, \dots, \lambda_{n-1}$ to niezerowe wartości własne jego macierzy Laplace'a (z krotnościami).

Lemat 1.26. Wartościami własnymi Q_n są liczby $2i$ dla $i \in [n+1]$, z krotnością $\binom{n}{i}$.

Dowód. Dowód indukcyjny – dla $n = 1$ macierz $\begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}$ ma wartości własne 0 i 2 z ortogonalnymi wektorami własnymi $[1, 1]$ i $[1, -1]$.

Mając macierz Laplace’a A_n grafu Q_n macierz A_{n+1} dla Q_{n+1} jest równa $\begin{vmatrix} A_n + I_{2^n} & -I_{2^n} \\ -I_{2^n} & A_n + I_{2^n} \end{vmatrix}$.

Niech v będzie wektorem własnym macierzy A_n dla wartości własnej λ . Wtedy:

$$A_{n+1} \cdot \begin{vmatrix} v \\ v \end{vmatrix} = \begin{vmatrix} A_n + I_{2^n} & -I_{2^n} \\ -I_{2^n} & A_n + I_{2^n} \end{vmatrix} \cdot \begin{vmatrix} v \\ v \end{vmatrix} = \begin{vmatrix} \lambda \cdot v + v - v \\ \lambda \cdot v + v - v \end{vmatrix} = \lambda \cdot \begin{vmatrix} v \\ v \end{vmatrix}$$

$$A_{n+1} \cdot \begin{vmatrix} v \\ -v \end{vmatrix} = \begin{vmatrix} A_n + I_{2^n} & -I_{2^n} \\ -I_{2^n} & A_n + I_{2^n} \end{vmatrix} \cdot \begin{vmatrix} v \\ -v \end{vmatrix} = \begin{vmatrix} \lambda \cdot v + v + v \\ -\lambda \cdot v - v - v \end{vmatrix} = (\lambda + 2) \cdot \begin{vmatrix} v \\ -v \end{vmatrix}.$$

Ponieważ wszystkie 2^n wektorów własnych z poprzedniego kroku były ortogonalne, to otrzymane teraz 2^{n+1} również takie będzie. Stąd dla wartości własnej $2i$ macierzy A_{n+1} otrzymujemy $\binom{n}{i} + \binom{n}{i-1} = \binom{n+1}{i}$ ortogonalnych wektorów własnych. \square

Uwaga 1.27. Korzystając z twierdzenia Kirchhoffa, Q_n ma $\frac{1}{2^n} \prod_{i=1}^n (2i)^{\binom{n}{i}}$ drzew rozpinających.

Macierz Laplace’a grafu pełnego o n wierzchołkach jest równa $n \cdot I_n - J_n$ (gdzie J_n , to macierz $n \times n$ złożona z samych 1), więc ma wartość własną n ($n-1$)-krotną (+ 0 jednokrotną).

Macierz Laplace’a grafu pełnego dwudzielnego, w którym każda część dwudzielna ma po n wierzchołków jest równa $\begin{vmatrix} nI_n & -J_n \\ -J_n & nI_n \end{vmatrix}$, więc ma wartość własną n ($2n-2$)-krotną i $2n$ jednokrotną (+ 0 jednokrotną).

Jeśli więc porówna się logarytmy ilości drzew rozpinających tych trzech grafów o tej samej liczbie wierzchołków, to wychodzi:

$$\log_2(t(K_{2^n})) = n(2^n - 2)$$

$$\log_2(t(K_{2^{n-1}, 2^{n-1}})) = (n-1)(2^n - 2)$$

$$\log(t(Q_n)) = 2^n - n - 1 + \sum_{i=1}^n \binom{n}{i} \log(i) \approx (\log_2(n))2^n \text{ (na podstawie lematu 1.16).}$$

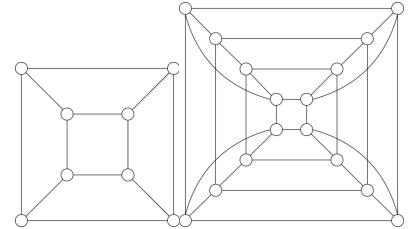
1.5.4. Rysowanie na płaszczyźnie

Q_n jest grafem planarnym, jedynie dla $n \leq 3$.

Dla $n = 4$ na płaszczyźnie powstaje minimum 8 przecięć krawędzi, jest ona jednak grafem toroidalnym (daje się narysować na powierzchni torusa bez przecięć).

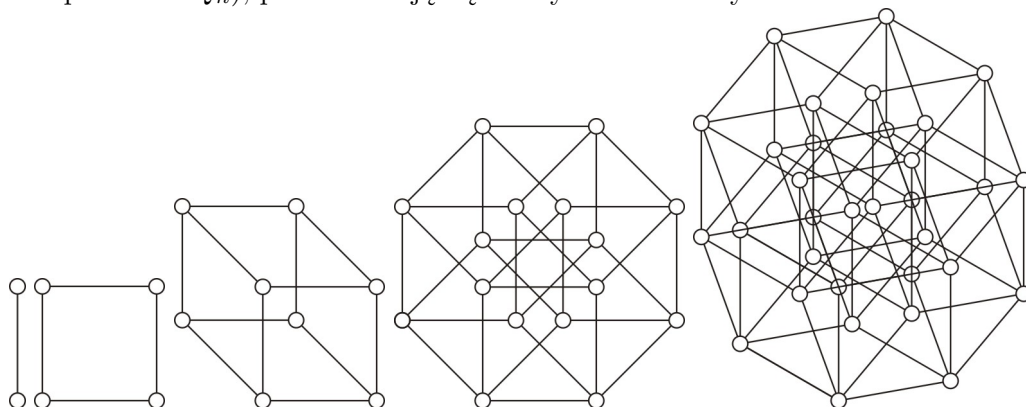
Ogólnie Q_n ma genus równy $\gamma(Q_n) = (n-4)2^{n-3} + 1$

([3]) – oznacza to, że kostkę taką daje się narysować bez przecięć na sferze z tyłoma "uszami".



Każda hiperkostka jest grafem jednostkowej odległości – można go narysować tak, żeby każda krawędź miała tę samą długość euklidesową. Rysunek taki można łatwo dostać indukcyjnie – malując dwie kopie rysunku Q_{n-1} o tej własności w odległości 1 w dowolnym kierunku (ponieważ kierunków jest continuum wiele zawsze można wybrać taki, by żadne dwa wierzchołki się nie pokrywały). Rysunki takie stają się jednak coraz gęstsze, jako że trzeba zmieścić 2^n wierzchołków i $n2^{n-1}$ krawędzi na kole o średnicy n (największa odległość

w hiperkostce Q_n), przez co stają się nieczytelne dla dużych n .



Rozdział 2

Spójność wadliwej hiperkostki

Ten rozdział jest napisany w większości na podstawie [4].

Uwaga 2.1. 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 (czy 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 mogły na niej występować wierzchołki, które zostały usunięte. Ścieżkę taką można naprawić – przed pierwszym odwiedzeniem i po ostatnim odwiedzeniu wierzchołka wadliwego odwiedzani są jacyś ich sąsiedzi – wystarczy ten fragment ścieżki zastąpić ścieżką między tymi dwoma sąsiadami, która istnieje w wadliwym grafie. \square

2.1. Podejście ekspansywne

Twierdzenie 2.2. 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 2.3. 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 2.4. 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 2.5. Może być tylko jedna składowa główna.

Dowód. (Lematu 2.3)

Weźmy spójną składową S grafu $G - F$ ($N_{G-F}(S) = \emptyset$), jeżeli $|S| \leq \frac{|V(G)|}{2}$, to z własności ε -ekspansji wierzchołkowej grafu G zachodzi $|N_G(S)| \geq \varepsilon \cdot |S|$ (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 $G - F$ otoczenie to było puste z grafu G trzeba usunąć co najmniej $N_G(S)$ wierzchołków. \square

Dowód. (Twierdzenia 2.2)

Chcemy sprawdzić, czy wszyscy sąsiedzi wierzchołków usuniętych należą do tej samej spójnej składowej. Na podstawie lematu 2.3, 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 przeczytanie odpowiedzi wyroczni zajmuje $O(\log(|V(G)|))$ czasu. Daje to złożoność z tezy twierdzenia. \square

Wniosek 2.6. *Ponieważ zgodnie z twierdzeniem 1.21 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 2.7. *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 przejranych 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.*

2.1.1. pseudokod i uwagi

W algorytmie wykorzystywana jest struktura T z operacjami

- $Insert(v, T)$ wstawiającą wierzchołek v do struktury T
- $Retrieve(v, T)$ zwracającą informacje o obecności wierzchołka v w strukturze T

które wymagają $O(n)$ czasu na wykonanie (jak w uwadze 2.7).

Przeszukiwanie grafowe odbywa się przy pomocy funkcji rekurencyjnej 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:

```

Conectivity( $n, F$ ){
     $T2 = \text{empty\_structure}T()$ ;
     $counter = 0$ ;
    foreach( $f \in F$ ){
         $Insert(f, T2)$ ;
         $counter++$ ;
    }
     $size = \text{sqrt}(\pi * n) * counter$ ;
    foreach( $f \in F$ ){
        foreach( $v \in N(f)$ ){
            if( $Retrieve(v, T2) == FALSE$ ){
                 $counter = 0$ ;
                 $T = T2$ ;
                if( $DFS(v) == FALSE$ ) return( $FALSE$ );
            }
        }
    }
    return( $TRUE$ );
}

```

Uwaga 2.8. Aby przeiterować po $N(v)$ wystarczy przeiterować się po współrzędnych uzyskując sąsiada poprzez zanegowanie tej współrzędnej w zapisie binarnym v .

Uwaga 2.9. 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. Redukcja do lokalnej spójności 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 2.10. Na potrzeby tego podrozdziału definiuję za pracą [4] 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 2.11. 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 $G(C) - C$.

Na potrzeby dowodu twierdzenia zdefiniuję kilka obiektów i operacji.

2.2.1. Transformacje ścieżek w hiperkostce

Definicja 2.12. 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 różnią się ciągi binarne wierzchołków v_{i-1} i v_i .

Fakt 2.13. 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 2.14. Dwie ścieżki, których sekwencje tranzycji τ, ρ spełniają $\forall_{i \in [n]} \#(\tau, i) = \#(\rho, i)$ nazywamy równoważnymi.

Uwaga 2.15. Dla dowolnych dwóch uv -ścieżek w Q_n istnieje sekwencja operacji $\text{swap}, \text{insert}, \text{delete}$ (w tej kolejności, bez przeplotów), 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 2.16. Na potrzeby dowodu twierdzenia 2.11 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).

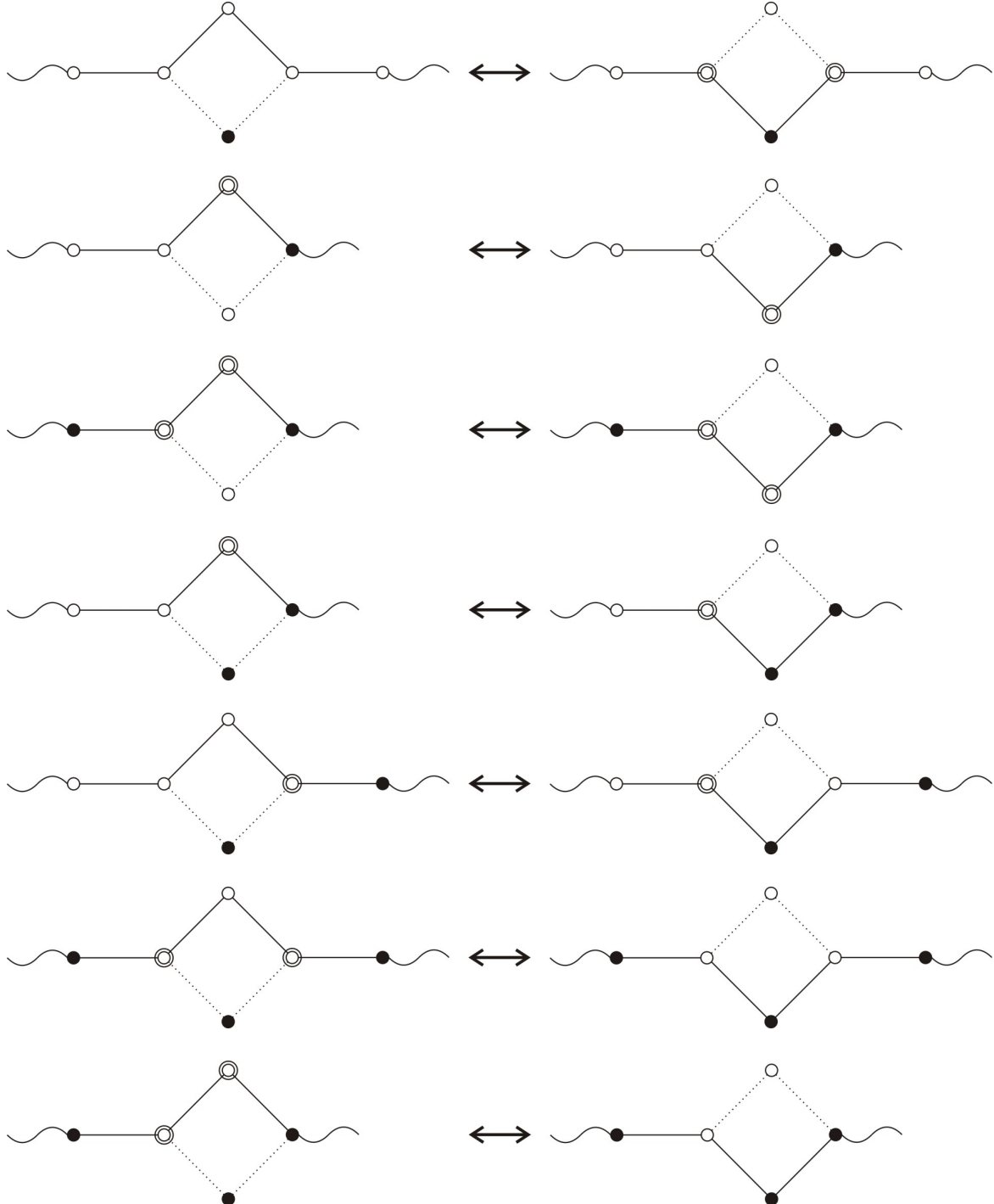
W przypadku tej definicji należy rozróżnić przeplatające się pojęcia wierzchołka grafu i jego wystąpienia na ścieżce – portem nazywane jest konkretne wystąpienie na ścieżce, inne jego wystąpienia nie muszą być portami.

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

Lemat 2.17. 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).

\circ – wierzchołek zwykły \bullet – wierzchołek wadliwy \odot – port
 ——— – krawędź ścieżki – krawędź spoza ścieżki ~~~~~ – nieistotna reszta ścieżki



Wszystkie rozrysowane tu wierzchołki należą do $G(F)$, a rozrysowane części niewadliwe tworzą podgraf spójny, dlatego też wszystkie te zmiany odbywają się w jednej spójnej składowej $G(F) - F$. Za każdym razem ilość portów zmienia się o 0 lub 2, a więc parzystość pozostaje bez zmian. \square

2.2.2. Dowód twierdzenia o lokalnej spójności

Lemat 2.18. *Niech $F \subseteq V(Q_n)$ będzie takie, że $G(F)$ jest spójne. Jeśli $Q_n - F$ jest spójne, to $G(F) - F$ również.*

Dowód. Załóżmy przeciwnie – istnieją wierzchołki $u, v \in A \cup B = V(G(F) - F)$ dla których istnieje ścieżka P w $Q_n - F$, ale nie ma takiej w $G(F) - F$. Skoro w $G(F) - F$ nie ma takiej ścieżki, to w P musi występować wierzchołek x spoza $A \cup B$.

$G(F)$ jest spójne, więc musi istnieć też druga ścieżka R łącząca u z v w tym właśnie grafie, na której występuje wierzchołek $y \in F$.

Podobnie jak w dowodzie lematu 2.15 ścieżki te mogą być napompowane w wierzchołkach x i y odpowiednio sekwencjami operacji *insert*.

Ponieważ x jest oddalone od F dodane do ścieżki P wierzchołki nie uczynią z żadnego wystąpienia x portu i same również nie staną się portami. Ponieważ y należy do F dodane do ścieżki R wierzchołki będą miały dokładnie dwóch sąsiadów z F (tego samego dwukrotnie), a więc nie będą portami.

Oznacza to, że uzyskane ścieżki równoważne mają tyle samo portów we wszystkich spójnych składowych $G(F) - F$ co odpowiadające im nieprzekształcone. Na podstawie lematów 2.15 i 2.17 ich parzystości się zgadzają (czyli zgadzają się również dla P i R). Na ścieżce P nie ma żadnych portów ponieważ nie występuje w niej żaden wierzchołek wadliwy. Ścieżka R w składowych $G(F) - F$ w których występują u i v ma nieparzyste ilości portów (np. można jako R dobrać taką ścieżkę, która wchodzi do/opuszcza składowe co najwyżej raz), co daje sprzeczność. \square

Wniosek 2.19. *Dla $F \subseteq V(Q_n)$ takiego, że $Q_n^2[F]$ jest spójne ze spójności $Q_n - F$ wynika spójność $G(F) - F$.*

Dowód. Jeśli dwa wierzchołki $Q_n^2[F]$ są połączone krawędzią, to w oryginalnym grafie musiały być w odległości ≤ 2 , a więc w $G(F)$ muszą być połączone albo bezpośrednią krawędzią albo poprzez pojedynczy wierzchołek ze zbioru A , a więc graf $G(F)$ jest spójny co sprawia, że spełnione są założenia lematu 2.18. \square

Lemat 2.20. *Niech $F \subseteq V(Q_n)$ będzie taki, że $G(C) - C$ jest spójne dla każdej C – spójnej składowej $Q_n^2[F]$. Wtedy $Q_n - F$ również jest spójny.*

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 spoza 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 tę wadliwą część ścieżki W zastąpić odpowiednią ścieżką z $G(C) - C$ aby zmniejszyć ilość wad na ścieżce. Po być może kilkukrotnym powtórzeniu tej operacji otrzymujemy poprawną ścieżkę w $Q_n - F$. \square

Dowód. (Twierdzenia 2.11)

Lemat 2.20 jest implikacją w jedną stronę. Implikacja w drugą stronę dla spójnego $Q_n^2[F]$ jest dana wnioskiem 2.19. 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

2.2.3. Algorytm

Stosując powyższe twierdzenie można uzyskać wielomianowy algorytm używając jedynie przeszukiwania grafowego podobnie jak w podrozdziale 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 (zaimplementowaną przy pomocy binarnego drzewa prefiksowego lub tablicy hashującej), przechowującą dla wierzchołka v_T informacje:
 - wskaźnik do wierzchołka v w strukturze D
 - informacje o wadliwości/braku wadliwości wierzchołka
 - binarną informację o tym czy wierzchołek należy do $F \cup N(F)$ i był już odwiedzony
 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 czasu $O(n)$ 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 $DFS(f)$ znajdująca spójne składowe $G(C) - C$, dla C – spójnej składowej $Q_n^2[F]$ zawierającej wadliwy wierzchołek f , o następującym pseudokodzie:

```

DFS(f){
  foreach( $u \in N(f)$ ){
     $u_T = Retrieve2(u, T)$ ;
    if( $u_T.visited == FALSE$ ){
       $u_T.visited = TRUE$ ;
      if( $u_T.healthy$ ){
        foreach( $v \in N(u)$ ){
           $v_T = Retrieve2(v, T)$ ;
          if( $v_T.healthy$ ){
            if( $Find(u, D) \neq Find(v, D)$ )   $Union(u, v, D)$ ;   $\setminus uv \in E(G(C) - C)$ 
          }else if( $v_T.visited == FALSE$ ){
             $v_T.visited = TRUE$ ;
             $DFS(v)$ ;   $\setminus$  wadliwy wierzchołek należący do  $C$  (oddalony od  $f$  o 2)
          }
        }
      }
    }else   $DFS(u)$ ;   $\setminus$  wadliwy wierzchołek należący do  $C$  (oddalony od  $f$  o 1)
  }
}

```

Powyższa prodedura uruchamiana jest z funkcji głównej:

```

Conectivity( $n, F$ ){
   $T = empty\_structureT()$ ;
  foreach( $f \in F$ ){
     $f_T = Insert(f, T)$ ;
     $f_T.healthy = FALSE$ ;
     $f_T.visited = FALSE$ ;
  }
  foreach( $f \in F$ ){
     $f_T = Retrieve(f, T)$ ;
    if( $f_T.visited == FALSE$ ){
       $f_T.visited = TRUE$ ;
       $D = empty\_structureD()$ ;
       $DFS(f)$ ;
      if( $D.counter > 1$ )   $return(FALSE)$ ;
    }
  }
  return( $TRUE$ );
}

```


2.2.4. Analiza złożoności

Wniosek 2.21. *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. Przeglądanie 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 czasowej przy pesymistycznym scenariuszu), zaś w przypadku struktur D łącznie $O(|F| \cdot n^2)$. \square

2.3. Wnioski i zastosowania

2.3.1. Cykl Eulera

Mając dostępny efektywny algorytm badania spójności wysnułem następujący wniosek:

Twierdzenie 2.22. *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 parzysta liczba krawędzi. Spójność można sprawdzić w czasie $O(|F| \cdot n^3)$ przy pomocy algorytmu z poprzedniego podrozdziału. 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

2.3.2. Istnienie ścieżki między dwoma punktami

Pomimo gorszej złożoności algorytmu ekspansywnego ma on wciąż przydatne zastosowania. Jeśli nie chcemy badać spójności całego grafu, a jedynie sprawdzić czy istnieje w nim ścieżka między dwoma wybranymi wierzchołkami algorytm ten łatwo zredukować:

Twierdzenie 2.23. *Dla $F \subseteq V(Q_n)$ i dwóch wierzchołków $u, v \in Q_n - F$ można w czasie $O(n^{2.5}|F|)$ rozstrzygnąć, czy w $Q_n - F$ istnieje uv -ścieżka.*

Dowód. Chcemy sprawdzić czy dwa wierzchołki należą do jednej spójnej składowej grafu $Q_n - F$. Zgodnie z lematem 2.3 jeśli spójne składowe w których znajdują się te wierzchołki są większe niż $\frac{|F|}{\varepsilon}$, to są składową główną (a więc tą samą), wystarczy więc z obu wierzchołków wystartować przeszukiwanie grafowe i zakończyć je kiedy zbada się tyle wierzchołków. Jeśli oba wyszukiwania zakończą się dzięki temu kryterium, lub jeśli zostanie napotkany ten drugi wierzchołek, to w $Q_n - F$ istnieje uv -ścieżka, w przeciwnym przypadku nie.

Tak przedstawiony algorytm działa dla dowolnego grafu (spójnego z którego usuwamy wierzchołki) z odpowiednią wartością ε . W przypadku hiperkostki wystarczy przeszukać $2\sqrt{\pi n} \cdot |F|$ wierzchołków, plus krawędzi z nich wychodzące (każdy ma ich n), podczas gdy każda taka operacja kosztuje $O(n)$, co daje złożoność z twierdzenia. \square

2.3.3. Długość ścieżki między dwoma punktami

Dzięki sprowadzeniu spójności do lokalnej spójności natychmiast zyskujemy konstrukcję ścieżki długości wielomianowej (od n i $|F|$) pomiędzy dowolnymi wierzchołkami w spójnej hiperkostce z wadami (choć da się uzyskać również ścieżki o wiele krótsze).

Fakt 2.24. *Dla spójnego grafu $Q_n - F$ pomiędzy każdymi dwoma wierzchołkami jest ścieżka długości nie większej niż $n + n^2 \cdot |F|$.*

Dowód. Jest to bezpośredni wniosek z dowodu lematu 2.20. Dwa dowolne wierzchołki są połączone w pełnej hiperkostce Q_n ścieżką długości co najwyżej n . W dowodzie zastępujemy (być może kilka razy) część takiej ścieżki inną ścieżką w grafie $G(C) - C$. Łącznie długość wszystkich takich zastąpień nie może wynosić więcej niż wynosi rozmiar $G(F) - F$, co dowodzi tezy. \square

Rozdział 3

Długie ścieżki i cykle w grafie

W poprzednim rozdziale 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 już 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".

3.1. Definicje

Definicja 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 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 3.3. *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 prawie tyle samo wierzchołków w obu jej częściach (różnica nie większa niż 1), 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 3.4. *Wierzchołek $v \in V(Q_n)$ jest otoczony przez $F \subseteq V(Q_n)$ gdy $N(v) \subseteq F$ (F zawiera wszystkich sąsiadów v).*

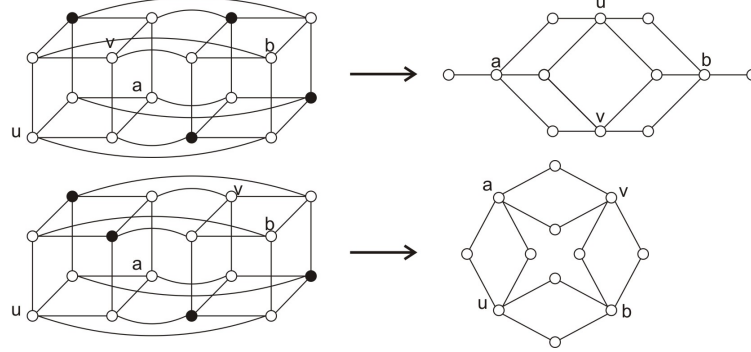
Definicja 3.5. *Dla $u, v \in V(Q_n), F \subseteq V(Q_n)$ trójka (u, v, F) jest zablokowana w Q_n gdy u jest otoczony przez $\{v\} \cup F$ lub v jest otoczony przez $\{u\} \cup F$.*

3.2. Długie ścieżki

Twierdzenie 3.6. *Dla Q_n i $F \subseteq V(Q_n)$, takich, że $2 \leq n \leq 5$ 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.

Na wyróżnienie zasługują przypadki negatywne, które z dokładnością do automorfizmów kostki są dokładnie 2. Rozpatruje je na obrazkach z zaznaczonymi wierzchołkami i uproszczoną ich wersją z pominiętymi wierzchołkami wadliwymi.



Jak widać w obu przypadkach najdłuższa możliwa uv -ścieżka ma długość $4 < 16 - 2 \cdot 4 - 2$ \square

Twierdzenie 3.7. *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 [5])

Dowód oparty jest na indukcji po wymiarze. Podstawę indukcji stanowi twierdzenie 3.6. 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 podgrafy indukowane 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).

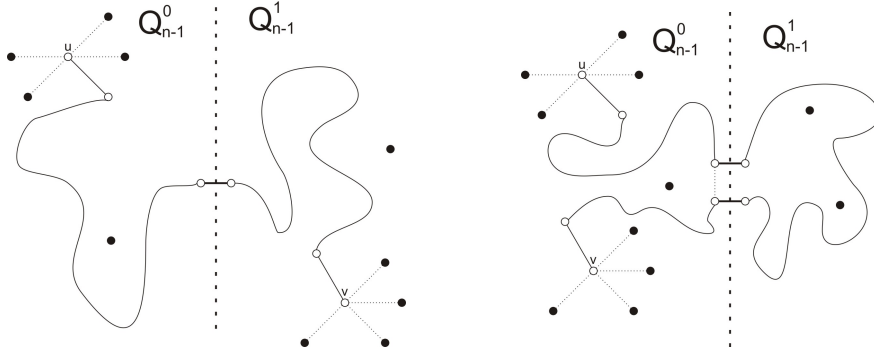
Dla $|F| = 2n - 4$ można rozpatrzyć macierz $|F| \times n$, w której w wierszach wypisane są ciągi binarne wszystkich wierzchołków wadliwych. Trzeba wybrać taką kolumnę, w której zarówno 0 jak i 1 jest co najmniej po 2. Gdyby nie dało się dokonać takiego wyboru oznaczałoby to, że w każdej kolumnie jest 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 macierz 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 $|F| \leq 2n - 5$ (nie otaczającego żadnego wierzchołka) 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 (rozbiecie 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.

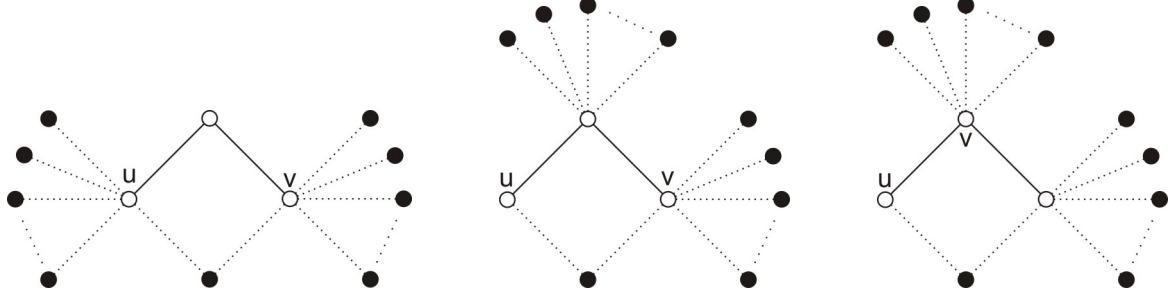


□

Uwaga 3.8. Dla Q_n , $F \subseteq V(Q_n)$, $|F| = 2n - 3$ teza twierdzenia 3.7 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 :



□

3.3. Długość cykle

Definicja 3.9. 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 3.10. 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 [6] i został zmodyfikowany do tej postaci w pracy [5] aby lepiej pasować do dowodu poniższego twierdzenia.

Twierdzenie 3.11. 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 [5])

Na podstawie lematu 3.10 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 niewadliwe 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 3.7 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 $\lceil \frac{n^2}{2|F| - n - 2} \rceil \geq 5$ umożliwiając ominięcie złych przypadków z twierdzenia 3.6. □

3.4. Długie pary ścieżek

Lemat 3.12. *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 3.7, gdzie ze względu na rozmiar F trójka (u, v, F) nie może być zablokowana. \square

Twierdzenie 3.13. *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 hiperkostki. 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 [7])

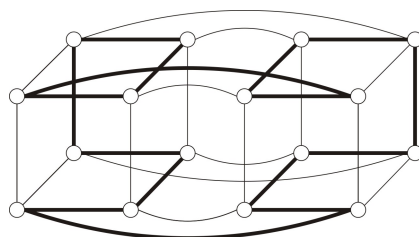
Dowód podobnie jak inne przebiega indukcyjnie – małe przypadki ($n \leq 5$) można sprawdzić ręcznie (tutaj trochę więcej sprawdzania niż w poprzednich dowodach), dla większych łatwo jest podzielić Q_n na dwie Q_{n-1} tak, żeby każda z nich miała nie więcej niż $n - 4$ wierzchołków wadliwych. Dalej rozpatrywane 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 3.7. \square

Uwaga 3.14. *Jeśli $A = \{u, w\}$, $B = \{v, w\}$, to jedna ze ścieżek musi mieć długość 0 i być zaczepiona w wierzchołku w . Twierdzenie 3.13 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 3.7 (możliwe jest to tylko dlatego, że u i v nie należą do jednej części dwudzielnej Q_n).*

Rozdział 4

Ścieżka i cykl Hamiltona

4.1. Kostka bez wadliwych wierzchołków



Pełna kostka Q_n ($n \geq 2$) jest grafem hamiltonowskim (posiadającym cykl Hamiltona) – bardzo wiele różnych cykli Hamiltona można uzyskać już łącząc dwa cykle z podkostek Q_{n-1} (wystarczy, że w dwóch takich cyklach jest taka sama krawędź). Cykle hamiltona na hiperkostce mają również zastosowania praktyczne. Jednym z przykładów użycia takich cykli są kody Graya – ciągi wszystkich wektorów binarnych określonej długości (po dokładnie raz), których sąsiednie wyrazy różnią się na dokładnie jednym bicie. Kody te znajdują zastosowanie między innymi przy efektywnej iteracji, minimalizacji kodowania i obwodów binarnych, oraz w algorytmach genetycznych i kodach korekcyjnych.

Pojawia się pytanie, czy po usunięciu kilku wierzchołków nadal da się zbudować takie cykle lub chociaż ścieżki Hamiltona.

4.2. Ogólne podejście do problemu

Sprawdzenie czy graf posiada cykl Hamiltona jest problemem NP–zupełnym, dlatego przy dowolnym wyborze zbioru wierzchołków usuniętych ciężko spodziewać się algorytmu w czasie $O(2^n)$, a co dopiero w $O(n^c)$. Jeśli jednak zbiór usuniętych wierzchołków jest odpowiednio mały lub tworzy graf z jakiejś wąskiej klasy, to istnieją dowody posiadania takiego cyklu przez wadliwą hiperkostkę.

Definicja 4.1. *Graf dwudzielny nazywamy zbalansowanym jeśli jego dwie części dwudzielne mają ten sam rozmiar, zaś prawie zbalansowanym jeśli ich rozmiary różnią się o 1.*

Hiperkostka jest grafem dwudzielnym, więc i po usunięciu części wierzchołków takim pozostanie. Jako że dowolny cykl w takim grafie odwiedza tyle samo wierzchołków w obu częściach dwudzielnych (z krotnościami), to podstawowym warunkiem koniecznym jest to, żeby z obu tych części usunąć dokładnie tyle samo wierzchołków. W przypadku ścieżki Hamiltona koniecznym jest żeby graf był zbalansowany lub prawie zbalansowany.

Definicja 4.2. Graf dwudzielny (z częściami U i V) nazywamy hamiltonowsko wiązalnym (Hamiltonian laceable) jeśli zachodzi jeden z warunków:

- jest zbalansowany i dla każdej pary $u \in U, v \in V$ istnieje ścieżka Hamiltona z u do v .
- jest prawie zbalansowany i dla każdej pary $u, u' \in U$ (gdzie U to ta większa składowa) istnieje ścieżka Hamiltona z u do u' .

Jeśli graf jest zbalansowany i hamiltonowsko wiązalny, to istnieje w nim cykl Hamiltona (jeśli ma więcej niż 2 wierzchołki) – wystarczy jako u i v wybrać wierzchołki połączone krawędzią. Hiperkostka jest grafem hamiltonowsko wiązalnym.

W 2001 Stephen Locke postawił hipotezę:

Przypuszczenie 4.3. Niech $F \subseteq V(Q_n)$ będzie zbiorem po k wierzchołków z obu dwudzielnych składowych Q_n . Wtedy $Q_n - F$ posiada cykl hamiltona jeśli $n \geq k + 2$.

Od tego czasu udowodnione zostały szczególne przypadki tej hipotezy ($k \leq 3, k \leq \frac{n-5}{6}, \dots$), a w 2009 podane zostało ich uzupełnienie – dowód indukcyjny po n w pracy [8], który jednak po kilka przypadków granicznych odsyła do drugiej pracy, którą autorzy mają dopiero opublikować.

4.3. Szkice dowodów dla szczególnych przypadków

Definicja 4.4. Podgraf H grafu G nazywamy izometrycznym jeśli $d_H(u, v) = d_G(u, v)$ dla każdych $u, v \in V(H)$.

Łatwo zauważyć, że podgraf izometryczny jest zawsze podgrafem indukowanym przez podzbiór wierzchołków, implikacja w drugą stronę jednak nie zachodzi (na przykład P_4 jako podgraf C_5).

4.3.1. podkostka

Fakt 4.5. Jeśli jako $F \subseteq V(Q_n)$, $n \geq 3$ weźmiemy zbiór wierzchołków podkostki Q_m ($1 \leq m < n$), to $Q_n - F$ posiada cykl Hamiltona.

Dowód. Dla $m \geq 2$:

Porządkany cykl Hamiltona w $Q_n - F$ można otrzymać poprzez podział grafu na $2^{n-m} - 1$ kostek Q_m , wybranie tego samego cyklu Hamiltona dla tych podkostek a następnie połączenie ich w następujący sposób:

Wybieramy jedną z podkostek, która wraz z usuniętą tworzy kostkę Q_{m+1} – taka kostka różni się od poprzedniej na dokładnie jednej współrzędnej (takiej która jest stała na całej podkostce). Łączymy obie i otrzymujemy w ten sposób wadliwą kostkę Q_{m+1} z cyklem Hamiltona. Pozostałe kostki łączymy parami, razem z cyklami (wystarczy wybrać taką samą krawędź w obu i dokonać "przebiecia") – wszystkie tak samo. Cykl z wadliwej hiperkostki ma tylko jedną krawędź, której nie ma w innych cyklach – można dokonać połączenia wadliwej Q_{m+1} z całą i dalej aż do otrzymania $Q_n - F$ z cyklem Hamiltona.

Dla $m = 1$ postępujemy analogicznie, jednak po pierwszym kroku otrzymana $Q_2 - Q_1$ nie ma cyklu a jedynie ścieżkę Hamiltona. $Q_3 - Q_1$ posiada już jednak cykl Hamiltona który można przedłużyć do cyklu Hamiltona na Q_3 zawierający tę krawędź z $Q_2 - Q_1$. Dalej można postępować już tak samo – zawsze da się wybrać krawędź według której można dokonać "przebiecia". \square

Podobnym sposobem można otrzymać cykle dla F równego sumie kilku kostek ($m \geq 2$), które nie są połączone (tak długo, jak jest ich na tyle mało, aby dało się wybierać krawędzie do "przepięcia" – to da się sprawdzić algorytmem wielomianowym od n i F).

Poniżej przedstawię szkice dowodów twierdzeń dla innych szczególnych przypadków F opisanych w pracy [9]. Ze względu na licznosc mało istotnie różniących się przypadków po dokładne obrazki i opisy wyborów wierzchołków, które łączymy ścieżkami Hamiltona w podkostkach odsyłam do tejże pracy.

4.3.2. Ścieżka izometryczna

Fakt 4.6. *W ścieżce izometrycznej w Q_n każda z krawędzi ma inny kierunek, a więc ścieżka taka może mieć długość co najwyżej n .*

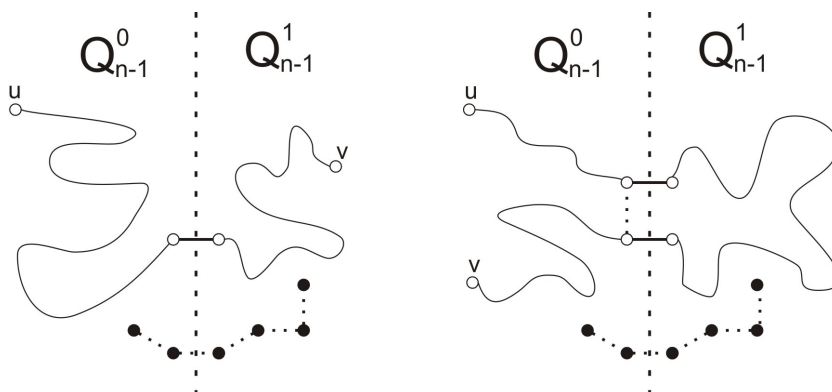
Dowód. Gdyby dwie krawędzie miały ten sam kierunek, to w Q_n można by było połączyć początek i koniec ścieżki inną ścieżką, krótszą o 2 (krawędzi o tych samych kierunkach, oprócz tych dwóch powtarzających się). \square

Twierdzenie 4.7. *Dla $F \subseteq V(Q_n)$ indukującego ścieżkę izometryczną nieparzystej długości $Q_n - F$ jest hamiltonowsko wiązalna jeśli $n \geq 4$.*

Dowód. Dla ścieżek długości większej niż 1 zakładamy indukcyjnie hipotezę dla mniejszych kostek. Dzielimy hiperkostkę na dwie względem kierunku drugiej krawędzi na ścieżce. W każdej podkostce znajduje się część ścieżki, która jest izometryczną nieparzystą ścieżką, więc obydwie wadliwe podkostki są zbalansowane.

Problem rozbijamy na dwa przypadki – w pierwszym wierzchołki u i v , które chcemy połączyć ścieżką Hamiltona znajdują się w różnych podkostkach. Wystarczy wtedy wybrać parę wierzchołków odpowiadających sobie w podkostkach, o tych własnościach, że żaden nie jest wadliwy, są różne od u i v , oraz są w innych dwuspójnych częściach niż u lub v z ich podkostki. Ze względu na ilość wierzchołków zawsze da się znaleźć taką parę, a z warunku indukcyjnego istnieją odpowiednie ścieżki Hamiltona w podkostkach.

W przypadku gdy u i v wylądują w jednej podkostce, można je w niej połączyć ścieżką Hamiltona i w odpowiednim miejscu (wzdłuż dowolnej krawędzi ścieżki Hamiltona, która istnieje w drugiej wadliwej podkostce) dopiąć ścieżkę Hamiltona z drugiej podkostki.



Dla ścieżki długości 1 dowód przebiega tak samo, z tym wyjątkiem, że w jednej podkostce ląduje cała ścieżka, zaś druga jest wolna od wad. Przypadek $n = 4$ jest natomiast rozważany ręcznie (tam są możliwe tylko 2 takie ścieżki z dokładnością do automorfizmów kostki). \square

Twierdzenie 4.8. Dla $F \subseteq V(Q_n)$ indukującego ścieżkę izometryczną parzystej długości $Q_n - F$ jest hamiltonowsko wiązalna jeśli $n \geq 5$.

Dowód. Ponownie indukcyjnie po n , tym razem dzieląc kostkę w kierunku pierwszej krawędzi ścieżki – zostawiając w pierwszej podkostce pojedynczy wadliwy wierzchołek, zaś w drugiej kostce wymiaru ≥ 4 wadliwą ścieżkę nieparzystej długości (przypadek z poprzedniego twierdzenia). Tym razem należy rozpatrzyć 3 przypadki (gdy u i u' są w różnych kostkach i 2 przypadki, gdy są w tej samej). Dla każdego z tych przypadków ponownie łatwo wybrać odpowiednie ścieżki w podkostkach i połączyć je tak aby uzyskać ścieżki Hamiltonowskie w $Q_n - F$ od u do u' (szczegóły z obrazkami można zobaczyć w pracy [9]). \square

Uwaga 4.9. Na podstawie [10] prawdziwe jest mocniejsze stwierdzenie:

Niech P będzie dowolną ścieżką długości $\leq 2n - 4$ w grafie Q_n , wtedy $Q_n - P$ jest hamiltonowsko wiązalna.

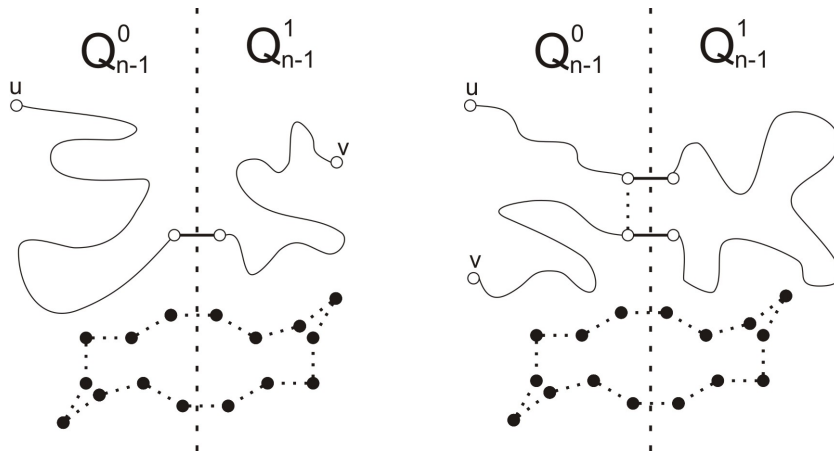
4.3.3. Cykl izometryczny

Uwaga 4.10. W cyklu izometrycznym na hiperkostce każdy kierunek krawędzi występuje dokładnie 2 razy, w antypodycznych krawędziach cyklu.

Dowód. Każdy kierunek krawędzi musi się pojawić na dowolnym cyklu w hiperkostce parzystą ilość razy. Dla cyklu długości $2k$ weźmy dwa wystąpienia tego samego kierunku, oddalone od siebie na cyklu o minimalną liczbę krawędzi. Dostajemy ścieżkę długości l zaczynającą i kończącą się krawędzią tego samego kierunku. Podobnie jak w dowodzie faktu 4.6 mamy w hiperkostce inną ścieżkę długości $l - 2$ łączącą te same końce. Z izometryczności $l - 2 \geq \min(l, 2k - l)$, zaś z minimalności $l - 2 \leq 2k - l \Rightarrow l - 1 = k$. \square

Twierdzenie 4.11. Niech C będzie izometrycznym cyklem w Q_n ($n \geq 6$), o długości $2k$ podzielnej przez 4, wtedy $Q_n - C$ jest hamiltonowsko wiązalna.

Dowód. Zaczynamy od podzielenia hiperkostki na dwie w kierunku wyznaczonym przez pierwszą (dowolnie wybraną) krawędź cyklu. W obu kostkach znajduje się wadliwa izometryczna ścieżka długości nieparzystej, a więc z twierdzenia 4.7 obie są hamiltonowsko wiązalne. Podobnie jak wcześniej wystarczy rozważyć dwa przypadki w zależności od tego czy u i v wpadają do różnych, czy do tej samej podkostki. W obu przypadkach parę wierzchołków do połączenia ścieżek, czy krawędź do podpięcia cyklu wybieramy analogicznie jak w dowodzie twierdzenia 4.7.



\square

4.3.4. Drzewo izometryczne

Uwaga 4.12. W drzewie izometrycznym w Q_n każda z krawędzi ma inny kierunek.

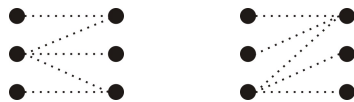
Dowód. Załóżmy, że istnieją dwie krawędzie w tym samym kierunku. Wynika z tego, że w drzewie jest ścieżka, która zawiera obie krawędzie, co daje sprzeczność z lematem 4.6 (w drzewie każde dwa wierzchołki są połączone tylko jedną ścieżką, więc nie ma innej krótszej). \square

Uwaga 4.13. Drzewo jest grafem dwudzielnym o częściach U i V . Jeśli niepuste drzewo jest zbalansowane, lub prawie zbalansowane z przewagą w V , to do V należy pewien jego liść.

Dowód. Nie wprost – jeśli w drzewie każdy liść należy do dwudzielnej części U , to można wybrać dowolny wierzchołek $v \in V$ jako korzeń. Usunięcie dowolnej gałęzi bez rozwidleń (od ostatniego rozwidlenia do liścia) nie produkuje nowych liści, może też pogorszyć balans jedynie zwiększając przewagę V . Gdy zostanie już tylko jedna ścieżka, to kończyć się ona będzie z obu stron wierzchołkami z U , co daje sprzeczność. \square

Lemat 4.14. Dla zbalansowanego drzewa T izometrycznego w Q_5 graf $Q_5 - T$ jest hamiltonowsko wiązalny.

Lemat można sprawdzić ręcznie lub komputerowo (są tylko dwa takie drzewa nie będące ścieżkami z dokładnością do umiejscowienia w kostce).



Twierdzenie 4.15. Dla T zbalansowanego lub prawie zbalansowanego drzewa izometrycznego w Q_n graf $Q_n - T$ jest hamiltonowsko wiązalny dla $n \geq 5$ (w przypadku zbalansowanego wystarczy $n \geq 4$).

Dowód. Dowód przez indukcję po rozmiarze hiperkostki. Jako krok bazowy można traktować lemat 4.14, przypadek drzewa prawie zbalansowanego (w Q_5 istnieje takie tylko jedno nie będące ścieżką) i zbalansowanego dla $n = 4$ (tu są tylko ścieżki) można rozpatrzyć analogicznie do tamtego. W kroku indukcyjnym dla drzewa zbalansowanego korzystamy z wymiaru o 1 mniejszego i drzewa prawie zbalansowanego zaś dla drzewa prawie zbalansowanego korzystamy z drzewa zbalansowanego w kostce o mniejszym wymiarze. Na podstawie uwagi 4.13 wybieramy liść z części dwudzielnej w której drzewo ma co najmniej połowę wierzchołków i dzielimy mniejsze według kierunku krawędzi drzewa wchodzącej do tego wierzchołka. Otrzymujemy w ten sposób w jednej podkostce pojedynczy wadliwy wierzchołek (jak w dowodach poprzednich twierdzeń) oraz w drugiej izometryczne drzewo zbalansowane lub prawie zbalansowane.

Jak we wcześniejszych dowodach rozpatrujemy przypadki w zależności od tego w których podkostkach znajdują się wierzchołki które chcemy połączyć ścieżką Hamiltona, tworzymy odpowiednie ścieżki w podkostkach i łączymy je. \square

Rozdział 5

Maksymalna ścieżka indukowana

Definicja 5.1. Średnicą grafu nazywamy największą odległość pomiędzy jego dwoma wierzchołkami $d(G) = \max_{u,v \in V(G)} d(u,v)$.

Jedną z najważniejszych zalet hiperkostki Q_n , jest to, że pomimo względnie niskiej gęstości jej średnica jest niewielka i równa n , a do tego po usunięciu części węzłów często pozostaje spójna. Graf ten posiada również tę przyjemną własność, że przy losowym usuwaniu wierzchołków odległości pomiędzy nieusuniętymi wierzchołkami rzadko się zwiększają (w szczególności dla wierzchołków bardziej oddalonych). Przy intencjonalnym usuwaniu aby zwiększyć odległość pomiędzy dowolnymi dwoma wierzchołkami potrzeba usunąć co najmniej dwa wierzchołki (w przypadku dowolnych wierzchołków oddalonych o 2), zaś aby zwiększyć średnicę grafu należy usunąć co najmniej $n - 1$ wierzchołków (wszystkich prócz jednego sąsiadów dowolnie wybranego wierzchołka). Aby uzyskać średnicę $n + 2$ należy usunąć $2n - 3$ wierzchołków (na przykład sąsiedzi wierzchołków numer 1 i 2 oprócz 4 i siebie nawzajem przy numerowaniu klasycznym), dalej jednak pojawia się coraz więcej możliwości usuwania wierzchołków i ciężko wybrać najlepszą.

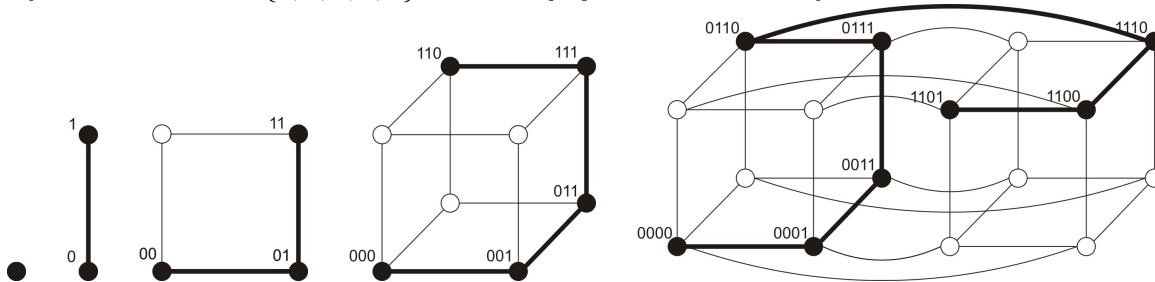
Zainspirowany tym postawiłem sobie pytanie jaką największą średnicę można uzyskać w ten sposób pozostawiając graf spójnym. Średnicę spójnego grafu $Q_n - F$ wyznacza najkrótsza ścieżka między pewnymi dwoma wierzchołkami, jeśli więc dla jakiegoś grafu uda się uzyskać taką maksymalną średnicę, to można wybrać tę ścieżkę i odrzucić wszystkie wierzchołki spoza niej (nie zmniejszając średnicy). Oznacza, to że to maksimum po wszystkich grafach $Q_n - F$ jest uzyskiwane dla grafu będącego ścieżką. Ścieżka taka ma tę własność, że ciągi binarne sąsiadujących w niej wierzchołków różnią się na dokładnie jednej pozycji, natomiast dla dowolnych dwóch nie będących sąsiadami ciągi te muszą się różnić na co najmniej dwóch pozycjach. Wyjściowy problem został więc zredukowany (okazał się równoważny) do problemu:

Znajdź najdłuższy ciąg wektorów binarnych długości n , że każde dwa sąsiednie wektory różnią się na co najmniej jednej pozycji, zaś każde dwa inne na co najmniej dwóch.

5.1. Ścieżka Fibonacciego

Twierdzenie 5.2. Dla danego n istnieje prosty algorytm konstruujący ciąg wektorów reprezentujący ścieżkę długości F_{n+1} , (gdzie F_n to n -ty wyraz ciągu Fibonacciego: $F_0 = F_1 = 1$), która jest podgrafem indukowanym hiperkostki. Co więcej prefiks długości F_n jest takim ciągiem dla $n - 1$ (po odrzuceniu ostatniej współrzędnej wektorów).

Dowód. Na potrzeby dowodu nazwę taką ścieżkę S_n . Dowód tradycyjnie przez indukcję po wymiarze – dla $n \in \{0, 1, 2, 3, 4\}$ konstrukcja jest widoczna na rysunkach:



Dla tak małych wymiarów jest to jedyny z dokładnością do wyboru punktu początkowego i przepermutowania współrzędnych sposób uzyskania ścieżki tej długości. Już na tych przypadkach widać, że ścieżka tak powstała wygląda identycznie od początku i od końca (przy zamianie punktów początkowego i końcowego oraz przepermutowaniu współrzędnych). Dodatkowo widać też, że tylko 3 podkostki Q_{n-2} zawierają wybrane do ścieżki wierzchołki, dokładniej pierwsza podkostka zawiera S_{n-2} , druga S_{n-3} i trzecia znów S_{n-2} (znów "z dokładnością").

Konstrukcja indukcyjna S_n przebiega następująco:

Startuję od S_{n-1} , który nie używa tylko współrzędne od 0 do $n-2$, kolejnym wierzchołkiem w ciągu będzie sąsiad aktualnie ostatniego w kierunku $n-1$. Mam dane już ponad pół kostki, jeśli więc chcę uzyskać S_n takie samo od początku jak i od końca wystarczy znaleźć odpowiednią permutację. Co do tej nowej części mamy pewną dowolność przy wyborze permutacji, jednak musi się ona zgadzać na tej środkowej części (S_{n-2} zaczynające się po krawędzi w kierunku $n-3$, a kończące przed dostawioną $n-2$ – te współrzędne zostaną ze sobą zamienione w permutacji i w całej ścieżce S_n krawędzie w tych kierunkach występują tylko po razie). Z kroku indukcyjnego S_{n-2} jest symetryczne w ten sposób, więc istnieje odpowiednia permutacja na kierunkach w niej używanych, wystarczy więc ją rozszerzyć o tą zamianę $n-1$ z $n-2$ (i formalnie dowolnie uzupełnić o kierunki w niej nie występujące, choć tak na prawdę takich nie ma). Z konstrukcji wynika, że zbudowana tak ścieżka posiada własność: dowolne dwa wierzchołki ze ścieżki są sąsiadami w hiperkostce tylko wtedy gdy są sąsiadami na ścieżce wewnątrz trzech swoich części. Pomiedzy pierwszą i drugą częścią własność ta jest spełniona z kroku indukcyjnego, zaś pomiedzy trzecią i drugą jest tak samo z symetrii. Pozostają części pierwsza i trzecia, ale skoro są one całkowicie zanurzone w dwóch podkostkach, które nie mają między sobą krawędzi, to warunek ten jest również spełniony w tym przypadku. \square

Wniosek 5.3. *Jako, że ciąg Fibonacciego rośnie w tempie wykładniczym – ilość użytych wierzchołków wynosi $\frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^{n+1} - (\frac{1-\sqrt{5}}{2})^{n+1}) \approx \frac{(1.62)^{n+1}}{\sqrt{5}} = \Omega(1.61^n)$, przy wielkości kostki 2^n .*

5.2. Wyniki eksperymentalne

Uwaga 5.4. *Dla wymiarów większych niż 4 da się otrzymać ścieżki jeszcze dłuższe – dla $n=5$ najdłuższa możliwą ścieżką jest:*

00000 00001 00011 00111 01111 01110 01100 11100 11101 11001 11011 11010 10010 10110.
Ma ona długość 13, jest więc o 1 dłuższa od ścieżki otrzymanej na podstawie twierdzenia 5.2

Aby znaleźć możliwie najdłuższe ścieżki dla większych n przeprowadziłem eksperymenty przy pomocy komputera. Wyszukiwanie ścieżek prowadziłem startując od wierzchołka $\bar{0}$ i

dobierając do istniejącej ścieżki nowy wierzchołek, który jest sąsiadem aktualnego końca i nie jest sąsiadem żadnego poprzedniego, aż do momentu, gdy taki wybór nie jest możliwy – w ten sposób otrzymuje "grę". Znalezienie wszystkich możliwych ścieżek otrzymywalnych w ten sposób (a są to wszystkie takie poprawne ścieżki, które zaczynają się w wierzchołku $\bar{0}$) otrzymywane jest poprzez przejrzenie drzewa gry, której wierzchołkami są ścieżki, zaś ścieżki w przodkach tego wierzchołka są jej prefiksami. Drzewo tej gry ma maksymalny stopień rozgałęzienia równy n , zaś głębokość $|P|$.

Daje to złożoność algorytmu przeglądającego wszystkie ścieżki mocno pesymistycznie ograniczoną przez $O(n^{|P|})$. Nawet po zastosowaniu różnych usprawnień, algorytm dokładny szybko staje się zbyt wolny, dlatego dla $n \geq 7$ musiałem skorzystać z metod losowych.

5.2.1. NMCS

Aby otrzymać pewną ścieżkę, która posiada tę własność, że nie da się jej już wydłużyć do innej poprawnej można przeprowadzić zejście po drzewie od korzenia do liścia poprzez losowe wybieranie jednego z dzieci za każdym razem, gdy taki wybór jest możliwy.

Przeprowadzając odpowiednio dużo takich losowych zejść po drzewie i wybierając najdłuższą ścieżkę ze znalezionych możemy uzyskać już w miarę dobry wynik, istnieją jednak metody, które pozwalają na lepsze ukierunkowanie losowości.

Jedną z takich metod jest przeszukiwanie Monte Carlo polegające na tym, że mając dany stan gry (aktualną ścieżkę) dla każdego możliwego ruchu (wyboru kolejnego wierzchołka) szacujemy jego wartość poprzez uruchomienie przeszukiwania losowego w tym kierunku i następnie wybranie najlepszego z nich. Metoda ta jest wykorzystywana głównie w przeypadkach gier w tym takich, w których gracz ma tylko częściowy wpływ na rozgrywkę (istnieje przeciwnik), jednak w przypadku szukania możliwie długiej ścieżki również się sprawdza.

Rozszerzenie tej metody zostało zaprezentowane w pracy [11] zwane Nested Monte Carlo Search, polegające na tym, że do oszacowania wartości możliwych ruchów zamiast zwykłych przeszukiwań losowych używamy zwykłego przeszukiwania Monte Carlo. Dokładniej metoda ta zakłada, że tworząc rozwiązanie przy pomocy NMCS z poziomem k używamy do wyboru ruchu NMCS z poziomem $k - 1$ i tak dalej rekurencyjnie aż do poziomu 0, który oznacza przeszukiwanie losowe. Można to przedstawić następującym pseudokodem:

```

NMCS( $v, level$ ){
  while( $TRUE$ ){
     $val.push\_back(v)$ ;
    if( $|v.children| == 0$ ) return  $val$ ;
    if( $level == 0$ ){
       $v = random(v.children)$ ;
    }else{
       $best = -1$ ;
      foreach( $u \in v.children$ ){
         $val2 = NMCS(u, level - 1)$ ;
        if( $value(val2) > best$ ){
           $best = value(val2)$ ;
           $move = u$ ;
        }
      }
       $v = u$ ;
    }
  }
}

```

}
}

5.2.2. Uwagi praktyczne

Ze względu na ogromną różnicę w złożoności algorytmu *NMCS* w zależności od użytego poziomu można wprowadzić lekką modyfikację – w przypadku, gdy wyższy poziom jest zbyt kosztowny, zaś niższy za słaby zamiast uruchamiać program z niższym poziomem wiele razy można poszerzyć przeszukiwanie. Poszerzenie takie można łatwo uzyskać uruchamiając dla każdego dziecka symulację poziomu niżej nie raz a T razy (plus uruchomienie algorytmu najwyższego poziomu tyle razy).

Mając na uwadze algorytm z tą modyfikacją można oszacować pesymistyczną złożoność algorytmu dla szukania długiej izolowanej ścieżki w hiperkostce w zależności od wymiaru n , poziomu L , liczby razy przy sprawdzaniu dzieci T oraz oczekiwanej długości maksymalnej ścieżki $|P|$. Złożoność ta jest wyrażona poprzez równanie rekurencyjne:

$$NMCS(n, 0, T, |P|) = n \cdot T \cdot |P|$$

$$NMCS(n, L, T, |P|) = n \cdot T \cdot NMCS(n, L - 1, T, |P| - 1) + NMCS(n, L, T, |P| - 1)$$

Które daje rozwiązanie:

$$NMCS(n, L, T, |P|) = O(T^{L+1} \cdot n^{L+1} \cdot |P|^{L+1})$$

W przypadku szukania izolowanej ścieżki w hiperkostce należy jeszcze doliczyć czas i pamięć potrzebne na sprawdzanie czy nowo wybrany (wylosowany) wierzchołek nie jest sąsiadem jednego z wierzchołków z początku ścieżki.

Jedną z możliwych metod jest za każdym razem przeiterowanie po aktualnej ścieżce i porównanie ciągów binarnych, daje to jednak złożoność czasową rzędu $n|P|$. Drugą metodą jest zapamiętywanie wszystkich sąsiadów wierzchołków ze ścieżki (plus ich samych) w tablicy binarnej rozmiaru 2^n . W tym wypadku sprawdzanie jest trywialnie proste (w czasie potrzebnym na odczytanie nazwy wierzchołka), jednak złożoność pamięciowa w przypadku NMCS poziomu 0 przeważa nad złożonością czasową samego algorytmu. W przypadku małych n nie jest to jednak złe rozwiązanie, ponieważ ze względu na liczne optymalizacje tablic binarnych jest to najszybsze rozwiązanie. Trzecią (pośrednią) metodą jest użycie jako struktury drzewa prefikсового lub tablicy hashującej uzyskując podobnie do tablicy binarnej złożoność $O(n)$ na odczytanie i wstawienie wierzchołka jednocześnie potrzebując jedynie $O(n^{1.5} \cdot |P|)$ lub $O(n \cdot |P|)$ pamięci w zależności od użytej struktury.

W przypadku użycia struktury zapamiętującej sąsiadów wierzchołków ze ścieżki należy jeszcze zwrócić uwagę na zmienianie jej stanu przy wchodzeniu i wracaniu z niższych poziomów wywołania. Można albo za każdym razem pamiętać wszystkie wstawione wierzchołki i przy zwracaniu wyniku do poziomu wyżej dokonywać ich usunięcia, albo można za każdym razem kopiować całą strukturę (dosyć efektywne przy użyciu tablic). W drugim przypadku wystarczy mieć $L + 1$ kopii struktury na raz (po jednej na każdy poziom) i przy powrocie z poziomu niżej zastępować starą wersję z tamtego poziomu kopią z poziomu o 1 wyżej.

Mając to na uwadze można zaprogramować algorytm tak, aby miał złożoności czasową $O(T^{L+1} \cdot n^{L+2} \cdot |P|^{L+1})$ i pamięciową $O(L \cdot n \cdot |P|)$.

Przeszukiwanie można dodatkowo zawęzić jeśli program weźmie pod uwagę, że kierunki, które nie pojawiły się jeszcze na ścieżce są równoważne (wystarczy pamiętać liczbę oznaczającą ilość kierunków już użytych i nie używać kierunków większych niż ta liczba + 1). W ten sposób ustalone są pierwsze 3 ruchy ($1 \rightarrow 2 \rightarrow 4 \rightarrow 8$ przy numerowaniu klasycznym), zaś na czwarty są tylko dwie możliwości ($8 \rightarrow 7, 8 \rightarrow 17$). Daje to duże przyśpieszenie przy małych przypadkach, jednak przy większych nie ma już większego znaczenia.

5.2.3. Uzyskane wartości

Przy pomocy komputera i przedstawionych wyżej algorytmów przeprowadziłem przeszukiwanie w celu znalezienia możliwie najdłuższej takiej ścieżki. W przypadku $n \leq 6$ przy pomocy pełnego przeszukiwania, a dalej NMCS z 4 poziomami dla $n = 7, 3$ dla $n = 8, 2$ dla $9 \leq n \leq 12$ i 1 dla $13 \leq n \leq 18$ z ilością wywołań niższego poziomu takich żeby podobny był czas dla wszystkich n . Dalej użyłem zwykłego błędzenia losowego i wybierania najdłuższej uzyskanej tak ścieżki (ilość przeszukiwań znów dostosowana do tego czasu). Program użyty do takiego przeszukiwania wraz z zapisami ścieżek dla mniejszych n (dla większych brak ze względu na wielkość plików) i dokładnymi parametrami jest dołączony do pracy.

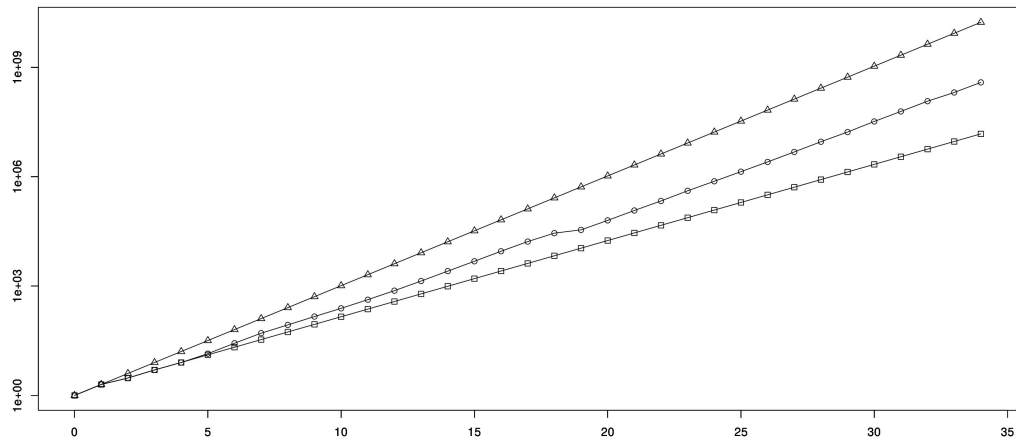
Poniższa tabelka i wykres logarytmiczny prezentują porównanie najlepszych rezultatów uzyskanych dla poszczególnych n ($|P|$) do wielkości ścieżki Fibonacciego i całego grafu.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
F_{n+1}	1	2	3	5	8	13	21	34	55	89	144	233	377	610
$ P + 1$	1	2	3	5	8	14	27	51	86	146	245	423	749	1373
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192
$\sqrt[n]{ P + 1}$		2	1.73	1.71	1.68	1.7	1.73	1.75	1.75	1.74	1.73	1.73	1.74	1.74

n	14	15	16	17	18	19	20	21	22
F_{n+1}	987	1597	2584	4181	6765	10946	17711	28657	46368
$ P + 1$	2568	4778	9017	16612	28287	34801	63271	118344	216033
2^n	16384	32768	65536	131072	262144	524288	1048576	2097152	4194304
$\sqrt[n]{ P + 1}$	1.75	1.76	1.77	1.77	1.77	1.73	1.74	1.74	1.75

n	23	24	25	26	27	28	29
F_{n+1}	75025	121393	196418	317811	514229	832040	1346269
$ P + 1$	410719	748502	1373384	2543441	4774783	9098482	16788856
2^n	8388608	16777216	33554432	67108864	134217728	268435456	536870912
$\sqrt[n]{ P + 1}$	1.75	1.76	1.76	1.76	1.77	1.77	1.77

n	30	31	32	33	34
F_{n+1}	2178309	3524578	5702887	9227465	14930352
$ P + 1$	32747927	61637291	117676035	204031449	386051791
2^n	1073741824	2147483648	4294967296	8589934592	17179869184
$\sqrt[n]{ P + 1}$	1.78	1.78	1.79	1.79	1.79



5.3. Znane ograniczenia długości węży

Problemem najdłuższej ścieżki indukowanej i najdłuższego cyklu induowanego w hiperkostce zajmowali się już wcześniej inni badacze, nazywając problemy odpowiednio "wąż w pudełku" (snake-in-a-box) i "zwój w pudełku" (coil-in-a-box) osiągając lepsze rezultaty. Dla $n \leq 7$ wyniki otrzymane przeze mnie są optymalne, dla dalszych są znane lepsze (8-98,9-190,10-370,11-708,12-1357,13-2687). Znane są również ograniczenia dolne i górne lepsze od tych oczywistych, czy opisanych przeze mnie w poprzednich podrozdziałach.

5.3.1. Ograniczenia dolne

Od kiedy zdefiniowany został problem pojawiły się coraz lepsze ograniczenia dolne ($c \cdot 2^{\frac{n}{2}}, (1.5)^n, c \cdot 6^{\frac{n}{4}}, c \cdot (2.5)^{\frac{n}{2}}$), później długo dominowało ograniczenie z pracy [12] Dające ograniczenie $\frac{7}{4} \frac{2^n}{n-1}$ (które jest już lepsze od mojego wynikającego z konstrukcji ścieżek Fibonacciego). Jeszcze lepsze ograniczenie wraz z konstrukcją ścieżki i cyklu pojawia się w pracy [13] (przedstawię szkic tego rozwiązania poniżej).

Fakt 5.5. *Zwój w pudełku można przerobić na węża w pudełku poprzez usunięcie dowolnego wierzchołka tracąc 2 na długości.*

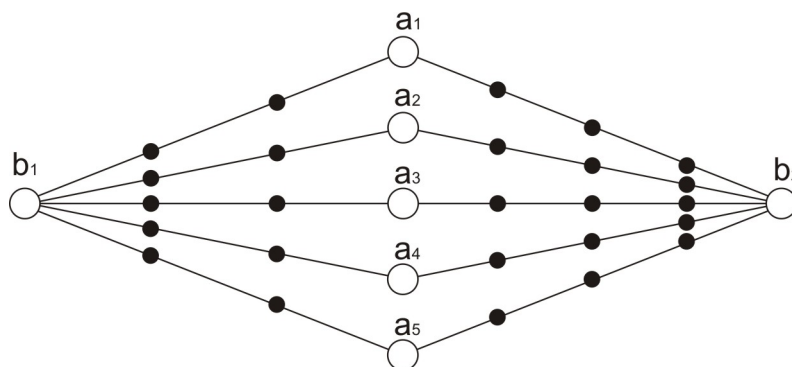
Twierdzenie 5.6. *Dla Q_n , $n \geq 7$ można skonstruować zwój w pudełku o długości $\frac{9}{64} \cdot 2^n$*

Dowód. (szkic konstrukcji i dowodu z pracy [13]) Q_n dzielimy na kostki wymiaru 5 otrzymując graf $Q_{n-5} \times Q_5$. W grafie Q_{n-5} bierzemy cykl Hamiltona otrzymany indukcyjnie: dla $n-5=2$ jedyny możliwy $((0,0) \rightarrow (0,1) \rightarrow (1,1) \rightarrow (1,0) \rightarrow (0,0))$, dalej dla otrzymania cyklu dla Q_{m+1} bierzemy dwie kostki Q_m z takimi cyklami i łączymy je usuwając krawędź, która wchodzi do wierzchołka $\bar{0}$ i łącząc te dwie ścieżki w jeden cykl w Q_{m+1} (konstrukcja jak w rozdziale 4). Kostki Q_5 występujące na tym cyklu można ponumerować startując od $\bar{0}$ w kierunku $(0, \dots, 0, 1)$.

Kolejnym krokiem jest zdefiniowanie ciągu 2^{n-5} ścieżek indukowanych w Q_5 (ponumerowanych według cyklu Hamiltona) spełniających następujące własności:

- ścieżka m kończy się tam, gdzie zaczyna się ścieżka $m+1 \pmod{2^{n-5}}$.
- ścieżki z dwóch kostek Q_5 będących sąsiadami w Q_{n-5} , ale nie na cyklu Hamiltona nie zawierają wspólnych wierzchołków.
- ścieżki o numerach $m = 1$ lub $2 \pmod{4}$ mają długość 3, zaś o numerach $m = 0$ lub $3 \pmod{4}$ długość 4.

Ścieżki konstruowane są poprzez chodzenie po grafie G (widocznym na obrazku) zanurzonym w Q_5 :



Zanurzenie jest zadane dla ścieżki $b_1, \dots, a_1, \dots, b_2$ wraz z pośrednimi wierzchołkami poprzez ścieżkę $(0, 0, 0, 0, 0), (0, 0, 0, 0, 1), (0, 0, 0, 1, 1), (0, 1, 0, 1, 1), (0, 1, 0, 1, 0), (0, 1, 1, 1, 0), (1, 1, 1, 1, 0), (1, 1, 1, 1, 1)$, zaś dla pozostałych czterech analogicznie po cyklicznej zamianie współrzędnych. Ścieżkami długości 3 są odpowiedniki $a_i \rightarrow b_1$ i $b_1 \rightarrow a_j$, zaś ścieżek długości 4 ścieżki $a_j \rightarrow b_2$ i $b_2 \rightarrow a_i$. Kolejne ścieżki tworzone są poprzez chodzenie w ten sposób po grafie G z odpowiednim wyborem jednej z 5 ścieżek kiedy startujemy w b_i . Postępując w ten sposób otrzymujemy porządkany zwój w pudełku (dowód drugiego warunku można znaleźć w pracy [13]). Po połączeniu wszystkich tych ścieżek otrzymujemy cykl długości $(\frac{3+4}{2} + 1) \cdot 2^{n-5} = \frac{9}{64} \cdot 2^n$. \square

Wniosek 5.7. Długość węża w pudełku Q_n jest ograniczona z dołu przez $\frac{9}{64} \cdot 2^n - 2$.

5.3.2. Ograniczenia górne

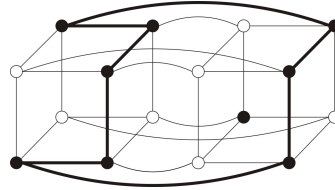
Definicja 5.8. Pseudowężem w pudełku nazwiemy dowolny indukowany podgraf hiperkostki, w którym stopień każdego wierzchołka jest nie większy niż 2.

Każdy wąż i każdy zwój jest pseudowężem (pseudowąż składa się ze zbioru węży, zwojów i wierzchołków izolowanych).

Fakt 5.9. Obcięcie pseudowęża do części z jednej podkostki jest pseudowężem w tej podkostce (w całej hiperkostce też).

Twierdzenie 5.10. dla $n \geq 5$ każdy dowolny pseudowąż (a więc i wąż) w Q_n ma rozmiar co najwyżej 2^{n-1} .

Dowód. (Dowód za [12]). Dla $n = 4$ jest tylko jeden z dokładnością do automorfizmów kostki pseudowąż rozmiaru 9



Gdyby w Q_5 był pseudowąż rozmiaru 17, to przy dowolnym podziale na dwie pseudokostki Q_4 w jednej z nich musiałby się znajdować pseudowąż rozmiaru co najmniej 9, a więc ten z rysunku. W drugiej kostce wierzchołki odpowiadające tym z cyklu nie mogą zostać wybrane, czyli w skład węża musiałby wchodzić wszystkie pozostałe, jednak taki graf nie byłby pseudowężem.

Dalej dla $n \geq 6$ indukcyjnie. Przy dowolnym podziale na dwie podkostki Q_{n-1} w jednej z nich musiałby się znajdować pseudowąż rozmiaru co najmniej $2^{n-2} + 1$, co daje sprzeczność. \square

Uwaga 5.11. W pracy [14] pojawiło się mocniejsze ograniczenie górne na długość węża w pudełku: $2^{n-1} \cdot (1 - \frac{1}{n^2-n+2})$ (dla $n \geq 7$), które zostało później jeszcze ulepszone do $2^{n-1} \cdot (1 - \frac{1}{20n-41})$ (dla $n \geq 12$) w pracy [15].

Bibliografia

- [1] L. Lovász, J. Pelikán 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] Frank Harary, John P. Hayes and Horng-Jyh Wu, "A survey of the theory of hypercube graphs" *Comput. Math. Applic.* Vol. 15, No 4, pp. 277-289, 1988
- [4] Tomáš Dvořák, Jiří Fink, Petr Gregor, Václav Koubek and Tomasz Radzik, "Efficient connectivity testing of hypercubic networks with faults"
- [5] Jiří Fink and Petr Gregor, "Long paths and cycles in hypercubes with faulty vertices"
- [6] 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.*
- [7] Jiří Fink and Petr Gregor, "Long pairs of paths in faulty hypercubes"
- [8] Nelson Castañeda and Ivan S. Gotchev, "Proof of Locke's Conjecture, I"
- [9] David Pěgřimek, "Hamiltonian cycles in hypercubes with removed vertices", *Charles University, Prague 2013*
- [10] Sun, Chao-Ming and Jou, Yue-Dar, "Hamiltonian laceability of hypercubes with some faulty elements", *In Proceedings of 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan, March 26-29, 2009, pp.626-630*
- [11] Tristan Cazenave, "Nested Monte-Carlo Search" *LAMSADE Universite Paris-Dauphine Paris, France*
- [12] Ludwig Danzer and Victor Klee, "Lengths of Snakes in Boxes", *Journal of combinatorial theory* 2, 258-265 (1967)
- [13] J. Wojciechowski, "A new lower bound for Snake-in-the-Box Codes", *Combinatorica* 9 (1) (1989) 91–99
- [14] F. I. Solov'jeva, "Upper bound for the length of a cycle in an n-dimensional unit cube", *Methods of Diskrete Analiz* 45 (1987)
- [15] Hunter S. Snevily, "The snake-in-the-box problem: A new upper bound", *Discrete Mathematics* 133 (1994) 307-314

- [16] Frank Harary, Marilynn Livingston, "Independent domination in hypercubes" *Appl. Math. Lett.* Vol. 6, No 3, pp. 27-28, 1993
- [17] Wojciech Rytter & Bartosz Szreder, "Wprowadzenie do kombinatoryki algorytmicznej"
- [18] Donald E. Knuth, "Generating All Tuples and Permutations" *The Art of Computer Programming volume 4, fascicle 2*
- [19] Frank Ruskey, "Combinatorial Generation" *Working Version, October 1, 2003*