

ZAKŁAD SYSTEMÓW INFORMACYJNYCH
WYDZIAŁ INFORMATYKI I ZARZĄDZANIA
POLITECHNIKA WROCŁAWSKA

Raport Serii SPR nr 31

**Grupowanie dokumentów hipertekstowych
na podstawie drzewa maksymalnych
przepływów**
(praca doktorska)

Przemysław Kazienko

Promotor:
prof. dr hab. inż. Czesław Daniłowicz

Słowa kluczowe: grupowanie dokumentów,
hipertekst, maksymalny
przepływ, drzewo maksy-
malnych przepływów,
system WWW

Pragnę złożyć podziękowania:

Promotorowi Panu prof. dr. hab. inż. Czesławowi Daniłowiczowi za zaangażowanie, bardzo cenne uwagi oraz kilka zużytych ołówków.

Dr. Janowi Kraszewskiemu za pomoc w formułowaniu definicji i twierdzeń.

Moim współpracownikom z Zakładu Systemów Informacyjnych, bez których niemożliwe byłoby przeprowadzenie eksperymentów.

Małgorzacie Micule, która dokonała korekty.

Mojej żonie i czwórce moich dzieci, które wykazały się niezwykłą cierpliwością i wyrozumiałością. Bez ich pomocy niniejsza praca by nie powstała.

SPIS TREŚCI

WAŻNIEJSZE OZNACZENIA	3
WPROWADZENIE	6
1. WYBRANE ELEMENTY TEORII GRAFÓW I TEORII MNOGOŚCI	13
1.1. Grafy	13
1.2. Maksymalny przepływ	16
1.2.1. Drzewo maksymalnych przepływów	18
1.2.2. Tworzenie drzewa maksymalnych przepływów. Algorytm Gusfielda	20
1.3. Hierarchia	22
2. STRUKTURA HIPERTEKSTU	25
2.1. Elementy struktury hipertekstu	25
2.2. Struktura systemów hipertekstowych	27
2.3. Klasyfikacja elementów struktury	28
2.3.1. Rodzaje stron WWW	29
2.3.2. Rola odsyłaczy hipertekstowych	29
2.3.3. Typy odsyłaczy	30
2.4. Środowisko hipertekstowe i kolekcja hipertekstowa	31
3. PRZEGLĄD METOD GRUPOWANIA DOKUMENTÓW HIPERTEKSTOWYCH	35
3.1. Funkcja podobieństwa dokumentów	35
3.1.1. Podobieństwo strukturalne	36
3.1.2. Podobieństwo treściowe	38
3.1.3. Inne nośniki podobieństwa. Podobieństwo złożone	39
3.1.4. Podsumowanie	42
3.2. Metody i algorytmy grupowania. Wymagania obliczeniowe i pamięciowe	44
3.3. Podział metod grupowania dokumentów	45
3.4. Metody niehierarchiczne	46
3.4.1. Metody jednego przebiegu	46
3.4.2. Metody ponownego przydziału	47
3.4.3. Metoda „rozrzucić i zbierać”	47
3.5. Metody hierarchiczne	48
3.5.1. Ogólny algorytm dla metod hierarchicznych	49
3.5.2. Metoda najbliższego sąsiedztwa	51
3.5.3. Metoda najdalszego sąsiedztwa	52
3.5.4. Metoda średniej grupowej	52
3.5.5. Metoda Warda	53
3.5.6. Metoda centroidalna i medianowa	54
3.6. Grafowe metody grupowania dokumentów hipertekstowych	54
3.6.1. Metoda k -krawędziowych składowych	54
3.6.2. Metoda k -klanów	56
3.7. Niegrafowe metody grupowania dokumentów hipertekstowych	57
3.7.1. Metoda STC drzewa fraz	57
3.7.2. Metoda k -wycinków	58
3.7.3. Metoda koncentratorów i autorytetów	59
3.7.4. Inne metody grupowania dokumentów hipertekstowych	61

4. PODOBIENSTWO DOKUMENTÓW HIPERTEKSTOWYCH.....	62
4.1. Założenia dla funkcji podobieństwa	62
4.2. Pierwotne podobieństwo strukturalne.....	62
4.3. Własności pierwotnego podobieństwa strukturalnego.....	70
4.4. Zmodyfikowane podobieństwo strukturalne.....	72
4.5. Podobieństwo treściowe.....	74
4.6. Przekształcenia podobieństwa treściowego uwzględniające specyfikę stron WWW.....	75
4.7. Podobieństwo kompleksowe.....	78
5. METODA GRUPOWANIA DOKUMENTÓW HIPERTEKSTOWYCH NA PODSTAWIE DRZEWA MAKSYMALNYCH PRZEPŁYWÓW	81
5.1. Ogólny opis metody grupowania	81
5.2. Graf ważony	83
5.3. Wydzielenie składowych grafu.....	87
5.4. Drzewo maksymalnych przepływów	88
5.5. Skalowanie wag krawędzi drzewa	89
5.5.1. Parametry procesu skalowania.....	90
5.5.2. Przykłady skalowania	92
5.6. Hierarchia grup	93
5.7. Algorytm TC grupowania drzewa maksymalnych przepływów.....	100
5.8. Własności algorytmu TC	104
5.8.1. Zgodność z definicją hierarchii.....	105
5.8.2. Niezależność hierarchii od drzewa maksymalnych przepływów.....	106
5.8.3. Rozkład dokumentów w hierarchii	113
5.9. Złożoność obliczeniowa metody grupowania.....	115
5.10. Reprezentacja grup dokumentów.....	115
5.10.1. Wskaźniki ważności.....	119
5.10.2. Wyznaczanie reprezentacji grupy dokumentów	120
6. EKSPERYMENTALNA WERYFIKACJA METODY GRUPOWANIA	123
6.1. Program eksperymentów.....	123
6.2. System CIS.....	126
6.3. Eksperymenty związane z wyborem wartości parametrów	130
6.3.1. Maksymalna liczba poziomów max^{lev}	130
6.3.2. Próg istotności podobieństwa τ	132
6.3.3. Parametr μ regulujący znaczenie odsyłaczy pośrednich.....	136
6.3.4. Parametry α, β, γ regulujące znaczenie terminów z tytułu, opisu i słów kluczowych...	139
6.4. Oceny hierarchii dokonane przez ekspertów	145
6.5. Omówienie wyników	154
7. PODSUMOWANIE	157
LITERATURA	160
ZAŁĄCZNIK A. ALGORYTM GUSFIELDA	169
ZAŁĄCZNIK B. LISTA PYTAŃ TESTOWYCH	176
ZAŁĄCZNIK C. WYPEŁNIONY FORMULARZ OCENY HIERARCHII	177

WAŻNIEJSZE OZNACZENIA

$c, c_i^{T_1}$	– poziom związania węzłów w elemencie hierarchii, na i -tym poziomie hierarchii powstałej z drzewa T_1
$cv_{t,Z}, cv_{t,Z}^C$	– wskaźnik ważności terminu t w dokumentach zbioru Z należących do elementu (Z,c,e) hierarchii; dla dopełnienia grupy poszerzonej $EC^C(Z,c,e)$
D, D^E, D^-	– zbiór dokumentów tekstowych należących do kolekcji hipertekstowej (kolekcja dokumentów do grupowania); należących do środowiska hipertekstowego; zbiór dokumentów nieistniejących
DP	– liczba dokumentów przeniesionych
E^G, E^S, E^T	– zbiór krawędzi grafu ważonego dla hipertekstu; składowej grafu ważonego; drzewa maksymalnych przepływów
$EC(Z,c,e)$	– grupa poszerzona dla elementu (Z,c,e) hierarchii
$EC^C(Z,c,e)$	– dopełnienie grupy poszerzonej $EC(Z,c,e)$
e	– numer elementu pustego hierarchii
e_{ij}^G, e_{ij}^T	– krawędź łącząca węzły v_i i v_j w grafie ważonym; w drzewie maksymalnych przepływów (także drzewie minimalnych przekrojów)
$F(Z,c,e)$	– zbiór liści, będących potomkami elementu (Z,c,e) hierarchii
$f_{x,y}, f'_{x,y}$	– wartość maksymalnego przepływu z węzła x do węzła y ; przeskalowana wartość maksymalnego przepływu
$G(V,E^G), G^S(V^S,E^S), G_i^S(V_i^S,E_i^S)$	– graf ważony; składowa grafu ważonego; i -ta składowa grafu ważonego
$H(V^S), H(T)$	– zbiór elementów hierarchii na zbiorze V^S ; hierarchii dla drzewa maksymalnych przepływów $T(V^S,E^T)$
$(H(V^S), \leq_H), (H(T), \leq_{H(T)})$	– hierarchia na zbiorze V^S ; hierarchia dla drzewa maksymalnych przepływów $T(V^S,E^T)$
$HT(D,L), HTE(D^E,L^E)$	– kolekcja hipertekstowa; środowisko hipertekstowe
K_i	– i -ty poziom hierarchii
L, L^E	– zbiór odsyłaczy kolekcji hipertekstowej; środowiska hipertekstowego
L_{ij}^s, L_{ij}^n	– zbiór odsyłaczy semantycznych; nawigacyjnych z d_i do d_j
L_i^s, L_i^n	– zbiór odsyłaczy semantycznych; nawigacyjnych z d_i do dokumentów z D
$L_i^{\rightarrow}$	– zbiór odsyłaczy wychodzących z dokumentu d_i ,
$L_{ij}^{\Rightarrow}, L_{ij}^{n\Rightarrow}, L_{ij}^{s\Rightarrow}$	– zbiór wszystkich; nawigacyjnych; semantycznych, pośrednich odsyłaczy cytowania z dokumentu d_i , względem dokumentu d_j
$L_{ij}^{\Leftarrow}, L_{ij}^{n\Leftarrow}, L_{ij}^{s\Leftarrow}$	– zbiór wszystkich; nawigacyjnych; semantycznych, pośrednich odsyłaczy współcytowania dokumentu d_i względem dokumentu d_j
$L_{ij}^{\Leftrightarrow}$	– zbiór odsyłaczy wychodzących z tych dokumentów, z których wychodzą także pośrednie odsyłacze współcytowania dokumentu d_i względem dokumentu d_j :
$L_{ij}^{r,t}$	– r -ty odsyłacz z dokumentu d_i do d_j typu t
M	– liczba wszystkich, różnych składowych w grafie $G(V,E^G)$
$\max^f, \min^f$	– największa; najmniejsza wartość maksymalnego przepływu
$\max^{lev}$	– liczba różnych wartości przeskalowanego, maksymalnego przepływu

N	– liczba dokumentów do grupowania; także stopień grafu (liczba jego węzłów)
$\mathbf{N}$	– zbiór liczb naturalnych, łącznie z $\{0\}$
$\mathbf{N}_+$	– zbiór dodatnich liczb naturalnych
NG	– liczba nowych grup utworzonych przez eksperta
nav	– rodzaj odsyłacza - nawigacyjny
nd_i	– liczba dokumentów z kolekcji, w których występuje termin i
nl_i, nl_i^n, nl_i^s	– liczba wszystkich; nawigacyjnych; semantycznych odsyłaczy wychodzących z dokumentu d_i i prowadzących do innych dokumentów ze zbioru D
$nl_{ij}, nl_{ij}^n, nl_{ij}^s$	– względna liczba wszystkich; nawigacyjnych; semantycznych odsyłaczy z dokumentu d_i do d_j
$nl_i^{\rightarrow}$	– liczba odsyłaczy wychodzących z dokumentu d_i
$nl_{ij}^{\Rightarrow}, nl_{ij}^{n\Rightarrow}, nl_{ij}^{s\Rightarrow}$	– względna liczba wszystkich; nawigacyjnych; semantycznych, pośrednich odsyłaczy cytowania z dokumentu d_i względem dokumentu d_j
$nl_{ij}^{\Leftarrow}, nl_{ij}^{n\Leftarrow}, nl_{ij}^{s\Leftarrow}$	– względna liczba wszystkich; nawigacyjnych; semantycznych, pośrednich odsyłaczy współcytowania dokumentu d_i względem dokumentu d_j
$nl_{ij}^{\Leftrightarrow}$	– liczba odsyłaczy wychodzących z tych dokumentów, z których wychodzą także pośrednie odsyłacze współcytowania dokumentu d_i , względem dokumentu d_j .
nl_{max}	– maksymalna liczba odsyłaczy łączących dwa dokumenty w zbiorze D
nt, nt_j	– liczba wszystkich, różnych terminów z dokumentów ze zbioru D ; różnych terminów zawartych w dokumencie d_j
$P(V^S)$	– zbiór wszystkich podzbiorów zbioru V^S
$p(v), p(v_1, v_2)$	– poziom węzła v w hierarchii; najniższy wspólny poziom dwóch węzłów v_1, v_2 w hierarchii
$\mathbf{R}$	– zbiór liczb rzeczywistych
$R(Z, c, e)$	– reprezentacja elementu (Z, c, e) hierarchii
r	– kolejny numer odsyłacza między dwoma dokumentami
$S(d_i, d_j), S^c(d_i, d_j), S^{cc}(d_i, d_j), S^{cs}(d_i, d_j), S^{ps}(d_i, d_j), S^s(d_i, d_j)$	– podobieństwo kompleksowe; treściowe; zmodyfikowane treściowe; zmodyfikowane strukturalne; pierwotne strukturalne; strukturalne między dokumentem d_i oraz d_j
sem	– rodzaj odsyłacza - znaczeniowy
$\acute{S}PDP, \acute{S}PDK$	– średnie przesunięcie dokumentów przesuniętych; dokumentów kolekcji
T^L	– zbiór typów (rodzajów) odsyłaczy
$T(V^S, E^T), T'(V^S, E^T)$	– drzewo maksymalnych przepływów; minimalnych przekrojów
t	– typ odsyłacza; także termin o numerze t
$tf_{d,j,k}, tf'_{d,j,k}$	– liczba wystąpień (częstość); zmodyfikowana częstość terminu k w dokumencie d_j
$tf_{d,j,k}^{BODY}, tf_{d,j,k}^{DESCRIPTION}, tf_{d,j,k}^{KEYWORDS}, tf_{d,j,k}^{TITLE}$	– częstość występowania (liczba wystąpień) terminu k w treści zasadniczej; opisie; słowach kluczowych; tytule dokumentu d_j
$tf'_{D,t}, tf'_{EC(Z,c,e),t}, tf'_{EC^C(Z,c,e),t}$	– zmodyfikowana częstość występowania terminu t w dokumentach z kolekcji D ; w grupie rozszerzonej $EC(Z,c,e)$; w dopełnieniu grupy rozszerzonej $EC^C(Z,c,e)$
U, U'	– liczba grup powstałych w procesie grupowania; po ocenie przez eksperta
V, V^S, V_i^S	– zbiór węzłów grafu ważonego; składowej grafu ważonego; i -tej składowej grafu ważonego
v_i	– węzeł drzewa lub grafu o numerze i

WP	– suma wartości wszystkich przesunięć
w_{ij}^G, w_{ij}^T	– waga nad krawędzią e_{ij}^G grafu ważonego; drzewa maksymalnych przepływów
$w_{d,k}^f, w_{d,k}^r$	– waga terminu; zmodyfikowana waga terminu k w dokumencie d_i
Z	– zbiór węzłów należących do jednego elementu hierarchii
ZG	– liczba grup zlikwidowanych przez eksperta
$(Z, c, e), (Z_0, c_0, e_0)$	– element; korzeń hierarchii
$\%DNP$	– procent dokumentów nieprzeniesionych
$\%NG, \%ZG$	– procent nowych; zlikwidowanych grup
$\leq_H, \leq_{H(T)}$	– relacja porządkująca elementy hierarchii; hierarchii dla drzewa $T(V^S, E^T)$
α, β, γ	– współczynnik zwiększenia ważności terminu pochodzącego z tytułu; opisu; słów kluczowych
φ	– funkcja odwzorowująca jedną hierarchię na drugą
λ	– współczynnik ważności odsyłaczy nawigacyjnych
τ	– próg istotności podobieństwa dokumentów

WPROWADZENIE

Hipertekst jest koncepcją organizacji zbioru dokumentów, wykorzystującą powiązania między dokumentami. Nowatorską cechą systemów hipertekstowych, w porównaniu do systemów liniowych, jest istnienie odsyłaczy łączących poszczególne elementy – węzły systemu. Dzięki nim autorzy mogą przenosić związki semantyczne między dokumentami bezpośrednio do struktury tworzonego systemu. Użytkownicy, natomiast, wybierając niektóre z odsyłaczy podczas korzystania z systemu (przeglądania), niejako dostosowują owe, zawarte w strukturze związki, do swoich indywidualnych potrzeb. Po przeczytaniu jednego dokumentu, użytkownik sam wybiera kolejny dokument, który chciałby oglądać, do którego chciałby „przejsć”. Odsyłacze umieszczane są wprost w tekście dokumentu (w kontekście) i umożliwiają łączenie pojedynczych wątków tematycznych danego dokumentu z całymi, innymi dokumentami.

Idea hipertekstu została po raz pierwszy przedstawiona przez Vannevara Busha w 1945 roku [Bus45]. Jednakże dynamiczny rozwój koncepcji hipertekstu nastąpił dopiero pod koniec lat osiemdziesiątych i w latach dziewięćdziesiątych. Opracowano wtedy wiele modeli hipertekstu, jak np. Dexter [Hal94], HDM [Gar93], Proxhy [Kac91], Microcosm [Hal93], czy XHMB [Pau98].

Szczególnym przykładem systemu hipertekstowego jest system WWW (*World Wide Web*), który wraz z pocztą elektroniczną, jest najważniejszą usługą dostępną w sieci Internet¹. Ze względu na swój dynamiczny rozwój i coraz większe znaczenie gospodarcze i społeczne (jest ważnym elementem kształtowania się społeczeństwa informacyjnego), system WWW stanowi obszar wielu badań prowadzonych na całym świecie.

Do początków lat dziewięćdziesiątych badania w zakresie hipertekstu ograniczały się głównie do modeli teoretycznych i prototypowych systemów autorskich. Obecnie można obserwować olbrzymie zainteresowanie problemami, które wiążą się bezpośrednio z WWW. Dotyczy to zwłaszcza wyszukiwania i wydobywania informacji (ang. *data mining, knowledge discovery*), komunikacji użytkownika z systemem i prezentacji danych (ang. *user interface, computer-human interaction*), modelowania zainteresowań użytkowników i „personalizacji” systemu WWW (ang. *user profile, user preference, user modeling*) oraz zarządzania udostępnianą informacją.

WWW jest obecnie najważniejszym systemem opartym na koncepcji hipertekstu. Precyzyjnie należałoby go nazwać systemem hipermedialnym, gdyż w dokumentach WWW występują także inne niż tekst media. Składa się on z milionów dokumentów, zwanych stronami (ang. *pages, web pages*), tworzących ogromną sieć powiązań², z której korzystają miliony użytkowników³. W środowisku WWW są zawarte wszystkie najważniejsze elementy hipertekstu. W związku z tym, rozwiązania problemów w nim występujących, można stosować do całej klasy systemów hipertekstowych.

¹ Z badań przedstawionych w pracy [GVU98] wynika, że poczta elektroniczna jest niezbędną technologią dla 93.3% użytkowników Internetu, zaś WWW dla 90.6%.

² Koehler na początku 1998r. szacował liczbę stron WWW na 500 mln [Koh99], zaś Lawrence i Giles na 320 mln. [Law98]. Rok później w czasopiśmie *Nature* opublikowano informację o 800 mln. stron [Law99b]. Według badań firmy Inktomi przeprowadzonych wraz z NEC Research Institute w sieci WWW jest obecnie ponad 1 miliard stron, rozlokowanych na prawie 6.5 miliona serwerów [Ink00]. Liczba wzajemnych powiązań jest także bardzo duża. Brin i Page w pracy [Bri98], analizując 26 mln stron, pochodzących z wyszukiwarki Google, stwierdzili istnienie na nich 518 mln odsyłaczy. Zatem na stronę przypada średnio prawie 20 odsyłaczy hipertekstowych.

³ Liczbę użytkowników Internetu na świecie, w połowie 1999r., szacowano na ponad 170 mln. [Har99], zaś obecnie w Polsce na ok. 2 mln [Świ00] a nawet wg badań OBOP na 5 mln. [Dur00].

Jednym z kluczowych problemów występujących w systemach hipertekstowych jest wyszukiwanie informacji. Dotyczy to także środowiska WWW, w którym bardzo wielu użytkowników sprawia ono trudności¹.

Z innymi trudnościami borykają się twórcy (autorzy i konserwatorzy) otwartych systemów hipertekstowych. Muszą oni poradzić sobie ze zmiennością zasobów informacyjnych swoich systemów oraz tych, z którymi ich systemy są powiązane. Wynika z tego ciągła potrzeba uaktualniania i konserwacji.

Kolejny problem, związany z hipertekstem, to prezentacja struktury informacji. W przypadku dużych systemów nie jest możliwe proste zobrazowanie całej sieci powiązań. Wiąże się z tym także – często pojawiający się w literaturze – problem poczucia zagubienia w hiperprzestrzeni (ang. *lost in hyperspace*), jakie ma użytkownik, nie wiedząc właściwie, w którym miejscu systemu się aktualnie znajduje.

Można więc wyróżnić trzy podstawowe funkcje, które są źródłem problemów, występujących w systemach hipertekstowych (wg [Kaz98b]):

- a) Wyszukiwanie informacji².
- b) Zarządzanie informacją.
- c) Prezentacja użytkownikom i autorom struktury strony oraz całego serwisu informacyjnego.

Problemy te nabierają szczególnego znaczenia w dynamicznie rozwijającym się, otwartym systemie WWW, ze względu na:

- a) Dużą zmienność w czasie wynikającą ze: zmiany nazw (serwerów, katalogów, plików); przesunięć dokumentów do innych katalogów na serwerze i między serwerami; braku synchronizacji między stronami lustrzanymi (ang. *mirror pages*), tzn. kopiami strony na różnych serwerach; usuwaniem dokumentów; podziałem i połączeniami stron; zmianami tytułów i treści stron.
- b) Złożoność struktury, w tym także dużą liczbę połączeń między poszczególnymi serwisami informacyjnymi.
- c) Dużą liczbę dokumentów. Nie jest np. możliwe umieszczenie w jednym dokumencie wszystkich odsyłaczy do stron relewantnych, gdyż stron relewantnych może być bardzo dużo.
- d) Powielanie informacji. Te same strony (lustrzane) występują jednocześnie w wielu miejscach (czasami jedynie z nieznacznie zmienioną zawartością), tzn. na różnych serwerach (ang. *mirror site*). Kopiowane są one zwykle w celu przyspieszenia transferu danych dla użytkowników znajdujących się na różnych kontynentach lub powstają w wyniku reorganizacji i przenoszenia serwisów w inne miejsca. Zdarza się także, że te same strony znajdują się w różnych kartotekach tego samego serwera. Są to np. kopie zapasowe lub kolejne wersje tej samej strony.

¹ Z badań przeprowadzonych przez Graphic, Visualization, & Usability Center's (GVU) na dziesiątkach tysięcy użytkowników systemu WWW [Pit96, Gvu98] wynika, że najczęściej zastrzeżeń mają oni do szybkości pracy systemu (pierwsze miejsce wśród wszystkich, zasygnalizowanych kłopotów; zgłaszane przez 65% badanych użytkowników). Drugie miejsce (58% użytkowników) zajmuje problem niemożności połączenia się z konkretnymi stronami, np. w wyniku wyłączenia lub niesprawności serwerów czy elementów sieci Internet, nieaktualnych odsyłaczy, itp. Trzecie miejsce (47%) wiąże się z wyszukiwaniem nowych stron WWW. Problem odnajdywania znanych stron, o których użytkownik wie, że istnieją, ale nie potrafi ich zlokalizować, zgłaszało 28% respondentów (czwarte miejsce).

² Przegląd problemów wyszukiwania informacji w sieci WWW zamieszczono w pracach [Kaz97, Dan00b].

Wartość informacyjna serwisów WWW jest bardzo różna, ze względu na to, że jest to system otwarty, w którym każdy może umieszczać swoje informacje. Jednakże w przeważającej większości serwisy WWW zawierają informacje cenne dla pewnej grupy użytkowników, w tym także informacje związane z nauką i dydaktyką¹:

„Sieć WWW zawiera różnorodny zakres materiałów naukowych, włączając w to serwisy informacyjne naukowców, prowadzonych projektów, uniwersytetów, preprinty, raporty techniczne, artykuły z czasopism i konferencji, materiały dydaktyczne a także bazy danych (np. sekwencje genów, struktury molekularne, biblioteki obrazów). Wiele z tych materiałów nie jest udostępnianych w tradycyjnych bazach danych. Artykuły naukowe, udostępniane w tradycyjnych bazach danych, często wcześniej są osiągalne w sieci (naukowcy często umieszczają preprinty publikacji na swoich stronach domowych).”^{2,3}

W celu rozwiązania problemu wyszukiwania informacji w systemie WWW, powstały tzw. wyszukiwarki internetowe (ang. *search engines*). Stosują one macierze równoległe połączonych, szybkich komputerów, posiadających szerokopasmowe łącza sieciowe⁴. Pojedyncze wyszukiwarki obejmują do 40% wszystkich stron całego systemu WWW [Bha98a, Sul00].

Inną próbą rozwiązania problemu wyszukiwania informacji, dla systemu WWW są katalogi tematyczne, które jednak posiadają zaindeksowany o 1-2 rzędy mniejszy zasób stron [Kaz97].

Jak wynika z badań statystycznych, użytkownicy systemów informacyjnych, przy wyszukiwaniu, podają zwykle małą liczbę słów kluczowych⁵. Mała liczba słów kluczowych w pytaniu, przy zaindeksowanych setkach milionów dokumentów, prowadzi zwykle do tysięcy stron zwracanych przez wyszukiwarki. Użytkownik nie jest oczywiście w stanie przejrzeć całej listy dokumentów z odpowiedzi w poszukiwaniu tych najbardziej relewantnych. W związku z tym, konieczne jest usystematyzowanie odpowiedzi wyszukiwarek przed ich prezentacją użytkownikowi. Można tego dokonać poprzez:

- a) Ranking dokumentów przeprowadzany na podstawie różnych kryteriów.
- b) Pogrupowanie zbioru stron z odpowiedzi wyszukiwarki i zaprezentowanie użytkownikowi grup dokumentów związanych tematycznie.
- c) „Personalizację”, tzn. uwzględnienie preferencji użytkownika.

Grupowanie⁶ (ang. *cluster analysis*) jest techniką statystyczną, używaną do generowania struktury kategorii, którym przyporządkowywane są grupy obiektów ze zbioru wejściowego. Tworzone grupy powinny charakteryzować się wysokim stopniem podobieństwa pomiędzy elementami tej

¹ Z badań przeprowadzonych przez Lawrence’a oraz Gilesa wynika, że największa część serwerów WWW, była związana z serwisami firmowymi - 83%, drugie miejsce zajęła nauka i edukacja - ok. 6% serwerów, trzecie – ochrona zdrowia - 2.8%. [Law99b].

² [Law99b, s.107], tłum. autora.

³ Większość pozycji zamieszczonych w bibliografii niniejszej pracy jest dostępna w sieci WWW. Autorowi zdarzało się także docierać do artykułów, przed ich opublikowaniem za kilka miesięcy w czasopismach lub materiałach konferencyjnych. Niektóre czasopisma udostępniają także artykuły jeszcze nie opublikowane, ale zatwierdzone do druku, np. ACM Transactions on Database Systems: <http://www.acm.org/tods/Upcoming.html>.

⁴ Opis konfiguracji sprzętowej wyszukiwarki AltaVista zawarty jest w pracy [Chak99].

⁵ Ponad połowa zadawanych przez użytkowników pytań – w badaniach przeprowadzonych przez Anicka – zawierała tylko jedno słowo, zaś ¼ kwerend posiadało mniej niż 3 słowa [Ani94]. Wg badań przeprowadzonych przez Pinkertona średnia liczba słów kluczowych w pytaniach zadawanych wyszukiwarce WebCrawler wynosiła 1.5 [Pin94]. W badaniach Crofta i Cooka, przeprowadzonych dla internetowego serwisu Thomas, 88% pytań miało mniej niż 3 słowa [Cro95]. Z analizy pytań zadawanych wyszukiwarce Altavista wynika, że średnia liczba słów w pytaniu wynosiła 2.35, zaś liczba operatorów logicznych, łączących słowa: 0.41 [Chak99].

⁶ Na określenie grupowania używa się także terminu: „analiza skupień” [Kuch82].

samej grupy i niskim stopniem podobieństwa pomiędzy obiektami różnych grup. Van Rijsbergen sformułował tzw. hipotezę grupowania (ang. *cluster hypothesis*): „mocno powiązane dokumenty mają tendencję do odpowiadania na te same potrzeby informacyjne”¹.

Niesłusznie, grupowanie nazywane jest czasami automatyczną **klasyfikacją** (ang. *automatic classification*). Różnica między klasyfikacją a grupowaniem polega na tym, że w klasyfikacji lista kategorii, którym przyporządkowywane są obiekty jest znana przed rozpoczęciem procesu przetwarzania. W przypadku grupowania, klasy są definiowane dynamicznie w trakcie procesu. Dlatego grupowanie bywa wykorzystywane, jako wstępna faza klasyfikacji służąca jedynie do rozpoznania klas istniejących w przetwarzanym zbiorze.

Grupowanie jest dobrym sposobem na tworzenie struktur w dużych, zróżnicowanych zbiorach danych. Jest ono również nazywane narzędziem do odkrywania (ang. *a tool of discovery*), ponieważ umożliwia wykrywanie wcześniej niezauważonych powiązań opartych na złożonych danych [Ras92]. W związku z tym, metody grupowania można stosować do wydobywania danych.

W dziedzinie systemów informacyjnych grupowaniu podlegają dokumenty lub ich elementy (terminy). Grupowanie terminów w oparciu o dokumenty, w których współwystępują, jest wykorzystywane przy konstruowaniu tezaursów lub poprawianiu (redefinicji) zapytań.

W celu pogrupowania obiekty w zbiorze danych, należy ustalić sposób określania stopnia podobieństwa pomiędzy nimi – miarę powiązań. Może to być miara odległości lub miara bliskości (podobieństwa).

W systemach hipertekstowych nośnikiem informacji o powiązaniach semantycznych między dokumentami są odsyłacze hipertekstowe. Jeżeli w dokumencie d_1 istnieje odsyłacz prowadzący do dokumentu d_2 , to można przypuszczać, że oba te dokumenty są do siebie podobne. Prowadzi to do sformułowania tzw. hipotezy odesłania (ang. *link hypothesis*):

„Dokumenty silnie powiązane odsyłaczami, mają tendencję do odpowiadania na te same potrzeby informacyjne.”²

Potraktowanie odsyłaczy hipertekstowych jako wyznaczających wspólne treści zawarte w dokumentach jest zbieżne z cytowaniami zamieszczanymi w publikacjach naukowych:

„Cytowania są formalnym, jasno wyrażonym połączeniem między publikacjami, które posiadają szczególne punkty wspólne.”³

„Odsyłacze w systemie WWW to rozszerzenie znanej i powszechnie stosowanej w tradycyjnych publikacjach metody odsyłania czytelnika do innych dokumentów poprzez cytowanie. (...) Odsyłacze pełnią jednak specyficzną rolę zarówno w indeksach cytowań jak i w WWW. Zwłaszcza w WWW odsyłacze często odzwierciedlają takie merytoryczne związki między dokumentami (stronami), których nie da się potwierdzić za pomocą znanych metod analizy dokumentów (np. odsyłacz do strony, na której znajduje się zdjęcie opisywanego obiektu) (...) W przypadku indeksów cytowań podobieństwo dwóch dokumentów można np. mierzyć liczbą dokumentów, w których oba są cytowane.”⁴

Odsyłacze nie są jedynym źródłem informacji o związkach między dokumentami, drugim jest tekst, będący tradycyjnym nośnikiem podobieństwa.

¹ [Rij79, s.30], tłum. autora.

² [Sal99, s.691], tłum. autora.

³ [Gar79, s.1], tłum. autora.

⁴ [Dan00a, s.1,12]

Celem niniejszej pracy było opracowanie hierarchicznej metody grupowania dokumentów hipertekstowych, w której maksymalny przepływ w nieorientowanym grafie ważonym jest kryterium umieszczania dokumentów w grupach. Funkcja podobieństwa, która określa wagi w grafie odpowiada podobieństwu kompleksowemu dokumentów, wyznaczonemu poprzez strukturę wzajemnych powiązań odsyłaczowych i podobieństwo treści tekstowej.

Kolekcja dokumentów poddawanych procesowi grupowania stanowi zazwyczaj część szerszego środowiska hipertekstowego. Jednocześnie odsyłacze wychodzące z kolekcji mogą prowadzić do dokumentów z tego szerszego środowiska, a nawet do dokumentów już lub jeszcze nie istniejących. W związku z tym niezbędne stało się precyzyjne zdefiniowanie pojęcia środowiska i kolekcji hipertekstowej.

Odsyłacze i dokumenty pełnią w hipertekście różną rolę, dlatego koniecznym było dokonanie analizy roli i rodzaju elementów tworzących hipertekst.

Chcąc przeprowadzić proces grupowania dokumentów hipertekstowych, niezbędne jest określenie miary podobieństwa między parami obiektów. W tym celu przeprowadzono analizę literaturową funkcji podobieństwa dokumentów hipertekstowych. Ponieważ wszystkie przedstawione funkcje posiadają pewne wady, zwłaszcza w kontekście zastosowania przy grupowaniu na podstawie drzewa maksymalnych przepływów, opracowano własną funkcję podobieństwa.

Odsyłacze wyznaczają związki między dokumentami, a jednocześnie treść może te związki potwierdzać, więc najlepiej połączyć oba te nośniki informacji w jedną funkcję podobieństwa. W efekcie opracowano funkcję podobieństwa kompleksowego. Jest ona złożeniem podobieństwa strukturalnego oraz podobieństwa treściowego. Pierwsze z nich – związane z hipotezą odesłania – odzwierciedla strukturę hipertekstu, gdyż opiera się na liczbie odsyłaczy łączących dokumenty. Struktura hipertekstowa ma zasadniczy wpływ na wartość podobieństwa kompleksowego. Podobieństwo treściowe, natomiast, jest wyznaczane przez częstości występowania poszczególnych terminów w dokumentach. Może ono zwiększyć podobieństwo określone przez strukturę, ale nie może go zmniejszyć. Jest to jedno z zasadniczych założeń dla opracowanej funkcji podobieństwa. Jednocześnie dokumenty hipertekstowe prócz struktury hipertekstowej posiadają strukturę treści¹, w której można wydzielić pewne elementy: tytuł, słowa kluczowe, opis zawartości. Przy opracowywaniu funkcji podobieństwa treściowego należy wykorzystać to, że te elementy zawierają specyficzne informacje dotyczące treści strony.

W pracy przedstawiono metodę grupowania na podstawie drzewa maksymalnych przepływów, w której wykorzystano podobieństwo kompleksowe. Podstawą podziału zbioru dokumentów (kryterium grupowania) jest wartość maksymalnego przepływu między parami obiektów. Im maksymalny przepływ między dwoma dokumentami jest większy, tym większe jest między nimi powiązanie. W efekcie dokumenty posiadające tę samą, wystarczająco dużą wartość maksymalnego przepływu między sobą, trafiają do tej samej grupy. Opisana metoda grupowania jest hierarchiczna. Efektem końcowym jest taka hierarchia grup dokumentów, w której obiekty należące do grupy niżej położonej w hierarchii są zawsze ściślej związane, niż dokumenty należące do grupy znajdującej się wyżej. Między dokumentami należącymi do grupy tworzącej korzeń, oczekuje się najsłabszych związków tematycznych.

Zaprezentowana metoda grupowania wykorzystująca maksymalny przepływ stanowi nowe podejście do grupowania. Poza tym tworzona w niej hierarchia jest także inna, niż wykorzystywane (w znanych z literatury) hierarchicznych metodach grupowania. Dopuszcza ona istnienie grup pustych na różnych poziomach hierarchii oraz to, że dokumenty nie muszą należeć do liści.

Maksymalny przepływ jest wyznaczany dla każdej pary węzłów (każdy węzeł odpowiada jednemu dokumentowi hipertekstowemu) w grafie, którego krawędzie ważone są podobieństwem

¹ W przypadku dokumentów WWW struktura treści jest określona przez język opisu strony HTML (*HyperText Markup Language*), którego specyfikacja jest zawarta w pracy [HTM98a].

kompleksowym, obliczanym dla pary dokumentów łączonych przez tę krawędź. Istnienie krawędzi w nieskierowanym grafie implikuje więc dodatnie podobieństwo między dokumentami. Graf ważony jest odzwierciedleniem powiązań tematycznych istniejących w systemie hipertekstowym, a maksymalny przepływ sposobem na przeniesienie tych powiązań na hierarchię grup dokumentów. Przy ważeniu krawędzi zdecydowano o zastosowaniu progu, dzięki któremu możliwe jest sterowanie postacią końcowych hierarchii.

Z grafu ważonego wydzielane są składowe a z każdej takiej składowej powstaje jedna hierarchia grup. W grafie może istnieć więcej niż jedna składowa. Efektem zastosowania proponowanej metody grupowania może być więc nie jedna hierarchia grup, ale zbiór hierarchii. Aby wyznaczyć wartość maksymalnego przepływu dla wszystkich par węzłów w składowej grafu ważonego, zastosowano drzewo maksymalnych przepływów. Drzewo to ma tę własność, że maksymalny przepływ między dowolną parą jego węzłów jest taki sam jak w grafie początkowym. W celu wyznaczenia tego drzewa, dla każdej składowej, zastosowano (znany z literatury) algorytm Gusfielda, opisany w załączniku A.

Wartości maksymalnego przepływu, będące wagami krawędzi drzewa maksymalnych przepływów, są liczbami rzeczywistymi. Z tego powodu przeskalowano wartości wag krawędzi drzewa, przenosząc je z liczb rzeczywistych w ograniczony, stały zbiór liczb naturalnych. Dzięki parametrom skalowania można wpływać na mniejsze lub większe rozbudowanie hierarchii grup, tzn. na liczbę poziomów i grup.

Z przeskalowanego drzewa wydzielane są grupy, które utworzą hierarchię, jest to dokonywane opracowanego przez autora algorytmem. W pracy zbadano także i udowodniono jego właściwości, a w związku z tym także właściwości, jakie posiadają hierarchie będące efektem końcowym procesu grupowania. Pewien zbiór dokumentów tworzy grupę w hierarchii wtedy, gdy maksymalny (przeskalowany) przepływ między dowolną parą dokumentów z tej grupy ma ściśle określoną wartość, która wyznacza siłę powiązań między dokumentami w grupie. Gdy przeskalowany maksymalny przepływ dla danej grupy jest duży, to dokumenty należące do grupy są ściśle ze sobą związane. Jeżeli grupa jest położona wyżej w hierarchii (bliżej korzenia), to wartość maksymalnego przepływu (wartość związania) jest mniejsza, niż dla grup niżej umiejscowionych.

Wartość maksymalnego przepływu ustalona dla danej grupy odnosi się także do wszystkich dokumentów należących do grup niżej położonych od danej. Oznacza to, że związanie dokumentów z danej grupy z dowolnymi dokumentami z grup podległych danej, jest co najmniej takie same jak w danej grupie.

Dla każdej grupy należy wyznaczyć jej reprezentację w postaci słów kluczowych. Dzięki tym słowom możliwe jest stosunkowo proste zastosowanie metody grupowania do redefinicji kwerendy, co powinno prowadzić do zwiększenia dokładności wyszukiwania. Reprezentacja grup opisuje także tematykę dokumentów w niej zawartych, co ułatwia użytkownikowi przeglądanie hierarchii.

W celu precyzyjnego określenia wykorzystywanych w pracy pojęć, umożliwienia rozróżnienia elementów będących wkładem autora od tych opisanych w literaturze, a także ułatwienia czytania kolejnych części, w rozdziale 1 podano definicje i niektóre podstawowe twierdzenia z teorii grafów oraz teorii mnogości.

W rozdziale 2 dokonano analizy elementów tworzących hipertekst. Zbadano role i wymieniono rodzaje dokumentów i odsyłaczy systemu WWW. Zdefiniowano także podstawowe pojęcia środowiska hipertekstowego i kolekcji hipertekstowej, które wykorzystano w dalszej części.

Opis wybranych funkcji podobieństwa dokumentów hipertekstowych znanych z literatury przedstawiono w rozdziale 3. Także w tym rozdziale zaprezentowano podział metod grupowania dokumentów oraz opisano najważniejsze metody grupowania, a w szczególności te, które były stosowane do dokumentów hipertekstowych.

W rozdziale 4 przedstawiono funkcję podobieństwa kompleksowego. Uwzględnia ona elementy charakterystyczne dla dokumentów systemu WWW oraz metody grupowania na podstawie drzewa maksymalnych przepływów. Opracowano cztery, różne wersje tej funkcji.

Nowa metoda grupowania na podstawie drzewa maksymalnych przepływów została opisana w rozdziale 5. Przedstawiono tam także opisy opracowanych algorytmów oraz własności hierarchii, będących efektem końcowym zastosowania metody. Poza tym omówiono złożoność obliczeniową metody i zaproponowano sposób wyznaczania reprezentacji grup w postaci listy słów kluczowych.

Metoda grupowania na podstawie drzewa maksymalnych przepływów została zaimplementowana w systemie CIS. Z wykorzystaniem tego systemu przeprowadzono eksperymenty, w których eksperci oceniali wygenerowane hierarchie. Wyniki eksperymentów i ich analizę zawiera rozdział 6.

W podsumowaniu przedstawiono obszary zastosowań metody grupowania dokumentów oraz kierunki dalszych prac badawczych związanych z opracowaną metodą.

W załączniku A umieszczono opis algorytmu Gusfielda, służącego do utworzenia drzew maksymalnych przepływów a także wyjaśniono jego działanie na podstawie przykładów.

Załącznik B zawiera listę kolekcji testowych wykorzystanych podczas badań eksperymentalnych, zaś załącznik C przykład wypełnionego formularza zastosowanego do dokumentowania tych badań.

Wszystkie definicje, twierdzenia, lematy i algorytmy zostały w tekście wyróżnione poprzez umieszczenie ich w ramkach.

1. WYBRANE ELEMENTY TEORII GRAFÓW I TEORII MNOGOŚCI

W celu ułatwienia czytania dalszych rozdziałów pracy oraz umożliwienia bardziej precyzyjnego rozróżnienia elementów opracowanych przez autora (od tych, znanych z literatury) przedstawiono niektóre fragmenty teorii grafów oraz teorii mnogości.

1.1. GRAFY

Przy opracowywaniu definicji z zakresu teorii grafów korzystano z prac Kulikowskiego [Kul86] oraz Deo [Deo80]; niestety natrafiono przy tym na problemy terminologiczne. Praktycznie w każdej pracy używana jest nieco inna terminologia. Właściwie oddaje to cytat:

„Oto jeszcze jeden przykład znacznych różnic terminologii stosowanej przez wielu autorów. W literaturze marszruta pojawia się jako droga, ścieżka, trasa lub progresja krawędziowa; łańcuch występuje jako droga, półprosta lub trasa; droga jako łańcuch, łuk, droga prosta lub łańcuch prosty; łańcuch zamknięty jako droga cykliczna, łańcuch cykliczny, kontur lub obwód; a cykl jako obwód, cykl elementarny, droga okrężna, łańcuch cykliczny lub obwód prosty!”¹

W związku z tym starano się, w miarę możliwości, wykorzystywać jedną terminologię - zgodną z tą, która została zaprezentowana w pracy [Kul86].

Definicja 1.1.

Prostym, ważonym grafem nieorientowanym nazywamy parę:

$$(V, E^G),$$

gdzie: V – przeliczalny zbiór elementów v_i zwanych **węzłami grafu**; E^G – przeliczalny zbiór elementów e_{ij}^G zwanych **krawędziami grafu**, przy czym e_{ij}^G utożsamia się z nieuporządkowaną parą węzłów $\{v_i, v_j\}$ ze zbioru V , gdzie $v_i \neq v_j$. Dla danej pary węzłów $\{v_i, v_j\}$ ze zbioru V w zbiorze E^G może istnieć tylko jedna krawędź. Z krawędzią e_{ij}^G jest związana pewna liczba rzeczywista zwana **wagą krawędzi**, oznaczana przez w_{ij}^G .

Węzły v_i i v_j związane z krawędzią e_{ij}^G nazywa się **węzłami końcowymi krawędzi** e_{ij}^G .

Tak zdefiniowany prosty, ważony graf nieorientowany oznaczać będziemy przez $G(V, E^G)$.

W literaturze grafy nieorientowane bywają także nazywane grafami nieskierowanymi.

Węzeł izolowanym nazywamy węzeł niezwiązany z żadną krawędzią.

W grafie prostym nie występują **pętle**, to jest krawędzie związane tylko z jednym węzłem (dla których $v_i = v_j$).

Stopniem węzła nazywamy liczbę krawędzi, dla których jest on węzłem końcowym.

¹ [Wil85, s.39]

Definicja 1.2.

Graf $G'(V', E^{G'})$ jest **podgrafem** grafu $G(V, E^G)$, jeżeli $V' \subset V$ oraz $E^{G'} \subset E^G$ i każda krawędź należąca do $E^{G'}$ ma te same wierzchołki końcowe w $G'(V', E^{G'})$ jak i w $G(V, E^G)$.

Definicja 1.3.

Droga w grafie niezorientowanym o początku w węźle v_{i_1} i końcu w węźle v_{i_k} nazywamy uporządkowaną parę $[S, L]$, w której S jest ciągiem wierzchołków $[v_{i_1}, v_{i_2}, \dots, v_{i_k}]$, tak dobranym, że dla każdej pary sąsiadujących wierzchołków ciągu $v_{i_s}, v_{i_{s+1}}$ istnieje krawędź $e_{i_s, i_{s+1}}^G$ łącząca węzeł v_{i_s} i $v_{i_{s+1}}$. Zbiór takich krawędzi tworzy zbiór L .

Drogę, w której początek i koniec znajdują się w tym samym węźle nazywamy **cyklem**.

Drogę, w której żaden wierzchołek nie pojawia się więcej niż jeden raz nazywamy **ścieżką** (drogą elementarną).

Definicja 1.4.

Graf niezorientowany nazywamy **grafem spójnym**, jeżeli dowolną parę jego wierzchołków można połączyć za pomocą drogi.

Definicja 1.5.

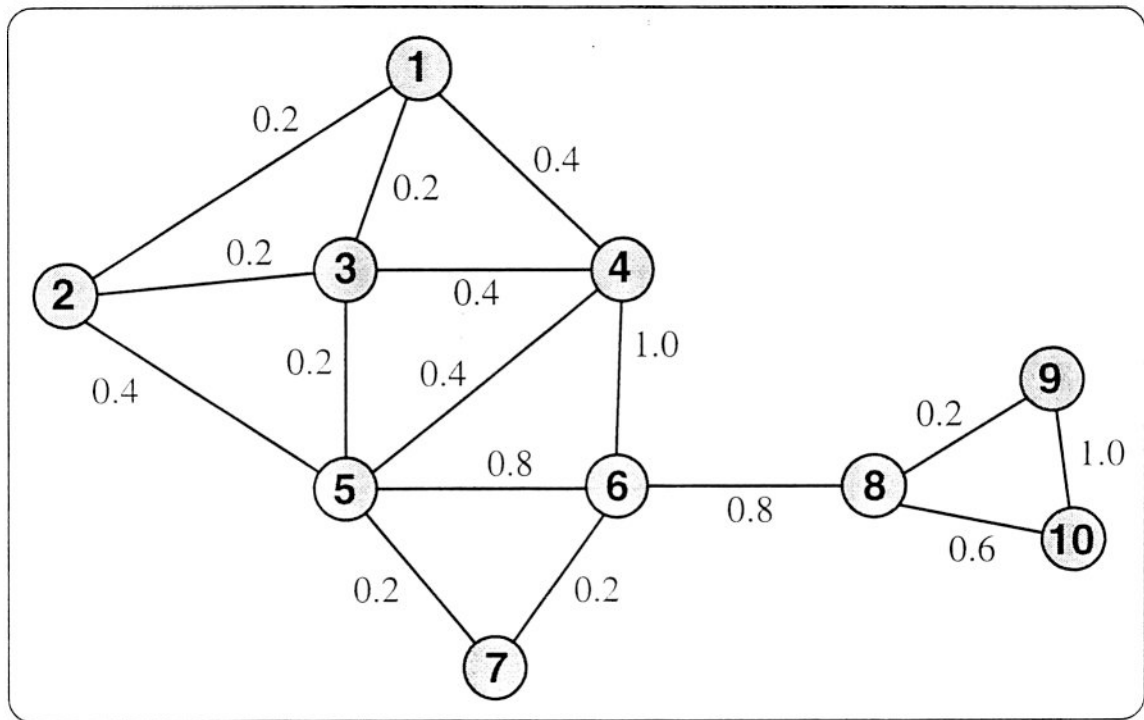
Maksymalny podgraf spójny grafu ważonego, taki że poszerzenie go o jakiejkolwiek dodatkowe wierzchołki tego grafu powoduje utratę spójności nazywamy **składową spójności grafu** lub krótko **składową grafu ważonego**.

Graf niezorientowany, mający k składowych nazywamy **grafem k -składowym**.

Składową grafu ważonego będzie się dalej oznaczać symbolem:

$$G^S(V^S, E^S).$$

Przykład składowej grafu ważonego zawiera rys. 1.1.



Rys. 1.1. Składowa grafu ważonego

Stopniem grafu nazywamy liczbę jego węzłów (moc zbioru V) i oznaczamy literą N .

Definicja 1.6.

Drzewem nazywamy spójny, nieorientowany graf pozbawiony cykli, o skończonym stopniu.

Twierdzenie 1.1.

Następujące właściwości grafu $G(V, E^G)$ są parami równoważne:

1. Graf $G(V, E^G)$ jest drzewem.
2. Dwa dowolne węzły grafu $G(V, E^G)$ można połączyć ścieżką, przy czym jest to połączenie jedyne.
3. $G(V, E^G)$ jest grafem spójnym, w którym jego stopień N oraz liczba krawędzi $\text{card}(E^G)$ spełniają zależność $N = \text{card}(E^G) + 1$,
4. $G(V, E^G)$ jest grafem pozbawionym cykli, w którym $N = \text{card}(E^G) + 1$.
5. $G(V, E^G)$ jest grafem spójnym, różnym od grafu zupełnego stopnia $N \geq 3$ (graf zupełny to taki, w którym dla każdej pary węzłów istnieje krawędź je łącząca), a spójność grafu $G(V, E^G)$ zostaje naruszona w przypadku usunięcia z niego dowolnej krawędzi (bez przyległych węzłów).

Dowód twierdzenia 1.1 (w rozszerzonej postaci) znajduje się w pracy [Kul86, s.124-125].

Twierdzenie to definiuje, na różne sposoby, pojęcie drzewa.

1.2. MAKSYMALNY PRZEPŁYW

Maksymalny przepływ jest nierozzerwalnie związany z sieciami komunikacyjnymi zwanymi także transportowymi.

Tę część rozdziału, która jest związana z maksymalnym przepływem, opracowano na podstawie monografii Kulikowskiego [Kul86], Deo [Deo80] Korzana [Kor78] i Lipskiego [Lip89], jak również prac Hu [Hu61, Hu82] oraz Gusfielda [Gus87, Gus90].

Definicja 1.7.

Sieciami komunikacyjnymi nazywamy prosty, spójny, niezorientowany graf ważony, w którym wagi związane z krawędziami są liczbami nieujemnymi.

Waga krawędzi reprezentuje pojemność krawędzi, zwaną także przepustowością krawędzi, którą można interpretować jako maksymalną ilość pewnego dobra (takiego jak: woda, gaz, energia), przenieszonego wzdłuż tej krawędzi.

W literaturze sieć komunikacyjna jest zwykle definiowana jako graf zorientowany. Jednakże w opisywanych w dalszej części pracy algorytmach i twierdzeniach wykorzystywana jest sieć jako graf niezorientowany.

Definicja 1.8.

Przekrojem w grafie spójnym, ważonym $G^S(V^S, E^S)$ nazywamy taki zbiór krawędzi tego grafu, których usunięcie z grafu spowoduje, że stanie się on niespójny, pod warunkiem, że usunięcie żadnego z podzbiorów właściwych tego zbioru nie uczyni grafu $G^S(V^S, E^S)$ niespójnym.

Sumę wag krawędzi przekroju nazywa się **pojemnością przekroju**.

Korzystając z faktu, że przekrój dzieli spójny graf na dwie składowe, można go opisywać za pomocą dwóch zbiorów węzłów z tych dwóch składowych. Oznacza to, że jeżeli dany przekrój dzieli graf na dwa zbiory węzłów X oraz Y , to przekrój ten można oznaczyć jako $(X|Y)$.

Definicja 1.9.

Przekrojem dzielącym parę węzłów v_1, v_2 w grafie spójnym $G^S(V^S, E^S)$ nazywamy taki przekrój, który umieszcza v_1 oraz v_2 w dwóch różnych składowych.

Minimalnym przekrojem, dla danej pary węzłów, nazywa się taki przekrój dzielący te węzły, którego wartość pojemności jest minimalna.

Definicja 1.10.

Przepływ, w danej sieci komunikacyjnej $G^S(V^S, E^S)$, jest przypisaniem pewnej liczby nieujemnej $f_{i,j}$ do każdej krawędzi $e_{ij}^G \in E^S$ tak, żeby były spełnione następujące warunki:

1. $0 \leq f_{i,j} \leq w_{ij}^G$, dla każdej krawędzi $e_{ij}^G \in E^S$.
2. Istnieje pewien określony węzeł v_s w grafie $G^S(V^S, E^S)$ zwany **źródłem** lub **węzłem źródłowym**, dla którego:

$$\sum_{i=1}^N f_{s,i} - \sum_{i=1}^N f_{i,s} = w, v_i \in V^S.$$

3. Istnieje inny, określony węzeł v_t w grafie $G^S(V^S, E^S)$ zwany **ujściem**, dla którego:

$$\sum_{i=1}^N f_{t,i} - \sum_{i=1}^N f_{i,t} = -w, v_i \in V^S.$$

4. Wszystkie inne węzły nazywa się węzłami pośrednimi. Dla każdego węzła pośredniego v_j mamy:

$$\sum_{i=1}^N f_{j,i} - \sum_{i=1}^N f_{i,j} = 0, v_i \in V^S.$$

Wielkość w nazywa się **wartością przepływu** z węzła źródłowego v_s do ujścia v_t .

Przepływ, który maksymalizuje wielkość w nazywa się **przepływem maksymalnym**. Ponieważ w następnych częściach pracy wykorzystany zostanie jedynie przepływ maksymalny, symbol $f_{s,t}$ będzie dalej oznaczał wartość maksymalnego przepływu ze źródła v_s do ujścia v_t .

W nieorientowanym grafie ważonym z rysunku 1.1, między węzłem nr 6 a 8, przetransportować można maksymalnie 0.8 jednostek „dobra”. Możliwe jest to tylko w bezpośredni sposób, czyli poprzez krawędź $\{6,8\}$ o wadze 0.8. Między węzłem 8 i 10, natomiast, wartość maksymalnego przepływu wynosi także 0.8, gdyż przez bezpośrednią krawędź łączącą węzeł 8 z 10 można przetransportować 0.6 jednostek a przez ścieżkę zawierającą węzły: 8,9,10 – 0.2 jednostki „dobra”. Oczywiście bezpośrednie połączenie źródła i ujścia w postaci krawędzi może nawet nie istnieć a wartość maksymalnego przepływu może być większa od zera, dzięki przepływom pośrednim. Na przykład maksymalny przepływ między węzłem 7 i 8 wynosi 0.4 jednostki, gdyż w ścieżce $\{7-0.2-6-0.8-8\}$ maksymalny przepływ wynosi 0.2 i poprzez ścieżkę $\{7-0.2-5-0.8-6-0.8-8\}$ także 0.2 jednostki.

Twierdzenie 1.2.

Wartość maksymalnego przepływu między źródłem i ujściem jest równa wartości minimalnego przekroju dzielącego źródło i ujście

Dowód twierdzenia 1.2, będącego jednym z podstawowych w teorii sieci komunikacyjnych, sformułowanego przez Forda i Fulkersona, zawarty jest w bardzo wielu opracowaniach między innymi w pracy Kulikowskiego [Kul86, s.438].

1.2.1. DRZEWO MAKSYMALNYCH PRZEPŁYWÓW

Definicja 1.11.

Dwa, niezorientowane, spójne grafy ważone $G_1^S(V^S, E_1^S)$ oraz $G_2^S(V^S, E_2^S)$ nazwiemy **grafami o jednakowych przepływach**, jeżeli wartości maksymalnego przepływu między każdą parą węzłów są w obu grafach takie same, czyli:

$$(\forall v_i, v_j \in V^S) f_{(1)ij} = f_{(2)ij},$$

gdzie: $f_{(1)ij}$ – wartość maksymalnego przepływu w grafie $G_1^S(V^S, E_1^S)$, a $f_{(2)ij}$ – wartość maksymalnego przepływu w grafie $G_2^S(V^S, E_2^S)$.

Termin „o jednakowych przepływach” jest odpowiednikiem ang. *flow-equivalent*.

Definicja 1.12.

Drzewem maksymalnych przepływów dla składowej grafu ważonego $G^S(V^S, E^S)$, oznaczanym jako $T(V^S, E^T)$, nazywamy drzewo składające się ze zbioru węzłów V^S oraz zbioru krawędzi E^T , takie że: $T(V^S, E^T)$, oraz $G^S(V^S, E^S)$ są grafami o jednakowych przepływach.

Krawędź łączącą węzły v_i oraz v_j w drzewie maksymalnych przepływów¹ $T(V^S, E^T)$ będziemy dalej oznaczać przez e_{ij}^T , a wagę z nią związaną przez w_{ij}^T .

Drzewo maksymalnych przepływów dla składowej grafu ważonego $G^S(V^S, E^S)$ można traktować jako „reprezentację” wartości maksymalnych przepływów między wszystkimi węzłami w grafie $G^S(V^S, E^S)$.

Twierdzenie 1.3.

Dla każdej pary węzłów v_s oraz v_t z drzewa maksymalnych przepływów $T(V^S, E^T)$ dla składowej grafu ważonego $G^S(V^S, E^S)$, jeżeli e_{ij}^T jest krawędzią o minimalnej wadze w_{ij}^T na drodze z v_s do v_t w drzewie $T(V^S, E^T)$, to wartość maksymalnego przepływu z v_s do v_t w grafie $G^S(V^S, E^S)$ wynosi dokładnie w_{ij}^T .

Dowód twierdzenia 1.4 znajduje się w pracach Hu [Hu61, Hu82, Hu92].

Wniosek 1.4:

Maksymalny przepływ między dwoma węzłami drzewa maksymalnych przepływów jest równy minimalnej wadze krawędzi, po usunięciu której węzły te znajdują się w różnych składowych.

¹ Ponieważ nie istnieje polski odpowiednik pojęcia *equivalent flow tree*, dlatego też w całej pracy będzie używany termin „drzewo maksymalnych przepływów”, gdyż najbardziej oddaje on charakter tego drzewa. Inne, możliwe tłumaczenie to: „drzewo równoprzepływowe”, „drzewo ekwiwalentnych przepływów”, „drzewo odpowiadających przepływów”, czy „drzewo odpowiadające przepływowo”.

Dowód:

Prawdziwość wniosku wynika wprost z twierdzenia 1.3 oraz 1.1, pkt. 5.

□

Definicja 1.13.

Drzewem minimalnych przekrojów dla składowej grafu ważonego $G^S(V^S, E^S)$, oznaczanym jako $T'(V^S, E^T)$, nazywamy drzewo, w którym dla każdej pary węzłów $v_s, v_t \in V^S$, jeżeli pewna krawędź e_{ij}^T jest krawędzią o minimalnej wadze na drodze z v_s do v_t w drzewie $T'(V^S, E^T)$, to dwie składowe grafu $T'(V^S, E^T \setminus \{e_{ij}^T\})$ wyznaczają minimalny przekrój między v_s oraz v_t w grafie $G^S(V^S, E^S)$. Zbiór węzłów jednej składowej odpowiada zbiorowi węzłów jednej strony minimalnego przekroju, a węzły drugiej składowej odpowiadają drugiej stronie przekroju.

Graf $T'(V^S, E^T \setminus \{e_{ij}^T\})$ składa się z dwóch składowych, wynika to z twierdzenia 1.1, pkt 5.

Drzewo minimalnych przekrojów (ang. *minimum cut tree*¹) dla składowej grafu ważonego będzie w skrócie nazywane **drzewem minimalnych przekrojów** lub **drzewem przekrojów**.

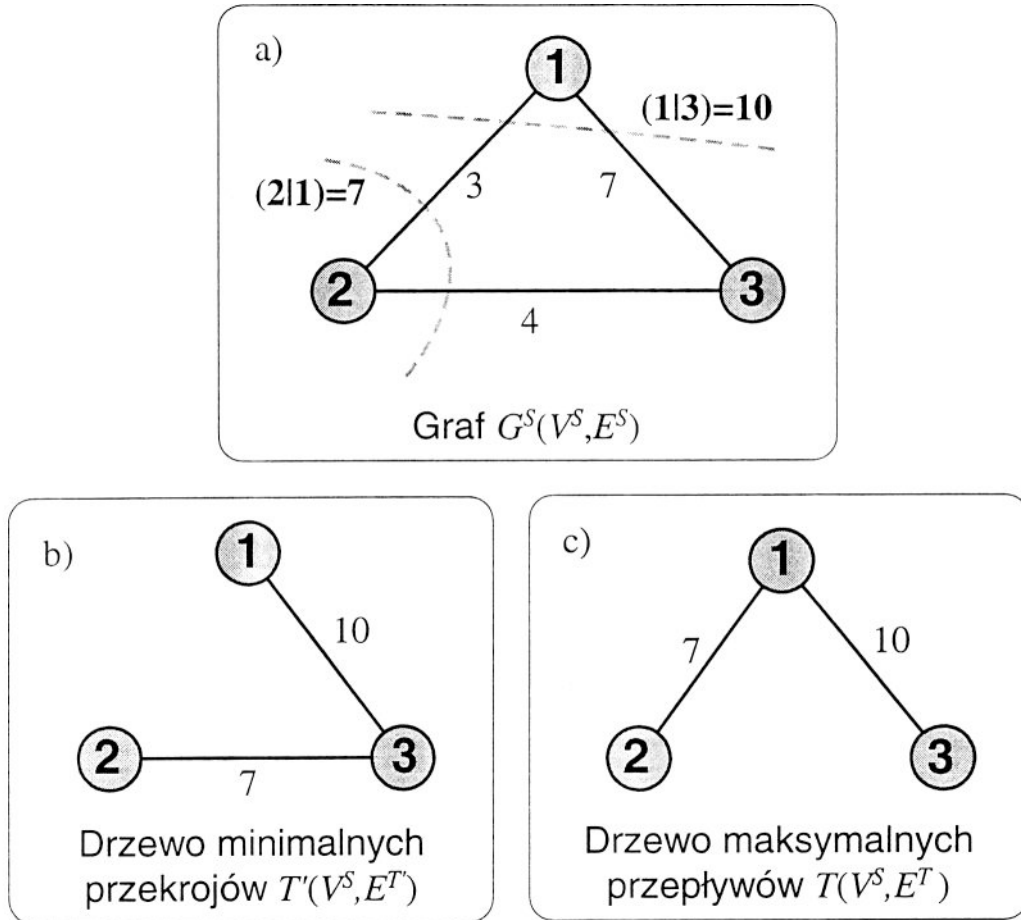
Drzewo przekrojów reprezentuje w skróconej formie jeden minimalny przekrój dla każdej pary węzłów grafu $G^S(V^S, E^S)$.

Na rysunku 1.2 przedstawiony jest 3-węzłowy graf $G^S(V^S, E^S)$ – (a), drzewo przekrojów $T'(V^S, E^T)$ odpowiadające grafowi $G^S(V^S, E^S)$ – (b) oraz drzewo maksymalnych przepływów $T(V^S, E^T)$ dla grafu $G^S(V^S, E^S)$ – (c). Zauważmy, że $T(V^S, E^T)$ nie jest drzewem przekrojów dla $G^S(V^S, E^S)$ a drzewo $T'(V^S, E^T)$ jest także drzewem maksymalnych przepływów. Gomory i Hu [Hu61, Hu82] sformułowali i udowodnili twierdzenie:

Twierdzenie 1.5.

Każde drzewo minimalnych przekrojów $T'(V^S, E^T)$ dla składowej grafu ważonego $G^S(V^S, E^S)$ jest zarazem drzewem maksymalnych przepływów dla tej składowej.

¹ W literaturze można spotkać także inne angielskie określenia tego drzewa: „*cut tree*” (drzewo przekrojów), „*Gomory-Hu cut tree*” (drzewo przekrojów Gomory-Hu), „*Gomory-Hu tree*” (drzewo Gomory-Hu) oraz „*Gomory-Hu min-cut tree*” (drzewo minimalnych przekrojów Gomory-Hu).



Rys. 1.2. Graf $G^S(V^S, E^S)$ z zaznaczonymi minimalnymi przekrojami (a), odpowiadające mu drzewo minimalnych przekrojów (b) oraz drzewo maksymalnych przepływów uzyskane za pomocą algorytmu EQ Gusfielda (c).

Twierdzenie 1.6.

Dla składowej grafu ważonego $G^S(V^S, E^S)$ wartości maksymalnych przepływów dla wszystkich par węzłów ze zbioru V^S mogą być wyznaczone poprzez $N-1$ obliczeń maksymalnego przepływu, gdzie $N = \text{card}(V^S)$.

Dowód twierdzenia 1.6 został przedstawiony przez Gomory i Hu w pracy [Hu61]. Dowód analogicznego twierdzenia dotyczącego minimalnych przekrojów ujęty jest w pracy Chenga i Hu [Hu92].

Zgodnie z twierdzeniem 1.6 wartości maksymalnego przepływu dla wszystkich $\binom{N}{2} = \frac{N(N-1)}{2}$

par węzłów mogą być obliczone za pomocą rozwiązania $N-1$ problemów maksymalnego przepływu w grafie $G^S(V^S, E^S)^1$.

1.2.2. TWORZENIE DRZEWA MAKSYMALNYCH PRZEPIWÓW. ALGORYTM GUSFIELDA

Gomory i Hu w pracach [Hu61, Hu82] opisali metodę tworzenia drzewa przekrojów za pomocą $N-1$ obliczeń maksymalnych przepływów (w ten sposób udowodnili twierdzenie 1.6). W metodzie

¹ Zyskuje się w ten sposób czynnik $N/2$, gdy porównamy to do obliczeń dokonanych dla wszystkich możliwych par.

Gomory-Hu wymagane jest jednak, aby wszystkie, obliczane przekroje były parami nieprzecinającymi się (ang. *noncrossing*)¹. Zapewnienie nieprzecinania się przekrojów wymaga dodatkowych obliczeń, które nie są bezpośrednio związane z ustalaniem wartości maksymalnych przepływów. Metodę Gomory-Hu nie jest łatwo zaimplementować w postaci programu komputerowego [Gus90]. Uzyskuje się w niej drzewo maksymalnych przepływów i zarazem drzewo minimalnych przekrojów, nawet wtedy, gdy drzewo przekrojów nie jest potrzebne.

Istnieje inna metoda, opracowana przez Granota i Hassina [Gra86], która mogłaby być wykorzystana do tworzenia drzewa maksymalnych przepływów, ale niekoniecznie będących zarazem drzewami minimalnych przekrojów. W metodzie tej także niezbędne jest rozwiązanie jedynie $N-1$ problemów maksymalnych przepływów, ale nie wymaga się tego, aby kolejne przekroje się nie przecinały.

Gusfield zaproponował znaczącą modyfikację metody Gomory-Hu, umożliwia to uzyskiwanie drzewa maksymalnych przepływów, niekoniecznie będącego zarazem drzewem minimalnych przekrojów. W wyniku tego powstał algorytm EQ Gusfielda, opisany w pracach [Gus87, Gus90, Kaz99b].

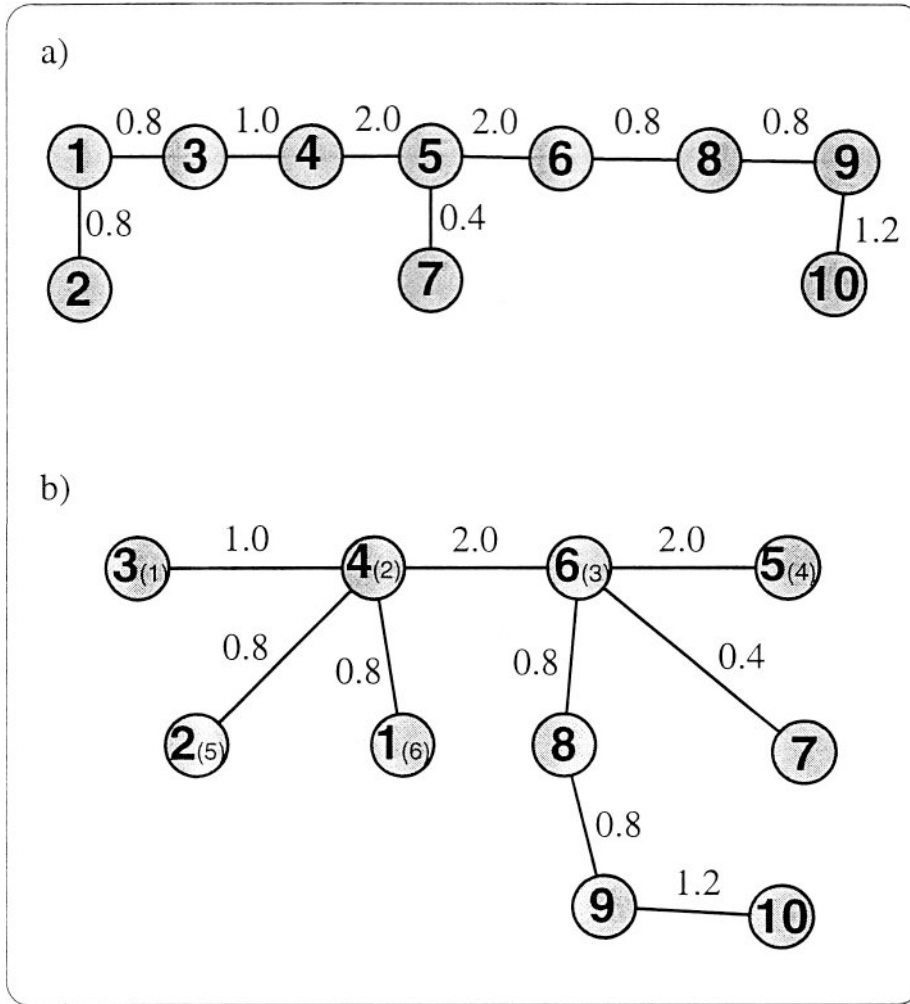
W algorytmie EQ Gusfielda najpierw numerowane są węzły grafu: od 1 do N i tworzone jest z nich drzewo - gwiazda o środku w węźle 1. Z gwiazdy tej, po przesunięciach węzłów i wyznaczeniu wag krawędzi, powstaje końcowe drzewo maksymalnych przepływów.

Algorytm Gusfielda nie jest jednoznaczny. W efekcie jego działania mogą być utworzone różne drzewa maksymalnych przepływów, w zależności od kolejności numeracji węzłów oraz od wybranego minimalnego przekroju przez zewnętrzną funkcję. Dla danej pary węzłów v_i oraz v_j może istnieć wiele minimalnych przekrojów o tej samej wartości sumy wag krawędzi przekroju czyli tej samej wartości maksymalnego przepływu. Oznacza to że, wartość maksymalnego przepływu między dwoma węzłami w grafie jest zawsze jedna, a minimalnych przekrojów może być wiele. Postać drzewa maksymalnych przepływów zależy m.in. od tego, który z minimalnych przekrojów zostanie wybrany przez zewnętrzną funkcję. W algorytmie EQ Gusfielda nie są stawiane żadne warunki dla minimalnego przekroju (mogą się one przecinać).

Na rys. 1.3 przedstawiono dwa drzewa maksymalnych przepływów utworzone za pomocą algorytmu EQ Gusfielda dla grafu z rys. 1.1. Są one różne, gdyż inaczej ponumerowano w nich węzły grafu wejściowego – składowej grafu ważonego. Na rys. b), w nawiasach, małymi literami zaznaczono zmienioną, nową numerację.

Szczegółowy opis algorytmu EQ i wyjaśnienie procesu tworzenia drzew z rys. 1.3 zawarty jest w załączniku A oraz w pracy Kazienki [Kaz99b].

¹ Dwa przekroje, dzielące zbiór węzłów V^S na dwa zbiory odpowiednio $(X|Y)$ oraz $(U|V)$, nazywa się przecinające się (ang. *cross cuts* lub *crossing cuts*), gdy wszystkie cztery podzbiory $X \cap U$, $X \cap V$, $Y \cap U$ oraz $Y \cap V$ są niepuste. Przekroje są nie przecinające się wtedy, gdy któryś z tych podzbiorów jest pusty.



Rys. 1.3. Dwa drzewa maksymalnych przepływów utworzone z zastosowaniem algorytmu EQ Gusfielda dla grafu z rys. 1.1

1.3. HIERARCHIA

W celu zdefiniowania hierarchii niezbędne jest określenie relacji porządkującej¹ na zbiorze X . Jest to relacja $\leq_p$, spełniająca warunki [Kur80]:

$$a) (\forall x \in X) x \leq_p x \text{ (zwrotność)}, \quad (1.1)$$

$$b) (\forall x, y \in X) x \leq_p y \wedge y \leq_p x \Rightarrow x = y \text{ (słaba antysymetria)}, \quad (1.2)$$

$$c) (\forall x, y, z \in X) x \leq_p y \wedge y \leq_p z \Rightarrow x \leq_p z \text{ (przechodność)}. \quad (1.3)$$

Parę $(X, \leq_p)$ nazywa się zbiorem uporządkowanym lub krótko - uporządkowaniem².

¹ Relacja porządkująca bywa także nazywana relacją uporządkowania lub relacją porządku. Terminologię zaczerpnięto z pracy Kuratowskiego [Kur80].

² „W dawniejszej terminologii relacje porządkujące nazywane były relacjami częściowego uporządkowania, natomiast termin „uporządkowanie” oznaczał uporządkowanie liniowe.” [Kur80, s.74].

Definicja 1.14.

Hierarchią na zbiorze X nazywamy uporządkowanie $(H(X), \leq_H)$, spełniające warunki:

1. $H(X) \subseteq X \times \mathbb{N}_+$.
2. $(\exists (x_0, c_0)) (\forall (x, c) \in H(X)) (x, c) \leq_H (x_0, c_0)$.
3. $(\forall (x, c), (x', c') \in H(X)) ((x, c) <_H (x', c') \Rightarrow c' < c)$.
4. $(\forall (x, c) \in H(X)) \{c \neq c_0 \Rightarrow (\exists! (x', c') \in H(X)) [(x, c) <_H (x', c') \wedge \neg((\exists (x'', c'') \in H(X)) (x, c) <_H (x'', c'') <_H (x', c'))]\}$,

gdzie:

$$a <_H b \Leftrightarrow a \leq_H b \wedge a \neq b.$$

Z warunku 1 definicji wynika, że zbiór $H(X)$ składa się z par (x, c) , nazywanych **elementami hierarchii**, w których c określa **zagłębienie elementu**, czyli odległość elementu od korzenia.

Warunek 2, który zapewnia istnienie tylko jednego, wyróżnionego elementu hierarchii, nazywamy **korzeniem hierarchii** lub **elementem najwyższym hierarchii**.

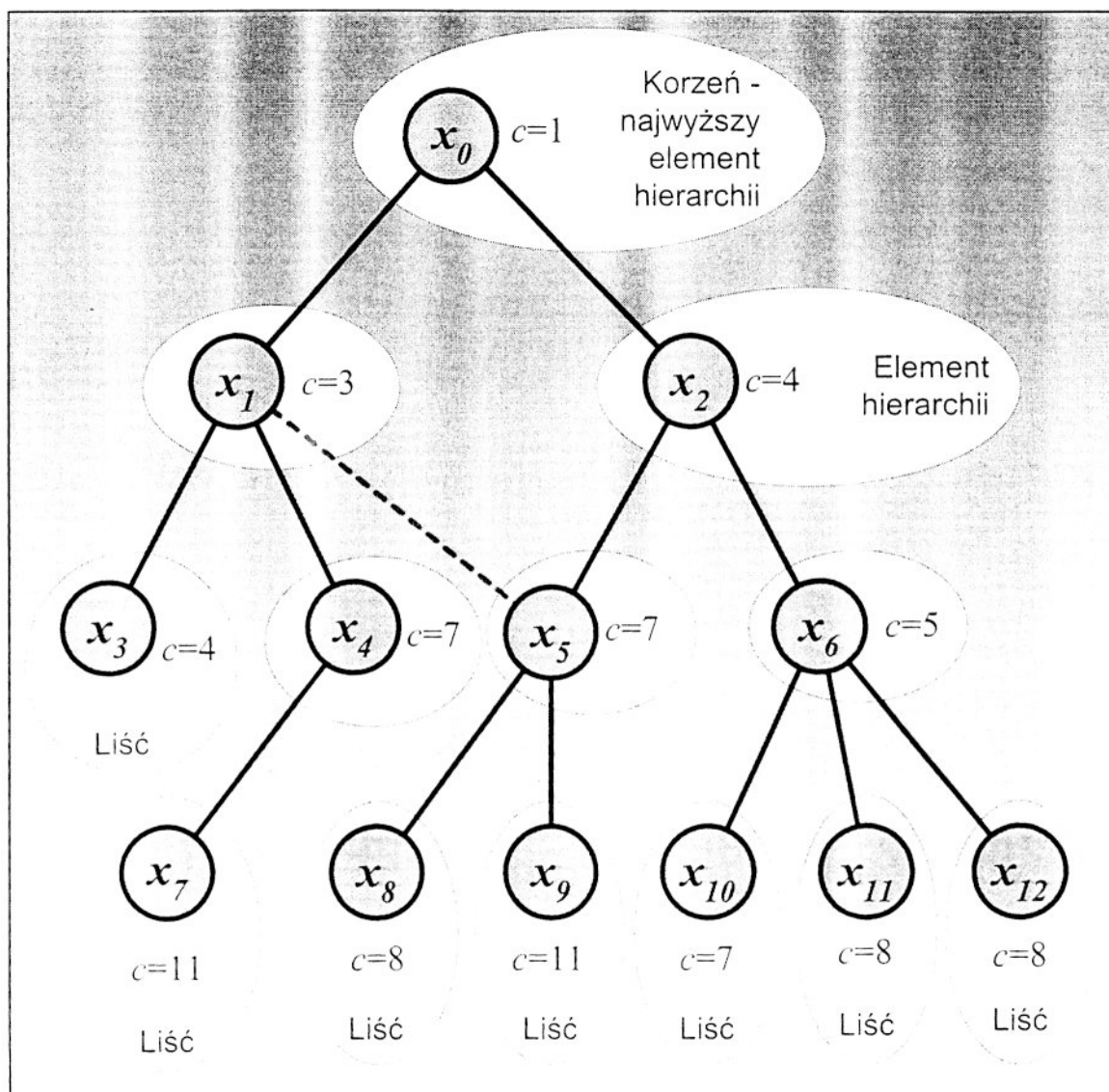
To, co jest niżej w hierarchii (warunek 3) ma większą wartość c – większe zagłębienie. Z tego korzeń hierarchii musi mieć najmniejszą wartość zagłębienia – c_0 .

Warunek 4 zapewnia strukturę drzewiastą, w której każdy element hierarchii ma tylko jednego rodzica – bezpośrednio nad nim znajdującego się przodka. W opisie warunku symbol $\exists!$ jest równoznaczny z: „istnieje dokładnie jedno”.

Relację uporządkowania hierarchii $\leq_H$ można rozumieć jako położenie „nieniżej” lub „niewyżej” w hierarchii, przy czym element najwyżej położony – najwyższy, to korzeń hierarchii. Formuła: $(x', c') \leq_H (x'', c'')$ oznacza, że (x', c') jest niżej lub w tym samym miejscu w hierarchii co (x'', c'') .

Hierarchię spełniającą warunki zawarte w definicji 1.14 przedstawiono na rysunku 1.4. Pomiedzy elementem hierarchii $(x_1, 3)$ oraz $(x_5, 7)$ linią przerywaną zaznaczono połączenie, które jest eliminowane przez warunek 4. Elementy hierarchii oznaczono białymi elipsami.

Relacja hierarchii $\leq_H$ nie jest uporządkowaniem liniowym, gdyż nie każdy element ze zbioru $H(X)$ hierarchii jest w relacji $\leq_H$ z każdym innym elementem tej hierarchii. Na przykład: element $(x_5, 7)$ nie jest w relacji $\leq_H$ z $(x_{10}, 7)$, podobnie jak $(x_1, 3)$ nie jest w relacji $\leq_H$ z $(x_2, 4)$, gdyż nie można stwierdzić, który z nich jest „wyżej” w hierarchii. Relacja $\leq_H$ jest widoczna na rysunku w postaci połączeń między elementami hierarchii. Na podstawie definicji uporządkowania relacja porządkująca $\leq_H$ elementy hierarchii jest przechodnia (wzór 1.3), więc nie tylko elementy połączone bezpośrednio znajdują się w relacji ze sobą, ale także te, które można połączyć idąc w dół lub w górę przez kolejne połączenia, bez zmiany kierunku. Element $(x_3, 4)$ jest w relacji $\leq_H$ (a nawet $<_H$) z $(x_0, 1)$, bo $(x_3, 4) \leq_H (x_1, 3) \leq_H (x_0, 1)$. Widać także, że korzeń jest w relacji z każdym elementem hierarchii.



Rys. 1.4. Hierarchia.

Jeżeli element hierarchii (x',c') jest w hierarchii wyżej niż element (x'',c'') , czyli: $(x',c') <_H (x'',c'')$, wtedy element (x',c') nazwiemy **przodkiem** elementu (x'',c'') , zaś element (x'',c'') **potomkiem** elementu (x',c') . Oczywiście korzeń hierarchii jest przodkiem wszystkich innych elementów hierarchii. Element $(x_{11},8)$ posiada następujących przodków: $(x_6,5)$, $(x_2,4)$ oraz $(x_0,1)$, zaś element $(x_1,3)$ posiada następujących potomków: $(x_3,4)$, $(x_4,7)$ oraz $(x_7,11)$. Bezpośredni przodek danego elementu (przodek najniżej położony) to **rodzic** danego elementu, zaś element bezpośrednio podległy to **dziecko** danego elementu. Element, który nie posiada dzieci nazywamy **liściem**.

Znaczenie wyżej wymienionych terminów i określeń jest zgodne z tym, jakie jest zwykle stosowane w informatyce, gdy wspomniane są struktury drzewiaste.

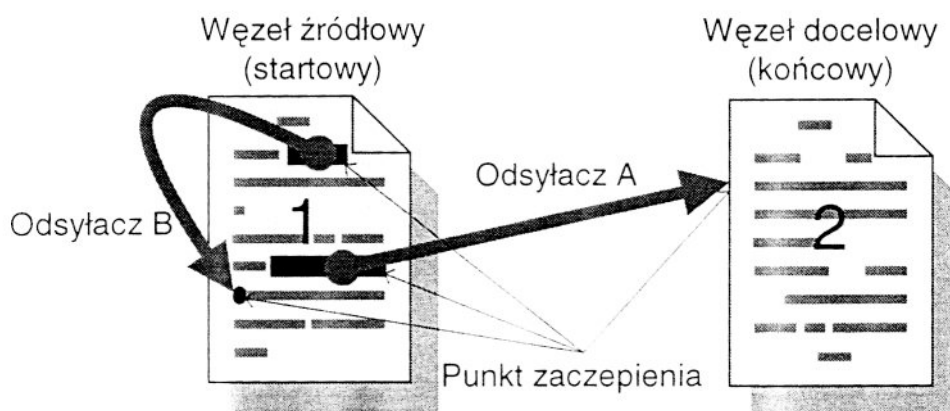
Definicja 1.14 jest zakorzeniona w teorii mnogości, chociaż możliwe byłoby wykorzystanie tutaj pojęcia dendrytu stosowanego między innymi w pracy Jardine'a i Sibsona [Jar71]. Jednak wydaje się, że przyjęta definicja stwarza większe możliwości przeprowadzenia oraz formalnego zapisania dowodów twierdzeń niezbędnych w dalszej części pracy.

2. STRUKTURA HIPERTEKSTU

2.1. ELEMENTY STRUKTURY HIPERTEKSTU

Hipertekst jest zbiorem powiązanych ze sobą dokumentów hipertekstowych. Składa się on z dwóch elementów:

- **Węzłów**, zwanych także dokumentami lub ramkami¹, zawierających zwykle tekstową porcję informacji.
- **Odsyłaczy**, zwanych także połączeniami lub linkami (ang. *link*), tworzących powiązania między poszczególnymi dokumentami i między częściami tego samego węzła.



Rys. 2.1. Elementy struktury hipertekstu.

W schemacie przedstawionym na rys. 2.1 odsyłacz A łączy **dokument źródłowy 1** (ang. *source node*) z **dokumentem docelowym 2** (ang. *destination node*), a odsyłacz B - różne części tego samego dokumentu. Można także stwierdzić, że **odsyłacz A prowadzi z dokumentu 1 do dokumentu 2**. Miejsce dokumentu, z którego i do którego prowadzi odsyłacz nazywane jest **punktem zaczepienia**² odsyłacza (ang. *anchor*). W przypadku odsyłacza A, końcowym punktem zaczepienia jest cały dokument docelowy 2, zaś dla odsyłacza B – wybrane miejsce w dokumencie 1.

Węzły prócz swojej treści tekstowej (w szczególnym przypadku, na przykład indeksu - jego część tekstowa może być pusta) zawierają także elementy odsyłaczy, to znaczy przynajmniej ich punkty zaczepienia oraz część prezentującą odsyłacz użytkownikowi - zwykle jest to nazwa węzła docelowego, do którego odsyłacz się odnosi. Jednostką informacji prezentowaną jednorazowo użytkownikowi jest tekst zawarty w węźle wraz z nazwami wychodzących z niego odsyłaczy. O strukturze hipertekstu przesądzają odsyłacze³, które odróżniają go od systemu linearnego.

W systemach hipertekstowych odsyłacze najczęściej są w całości umieszczone w węźle startowym. Tak jest na przykład w przypadku dokumentów tworzonych z wykorzystaniem języka opisu

¹ „Węzły są w systemach hipertekstu różnie określane: notecards, cards, frames, pages, boxes, information units, notes, content nodes, statements, messages, articles, hyper-molecules, guidelines, documents. Nazwy te wskazują na interpretacje pojęcia węzeł przez autorów systemów hipertekstu” [Abr96, s.4]

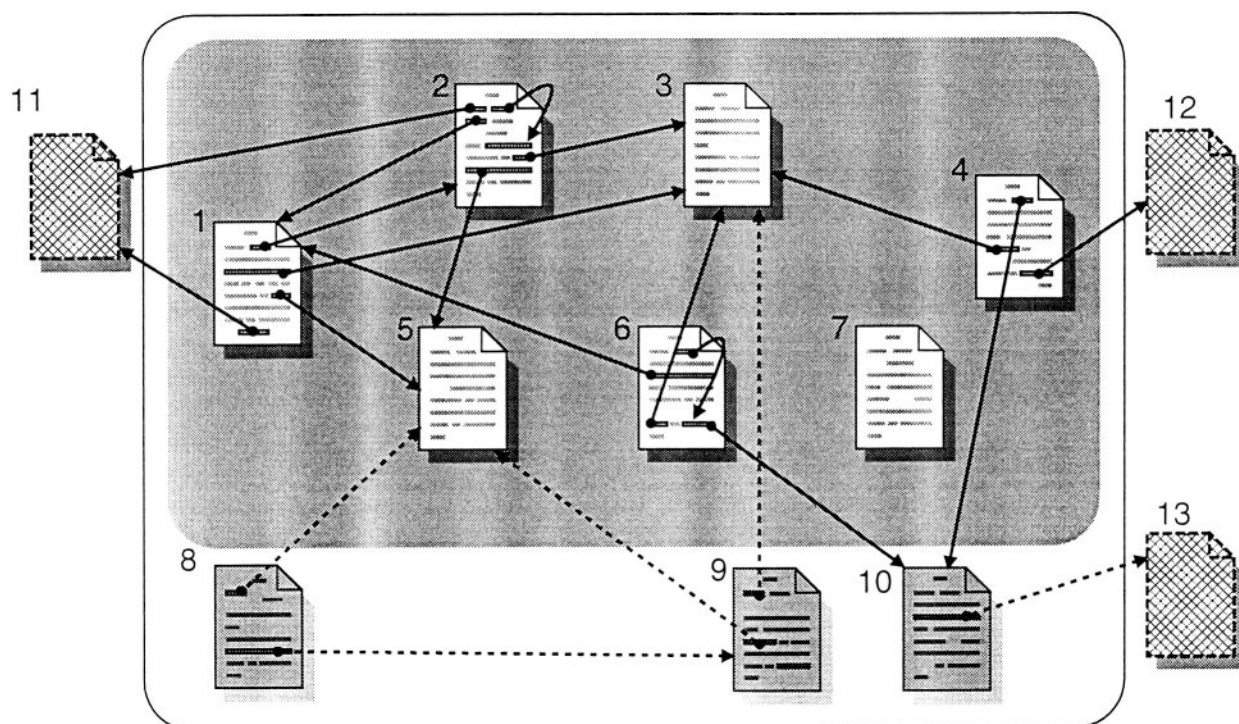
² nazywane czasami miejscem zakotwiczenia; wykorzystywane jest także pojęcie startowego i końcowego punktu zaczepienia

³ „Odsyłacze są dominującym elementem struktury WWW, odzwierciedlonym przez twórców systemu w jego nazwie (World Wide Web).” [Dan00b, s.11]

stron WWW - HTML. W systemie WWW odsyłacze przyjmują postać adresu URL¹, umieszczonego w konkretnym miejscu węzła źródłowego. Adres ten jednoznacznie identyfikuje węzeł docelowy a nawet konkretne miejsce w tym węźle. W związku z tym, można zauważyć, że odsyłacz „znajduje się na danej, stronie WWW”, gdyż wszystkie informacje są umieszczone w węźle źródłowym.

Innym rozwiązaniem jest umieszczanie odsyłaczy w osobnej bazie danych lub generowanie ich wtedy, gdy są potrzebne [Ash94, Bie97].

Na rysunku 2.2 przedstawiono zbiór dokumentów hipertekstowych. Niektóre z nich nie są źródłowymi dla żadnych odsyłaczy (3, 5, 7), a niektóre nie są docelowymi (4, 7). Dokument 7 nie jest ani startowym ani końcowym. Może on istnieć w systemie - jest izolowany, chociaż taka sytuacja nie jest typowa dla hipertekstu. Startowe punkty zaczepienia, w węzłach źródłowych, oznaczono na rysunku czarnymi prostokątami.



Rys. 2.2. Zbiór dokumentów hipertekstowych.

W węzłach hipertekstowego środowiska WWW (na stronach WWW), prócz tekstu, występują także inne media: grafika – obrazy, dźwięk, sekwencje wideo. Z tego powodu w stosunku do niego bardziej adekwatnym określeniem byłoby „hipermedialny”. Inne media ze względu na swoją specyficzną zawartość informacyjną spełniają bardzo ważną rolę. Ich istnienie innych mediów nie ma jednak znaczenia dla jego struktury. Elementy multimedialne, które nie są związane z odsyłaczami, nic nie wnoszą do struktury systemu, a sytuacja, w której startowym punktem zaczepienia odsyłacza w dokumencie jest obraz (na przykład ikona) lub jego fragment (na przykład część „klikowalnej” mapy) niczym się nie różni od tej, gdy punktem zaczepienia jest miejsce w tekście. Na niektórych stronach WWW istnieją, prócz odsyłaczy prowadzących z elementów graficznych, także odpowiadające im odsyłacze umieszczone w tekście po to, aby umożliwić korzystanie ze stron również użytkownikom posiadającym jedynie tekstowe wersje przeglądarek. Dotyczy to na przykład wielu przeglądarek dla systemu UNIX. W dużej części prac poświęconych

¹ Adres URL (*Uniform Resource Locator*) wskazuje na zasób w sieci Internet - dokument WWW. Może on jednak także wskazywać na zasoby pochodzące z innych niż WWW systemów, na przykład listy z grup wiadomości USENET, zbiory dostępne za pomocą protokołu FTP [Kaz96], itd. Specyfikacja URL zawarta jest w RFC 1738 [Ber94].

problematyce hipertekstu termin „hipertekstowy” i „hipermedialny” jest używany zamiennie [Bie97].

Można wymienić również inne elementy struktury hipertekstowej, na przykład **ścieżka** (ang. *path*), czyli droga, jaką pokonuje użytkownik poprzez węzły i odsyłacze, **perspektywa** (ang. *perspective*) - całość informacji, jaka jest zawarta w zadanej ścieżce, **zakładka** (ang. *bookmark*), która umożliwia użytkownikowi zaznaczenie i powrót do określonych węzłów, **zaczepnik** (ang. *thumb tab*), czyli nadane przez autora specjalne oznaczenie odsyłaczy do kluczowych węzłów [Blo94] a także **trasa** (ang. *tour*), czyli sugerowana przez autorów ścieżka.

2.2. STRUKTURA SYSTEMÓW HIPERTEKSTOWYCH

Ze względu na sposób tworzenia można wyróżnić zasadniczo dwa rodzaje systemów hipertekstowych:

- **Systemy autorskie** – tworzone są przez jednego autora lub grupę ściśle współpracujących ze sobą autorów. Służą one zwykle do zaprezentowania wiedzy z jakiejś wybranej, wąskiej dziedziny, w której są oni specjalistami. Autorskimi można nazwać na przykład książki ukazujące się od kilkunastu lat jako systemy hipertekstowe lub hipermedialne (choćby [Shn89]), czasami nawet bez formy drukowanej. Systemem autorskim będzie także pojedynczy serwis WWW tworzony przez firmę komercyjną, a służący np. do zaprezentowania oferty produkowanych wyrobów. Serwis taki jest opracowywany i utrzymywany przez jej pracowników lub przez wynajętą, inną, wyspecjalizowaną firmę.
- **Systemy otwarte** – tworzone są przez wielu autorów, którzy ze sobą nie współpracują lub współpracują w ograniczonym zakresie. Można je zwykle także traktować jako zbiór słabo zależnych od siebie systemów autorskich. Ważną ich cechą jest stosunkowo duża elastyczność wykorzystywanego modelu i narzędzi. Zwykle w systemie otwartym istnieją jedynie nieliczne reguły (w przypadku WWW będzie to język opisu strony – HTML) a każdy z autorów może dodawać do tego swoje elementy. Oznacza to, że system jest tym bardziej otwarty, im posiada mniejszą liczbę reguł i dopuszcza jak największą elastyczności. Największym, hipertekstowym systemem otwartym jest WWW w sieci Internet¹. Każdy, kto posiada dostęp do programów służących do tworzenia stron i odpowiednie umiejętności, może opracować i zaprezentować w nim swój, własny system autorski.

Systemy autorskie charakteryzują się spójnością i dotyczą jednego zagadnienia (jednej dziedziny wiedzy, jednej firmy), zaś odsyłacze między dokumentami najczęściej odzwierciedlają wyłącznie powiązania treściowe między dokumentami. W systemach otwartych zdarzają się odsyłacze, które służą jedynie przemieszczaniu się.

Na rysunku 2.2 został przedstawiony prosty, otwarty system hipertekstowy, w którym istnieją powiązania (narysowane linią przerywaną) z innymi systemami hipertekstowymi lub innymi częściami większego systemu. Przykładem takiego systemu może być w sieci WWW pojedynczy **serwis informacyjny**. Posiada on zwykle wyróżnioną stronę główną (domową, początkową – ang. *home page*). Serwis taki utrzymywany jest przez jedną organizację (firmę) lub jedną osobę. Cały system WWW można traktować jako połączenie takich pojedynczych serwisów informacyjnych (rys. 2.3).

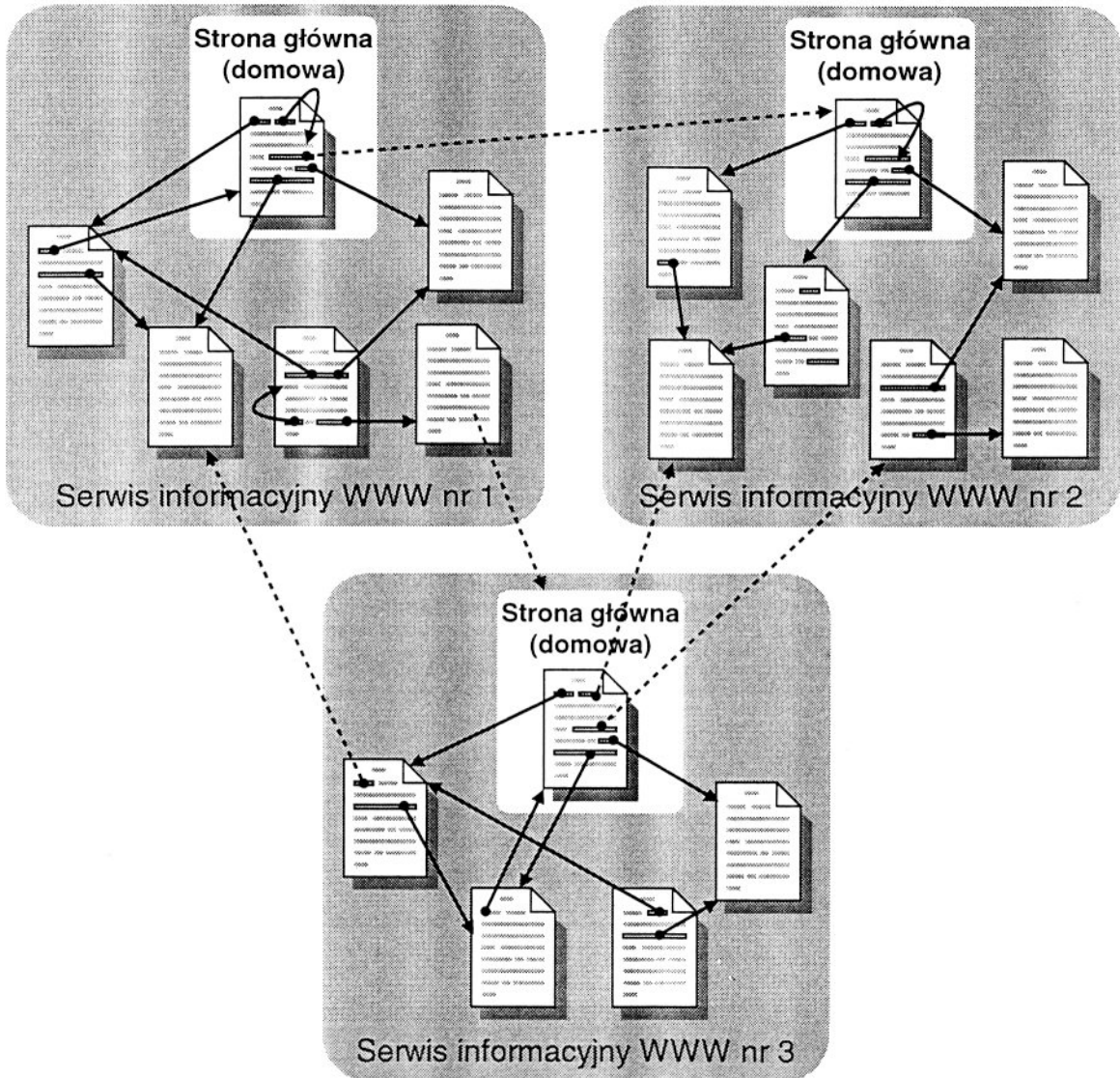
Ze względu na umiejscowienie systemy hipertekstowe można podzielić na:

- **Skupione** – cały system znajduje się w jednym miejscu, na przykład na płycie CD lub pojedynczym serwerze plikowym w sieci lokalnej. Można tutaj wymienić multimedialne

¹ „Ten specyficzny i niepowtarzalny system, może być traktowany jako wzorzec systemu otwartego, ponieważ większość jego własności intuicyjnie kojarzy się z definicjami otwartości systemu informacyjnego. (...). Podobnie każdy użytkownik (...) może tworzyć własne strony WWW” [Dan00b, s. 11]

firmowe systemy prezentacyjne oraz książki wydane w formie papierowej i równocześnie w postaci komputerowego systemu hipertekstowego (np. [Shn89]).

- **Rozproszone** – elementy systemu są rozproszone przede wszystkim poprzez sieć komputerową, tzn. dokumenty znajdują się na różnych, oddalonych od siebie komputerach. WWW jest przykładem takiego systemu.



Rys 2.3. Struktura WWW.

2.3. KLASYFIKACJA ELEMENTÓW STRUKTURY

Nośnikiem informacji w systemie hipertekstowym jest treść tekstowa dokumentów oraz odsyłacze łączące dokumenty. Z punktu widzenia powiązań semantycznych między dokumentami odsyłacze na równi z tekstem, mogą pełnić bardzo ważną rolę. Ma to szczególne znaczenie w otwartym środowisku, jakim jest WWW, w którym autorzy używają odmiennego słownictwa. Poszczególne strony mogą posiadać równoległe wersje językowe oraz różne postacie, na przykład graficzną i osobną – tekstową. Tekst nie musi już tak dobrze opisywać treści dokumentu, jak to ma miejsce w systemach autorskich. Jednocześnie, jako nośnik informacji w systemie WWW, wykorzystywane są także inne niż tekst media (głównie obrazy). Strony multimedialne często w ogóle nie posiadają treści tekstowej, ale zwykle zawierają odsyłacze. Odsyłacze więc, z punktu widzenia grupowania tematycznego dokumentów, będąc niezależnymi od tekstu i medium, nabierają w systemie WWW szczególnego znaczenia.

2.3.1. RODZAJE STRON WWW

Klasyfikacja, przeznaczona dla stron WWW, została zaproponowana przez Kleinberga, a potem przez zespół badawczy IBM [Kle99, Chak98a]. Wyodrębniono w niej dwa rodzaje stron: **autorytety** (ang. *authority pages*), zawierające dużo informacji na pewien temat oraz **koncentratory** (ang. *hub pages*), które posiadają dużo odsyłaczy do stron – autorytetów, czyli takie, które są dobrym indeksem, magazynem punktów startowych dla użytkownika w celu pozyskania informacji z wybranej dziedziny. Miernikiem tego, czy strona jest autorytetem jest to, że wiele stron *koncentratorów* posiada do niej odsyłacze. Dobre koncentratory posiadają wiele odsyłaczy do stron – autorytetów.

Kazienko zaproponował własną klasyfikację stron – dokumentów systemu WWW [Kaz99a]. Wyróżniono w niej następujące kategorie stron:

- a) **Strony domowe** (ang. *home page*), w tym strony organizacji, indywidualne oraz wizytówki – migawki (ang. *splash page*).
- b) **Strony organizacyjne** - nawigacyjne. Są to: spisy treści, hierarchie, mapy stron, indeksy a także spisy odsyłaczy do innych systemów z danej dziedziny (autorzy często nazywają je „ciekawe adresy”, „ulubione adresy”, „zobacz także”, „inne miejsca na temat...” itp.).
- c) **Dokumentacje** (instrukcje), czyli opisy produktów i usług, najczęściej zadawane przez użytkowników pytania wraz z odpowiedziami na nie¹, podręczniki wykonywania jakichś czynności, pomoc.
- d) **Teksty**, a w nich: teksty ogólne, wstępy, na przykład opisujące: misję i cele organizacji, artykuły, prace naukowe, opracowania, wykłady, raporty, prawa autorskie i informacje prawne, listy, życiorysy, biografie, historie; literatura piękna: teksty piosenek, poezje.
- e) **Strony narzędziowe** - strony interaktywne umożliwiające wpisanie przez użytkownika jakichś informacji i wysłanie ich do organizacji a w szczególności: strony wyszukiwania; formularze: zgłoszenia, rejestracje, zamówienia, itd., komentarze, uwagi, pytania, interfejsy dla innych usług i systemów dostępnych w sieci Internet, np. e-mail, WAIS, FTP, itd.
- f) **Strony dynamiczne**, których treść zmienia się w czasie (najczęściej okresowo). Można tutaj wyróżnić następujące rodzaje stron: dostęp do baz danych oraz serwisy informacyjne (wiadomości) a także systemy monitorujące i systemy interaktywne, na przykład gry lub quizy.
- g) **Multimedia** – strony, których zasadnicza zawartość składa się z innych niż tekst mediów; są to przede wszystkim strony radia *online* (ang. *real audio*), telewizji *online* (ang. *real video*), wirtualnej rzeczywistości VRML czy galerii zdjęć.

Wszystkie wymienione wyżej rodzaje stron, obejmujące łącznie całe środowisko systemu WWW, niosą (a przynajmniej mogą nieść ze sobą) informację tematyczną. Nawet strony organizacyjne, ze zwykle małą zawartością tekstową, będąc koncentratorami odsyłaczy do stron semantycznie związanych – poprzez te odsyłacze – stanowią cenne źródło informacji, którą można wykorzystać w procesie określania podobieństwa stron.

Na stronach dynamicznych, których zawartość tekstowa jest zmienna w czasie, często jedynym, niezmiennym źródłem informacji o treści są odsyłacze.

2.3.2. ROLA ODSYŁACZY HIPERTEKSTOWYCH

Garzotto, Paolini i Schwabe [Gar93] twierdzą, że odsyłacze pełnią w systemie hipertekstowym dwie podstawowe role:

- **reprezentacyjną** ujmując i prezentując relacje między porcjami informacji w tekście;

¹ tzw. strony FAQ (*frequently asked questions*) oraz Q&A (*questions & answers*)

- **nawigacyjną** obejmując ścieżki poruszania się użytkownika po systemie.

Czasami te dwie role przenikają się nawzajem, czasem są rozdzielone. Ten drugi przypadek występuje w sytuacjach, w których istniejące związki między poszczególnymi rodzajami informacji, umieszczonymi w różnych dokumentach, są nieodpowiednie dla konkretnej koncepcji nawigacji po systemie ustalonej przez autorów, to znaczy, odsyłacze służące wyłącznie poruszaniu się, mogą łączyć ze sobą dokumenty o słabszych związkach semantycznych.

Według Bloomfielda [Blo94] odsyłacze w swojej roli nawigacyjnej mogą służyć do:

- a) **Przeglądania hipertekstu przez użytkownika**, przy czym rozróżnia się tutaj przeglądania celowe (ang. *purposeful browsing*), w którym umożliwia ono świadome dotarcie do informacji poszukiwanej przez użytkownika i niecelowe (ang. *aimless browsing*), w którym użytkownik nie posiada ściśle sprecyzowanego celu, na przykład przegląda hipertekst dla zabawy.
- b) **Instruowania użytkownika**, czyli prezentacji użytkownikowi ściśle określonej ścieżki w systemie hipertekstowym. Ważnymi odsyłaczami tutaj są: „dokument następny”, „dokument poprzedni” oraz „początek”. Tego typu interakcja wykorzystywana jest szczególnie często w systemach prezentacyjnych, oraz w hipertekstowych systemach komputerowego wspomagania nauczania (CAL – *Computer Assisted/Aided Learning*) [Rob98]. Abramowicz twierdzi, że odsyłacze wyrażają w ten sposób związki pragmatyczne między łączonymi dokumentami [Abr96].

Na ogólnym poziomie odsyłacze łączą ze sobą porcje informacji. W związku z tym mogą one (według Conklina [Con87] oraz Bieberta i Kimbrough [Bie92]):

- a) Łączyć odnośniki (na przykład przypisy) z tekstem właściwym, przy czym zarówno odnośniki jak i tekst tworzą jedną całość, na przykład powiększenia rysunków.
- b) Łączyć komentarz z dokumentem właściwym.
- c) Łączyć dokumenty między sobą. W ten sposób powstaje hierarchia dokumentów lub inna struktura, na przykład graf.
- d) Łączyć tekst z następnikiem (potomkiem, dzieckiem).
- e) Być powiązane z jakąś akcją, na przykład: wysłanie wypełnionego formularza, rozpoczęcie wyszukiwania, wygenerowanie indeksu, itd.
- f) Być raczej wnioskiem wynikającym z treści lub struktury niż bezpośrednio istniejące w hipertekście.

Ta ostatnia rola pokazuje, że teoretyczne modele hipertekstu uwzględniają, prócz odsyłaczy jawnie wprowadzonych do struktury przez autorów, także te, które nie są dostępne w trakcie przeglądania hipertekstu przez użytkowników. Tego rodzaju połączenia hipertekstowe są dynamicznie generowane i udostępniane użytkownikom jako efekt na przykład wyszukiwania. Takie rozwiązanie zaproponował Golovschinsky w systemie VOIR (*Visualization of Information Retrieval*) [Gol97].

2.3.3. TYPY ODSYŁACZY

Dotychczasowe (opisane w literaturze) klasyfikacje odsyłaczy hipertekstowych były najczęściej przeprowadzane na podstawie analizy systemów autorskich, w których twórca lub grupa twórców brała pełną odpowiedzialność za całość systemu. Tego typu klasyfikację zaprezentował Trigg i Weiser [Tri86]. Ich podział dotyczył hipertekstowych tekstów naukowych, dla których opracowano rozbudowaną klasyfikację odsyłaczy (75 kategorii) uwzględniającą przede wszystkim relacje semantyczne pomiędzy łączonymi przez nie dokumentami.

Wielu autorów biorąc pod uwagę dwie, wcześniej wymienione zasadnicze role (reprezentacyjną i nawigacyjną), jakie odgrywają odsyłacze w systemach hipertekstowych, rozróżnia dwa główne typy odsyłaczy:

- **Znaczeniowe** (semantyczne - ang. *semantic*) [Fre95], czyli bazujące na treści (ang. *content-based*) [Bar96], mające za zadanie wiązać dokumenty mieszczące się w tej samej lub pokrewnej tematyce. Większa część badaczy z dziedziny hipertekstu zajmuje się jedynie odsyłaczami znaczeniowymi.
- **Organizacyjne** (ang. *organizational*) [Bar96], czyli odsyłające (ang. *referential*) [Fre95], których głównym celem jest lepsza (szybsza, łatwiejsza) nawigacja po systemie. W związku z tym, bywają one także nazywane **nawigacyjnymi**.

Inny podział odsyłaczy zaproponował De Rose, tworząc hierarchię 12 typów odsyłaczy [DeR89]. Uwzględnia on następujące 2 gałęzie klas odsyłaczy: rozwinięte (ang. *extensional*) – zapamiętane bezpośrednio w systemie¹ oraz mocne (ang. *intensional*), które wynikają z treści dokumentów i z tego powodu nie muszą być w sposób bezpośredni pamiętane w systemie hipertekstowym.

Analizę różnych typologii odsyłaczy przeprowadzono w pracy [Kaz98b], a w pracy [Kaz99a] przedstawiono klasyfikację, przygotowaną dla systemu WWW. Podzielono w niej odsyłacze znaczeniowe i nawigacyjne (organizacyjne) na podklasy. Wyodrębniono także odsyłacze dynamiczne, związane z dokumentami, których zawartość jest zmienna w czasie (dostęp do baz danych, interakcja z użytkownikiem).

Klasyfikacja odsyłaczy ma istotne znaczenie przy rozwiązywaniu problemów w dwóch dziedzinach: w dynamicznej prezentacji stron użytkownikowi oraz w automatycznym ich przetwarzaniu (a także ich otoczenia). Wiele systemów informacyjnych w sieci Internet korzysta z automatycznego przetwarzania stron, zwłaszcza przy wyszukiwaniu informacji. Odsyłaczom innym niż semantyczne (nawigacyjnym) powinno być w takim przetwarzaniu nadawane mniejsze znaczenie, gdyż strony, które one łączą mają ze sobą często mniejsze związki semantyczne. Jednakże nie należy odsyłaczy nawigacyjnych pomijać zupełnie:

„Niektórzy mogą twierdzić, że odsyłacze hipertekstowe często służą tylko potrzebom nawigowania. Jednakże, rola odsyłaczy w nawigowaniu może być traktowana jako przypuszczenie autora hipertekstu, że ktoś zainteresowany aktualną stroną będzie także zainteresowany stroną do której prowadzi odsyłacz. Naszym przekonaniem było to, że pokazując ściśle podobieństwo odsyłaczy do cytowań, w efekcie zastosowania analizy cytowań do WWW, pojawią się sensowne struktury”²

Haas i Grams [Haa98] przeprowadziły badania dotyczące odsyłaczy w systemie WWW. Na ich podstawie stwierdziły, że 82% odsyłaczy nawigacyjnych, występujących bez kontekstu (sąsiedztwa) tekstowego, jest izolowanych na stronach. Natomiast 84% odsyłaczy izolowanych pełni rolę nawigacyjną. Z dużym prawdopodobieństwem można więc stwierdzić, że odsyłacze nawigacyjne są umieszczane bez bezpośredniego sąsiedztwa zasadniczej treści strony. Jednocześnie autorki wykazały, iż odsyłacze nawigacyjne występują najczęściej u góry (na początku) lub u dołu (na końcu) strony. Są to – jak dotychczas – jedyne badania, których wyniki można wykorzystać do automatycznego rozpoznawania rodzajów odsyłaczy w systemie WWW.

2.4. ŚRODOWISKO HIPERTEKSTOWE I KOLEKCJA HIPERTEKSTOWA

W otwartych systemach hipertekstowych występują zarówno odsyłacze łączące dokumenty (należące do tego systemu) jak i takie, które prowadzą do dokumentów już lub jeszcze nieistniejących. Między dokumentami może występować wiele odsyłaczy i zgodnie z wyżej przedstawionym podziałem mogą być one nawigacyjne lub semantyczne. Wszystkie te cechy hipertekstu uwzględniono w definicji **środowiska hipertekstowego**.

¹ Conklin [Con87] nazywa je odnośnikowe (ang. *referential*).

² [Pit97, s. 385], tłum. autora.

Definicja 2.1.

Środowiskiem hipertekstowym nazywamy parę (D^E, L^E) , w której:

1. D^E – skończony zbiór dokumentów tekstowych,
2. L^E – skończony zbiór odsyłaczy,

$$L^E \subseteq D^E \times (D^E \cup D^-) \times \mathbf{N}_+ \times T^L,$$

gdzie:

D^- – skończony zbiór dokumentów nienależących do D^E , $D^E \cap D^- = \emptyset$.

Dla ustalonych $d_i \in D^E$ oraz $d_j \in D^E \cup D^-$ zbiór $\{r \in \mathbf{N}_+ : (\exists t^L \in T^L)(d_i, d_j, r, t^L) \in L^E\}$ – będący zbiorem numerów odsyłaczy w ramach pary (d_i, d_j) – jest zbiorem kolejnych liczb naturalnych od 1.

$T^L = \{nav, sem\}$ jest zbiorem typów odsyłaczy; *nav* – odsyłacz nawigacyjny (ang. *navigational link*), zaś *sem* – odsyłacz znaczeniowy (ang. *semantic link*).

Jeżeli $(d_i, d_j, r, t^L) \in L^E$ to taką czwórkę nazywamy ***r*-tym odsyłaczem o początku w dokumencie d_i i końcu w dokumencie d_j typu t^L** i oznaczamy:

$$l_{ij}^{r, t^L} = (d_i, d_j, r, t^L),$$

gdzie:

d_i – oznacza początek odsyłacza (dokument początkowy – startowy),

d_j – oznacza koniec odsyłacza (dokument końcowy),

r – kolejny numer odsyłacza,

t^L – typ odsyłacza.

Tak zdefiniowane środowisko hipertekstowe oznaczać będziemy przez $HTE(D^E, L^E)$.

Elementami zbioru D^E są dokumenty składające się z tekstu w języku naturalnym.

D^- jest zbiorem dokumentów tekstowych, które już albo jeszcze nie istnieją, ale w dokumentach ze zbioru D^E istnieją do nich odsyłacze. Dokumenty należące do D^- to takie, które:

- a) należały do D^E , ale zostały z tego zbioru usunięte lub są w danym momencie niedostępne,
- b) przewiduje się, że będą dodane do D^E w przyszłości.

Jeśli dokument końcowy d_j odsyłacza należy do D^- (odsyłacz prowadzi do dokumentu nieistniejącego), wtedy taki odsyłacz nazwiemy **odsyłaczem nieaktualnym**.

Z definicji 2.1 wynika, że numeracja r odsyłaczy zaczyna się od 1 i kolejno wzrasta o jeden, w ramach pary (d_i, d_j) , bez względu na rodzaj t^L odsyłacza.

Możliwa jest sytuacja, w której $d_i = d_j$, czyli odsyłacz prowadzi do tego samego dokumentu, z którego wychodzi. W praktyce jest to najczęściej odesłanie w inne miejsce tego samego dokumentu, służące jedynie do przemieszczenia się w ramach dłuższego tekstu. Taki odsyłacz jest więc nawigacyjny.

W dalszej części pracy będzie używane pojęcie **dokument hipertekstowy** oznaczające zarówno dokument tekstowy, jak i dokument tekstowy wraz z wychodzącymi z niego odsyłaczami.

Jeżeli zbiór dokumentów hipertekstowych jest zmienny w czasie, wtedy środowisko hipertekstowe $HTE(D^E, L^E)$ odnosi się do konkretnego punktu czasowego.

Środowiskiem hipertekstowym można nazwać cały system WWW w sieci Internet. Wtedy strony WWW (w sensie ich treści tekstowej) odpowiadają dokumentom tekstowym, a zbiór odsyłaczy wychodzących z tych stron stanowi zbiór L^E . Odpowiednikiem zbioru D^- w systemie WWW jest zbiór stron (adresów URL dokumentów końcowych odsyłaczy umieszczanych na stronach), które nie istnieją lub nie są aktualnie dostępne w systemie¹, a do których autorzy stron mimo tego kierują odsyłacze.

Środowisko hipertekstowe zostało przedstawione na rys. 2.2. Dokumenty od 1 do 10 tworzą zbiór D^E , a dokumenty od 11 do 13 – zbiór D^- (dokumenty te nie posiadają treści ani nie są węzłami źródłowymi odsyłaczy). Odsyłaczami nieaktualnymi są te, dla których węzeł docelowy to dokumenty 11-13.

Grupowanie dokumentów w niniejszej pracy będzie przeprowadzane dla zbioru dokumentów hipertekstowych, stanowiącego pewną wydzieloną część środowiska hipertekstowego. Zdefiniujemy więc pojęcie **kolekcji hipertekstowej w środowisku hipertekstowym**. Nie jest ona równoznaczna z systemem hipertekstowym, gdyż kolekcja hipertekstowa może być zbiorem fragmentów wielu systemów, powstałym na przykład w wyniku procesu wyszukiwania w całym środowisku hipertekstowym.

Definicja 2.2.

Kolekcja hipertekstowa w środowisku hipertekstowym $HTE(D^E, L^E)$ nazywamy parę (D, L) składającą się ze zbioru dokumentów tekstowych D oraz zbioru odsyłaczy L , taką, że:

1. $D \subseteq D^E$,
2. $L \subseteq L^E$ jest zbiorem wszystkich odsyłaczy ze zbioru L^E , dla których dokument początkowy należy do D , czyli:

$$L = \{ l_{ij}^{r, t^L} : l_{ij}^{r, t^L} = (d_i, d_j, r, t^L) \in L^E, d_i \in D \}.$$

Tak zdefiniowaną kolekcję hipertekstową w środowisku hipertekstowym oznaczać będziemy przez $HT(D, L)$.

Kolekcja hipertekstowa w środowisku hipertekstowym jest więc podzbiorem dokumentów tekstowych, z tego środowiska, wraz z odsyłaczami z tych dokumentów.

W dalszej części pracy będzie wykorzystywane w zasadzie tylko jedno, ustalone środowisko hipertekstowe, więc termin „kolekcja hipertekstowa w środowisku hipertekstowym” będzie używany w skróconej formie: **kolekcja hipertekstowa**.

Dla ustalonej kolekcji hipertekstowej $HT(D, L)$ numerujemy, należące do niej, dokumenty, czyli:

$$D = \{d_1, d_2, \dots, d_N\},$$

gdzie: N – liczba dokumentów kolekcji.

Zauważmy, że $d_j \in D^E \cup D^-$, czyli w kolekcji hipertekstowej mogą istnieć odsyłacze prowadzące do dokumentów ze środowiska hipertekstowego, nienależących do tej kolekcji (jeżeli $d_j \in D^E \setminus D$) oraz do dokumentów nieistniejących (jeżeli $d_j \in D^-$).

Zbiór D będzie także dalej nazywany **kolekcją dokumentów**.

¹ Powodem może być to, że strona została usunięta, przeniesiona, jeszcze nie została stworzona lub serwer WWW, na którym się znajduje, jest wyłączony.

Kolekcja hipertekstowa została przedstawiona na rys. 2.2. Tworzą ją dokumenty umieszczone na szarym tle (od 1 do 7) wraz z odsyłaczami narysowanymi linią ciągłą. Dokumenty od 8 do 10 i odsyłacze narysowane linią przerywaną należą do środowiska hipertekstowego, ale nie należą do kolekcji hipertekstowej.

Definicja 2.3.

Odsyłacz $l_{ik}^{r,i^L} \in L$, wychodzący z dokumentu $d_i \in D$, prowadzący do dokumentu $d_k \in D^E \cup D^-$, $d_i \neq d_k$, nazywamy **pośrednim odsyłaczem cytowania z dokumentu d_i względem dokumentu d_j** , $d_j \in D$, $d_j \neq d_k$, $d_j \neq d_i$, wtedy gdy istnieje także odsyłacz $l_{jk}^{r,i^L} \in L$.

Pośredni odsyłacz cytowania jest więc odesłaniem do wspólnego, dla dokumentów d_i oraz d_j , potomka-dziecka (dokument d_k). W sposób pośredni łączy on dokumenty d_i i d_j , które są elementami kolekcji D . Wspólny potomek nie musi należeć do kolekcji ($d_k \in D^E \cup D^-$).

Dla dokumentu 4 z rys. 2.2, pośrednim odsyłaczem cytowania względem dokumentu 6 jest odesłanie $4 \rightarrow 3$ oraz $4 \rightarrow 10$. Dla dokumentu 1 względem dokumentu 2 pośrednimi odsyłaczami cytowania są: $1 \rightarrow 5$ oraz $1 \rightarrow 11$ (jest to odsyłacz nieaktualny, gdyż dokument 11 należy do zbioru D^-).

Definicja 2.4.

Odsyłacz $l_{ki}^{r,i^L} \in L^E$, wychodzący z dokumentu $d_k \in D^E$, prowadzący do dokumentu, $d_i \in D$, $d_i \neq d_k$, nazywamy **pośrednim odsyłaczem współcytowania dokumentu d_i względem dokumentu d_j** , $d_j \in D$, $d_j \neq d_k$, $d_j \neq d_i$, wtedy gdy istnieje także odsyłacz $l_{kj}^{r,i^L} \in L^E$.

Pośredni odsyłacz współcytowania dokumentu d_i względem d_j jest odsyłaczem prowadzącym do dokumentu d_i od wspólnego, dla obu tych dokumentów, przodka-rodzica (dokument d_k). Przyczym, w przeciwieństwie do dokumentu - potomka, przodek nie może należeć do zbioru D^- .

Dla dokumentu 5 z rys. 2.2 pośrednim odsyłaczem współcytowania względem dokumentu 3 są odsyłacze $2 \rightarrow 5$ oraz $1 \rightarrow 5$, a także $9 \rightarrow 5$ (mimo, że dokument 9 oraz odsyłacze $9 \rightarrow 5$ i $9 \rightarrow 3$ nie należą do kolekcji hipertekstowej).

3. PRZEGLĄD METOD GRUPOWANIA DOKUMENTÓW HIPERTEKSTOWYCH

3.1. FUNKCJA PODOBIENSTWA DOKUMENTÓW

Podobieństwem dokumentów zajmowano się w nauce od wielu lat. Przegląd najważniejszych, klasycznych podejść do wyznaczania podobieństwa przedstawiono w pracy Saltona [Sal83], a w języku polskim w monografii Dąbrowskiego i Laus-Maczyńskiej [Dąb78]. Większość ówczesnych autorów opierała swoje metody na statystyce występowania słów w dokumentach. Metody te nadal są rozwijane, na przykład przez Singhal, Buckley'a i Mitrę [Sin96]. Inne podejście związane było z rozpoznawaniem treści dokumentów poprzez analizę podobieństwa całych akapitów, badanie sąsiedztwa terminów, analizę całych zdań, analizę morfologiczną.

Osobną ścieżką, którą podążali badacze, była analiza cytowań. Najszerzej znane są prace Garfielda i jego współpracowników z Instytutu Informacji Naukowej z Filadelfii. Obszerny zbiór dokumentów źródłowych z zakresu analizy cytowań zamieszczono w bibliografii załączonej do monografii Garfielda [Gar79]. Dotyczy ona jednak wyłącznie tekstów naukowych. Systemy hipertekstowe przejęły niejako koncepcję cytowania z literatury naukowej. Odsyłacze umieszczane w treści dokumentów hipertekstowych pełnią taką samą rolę, jak cytowania – odsyłają czytelnika do innych dokumentów związanych tematycznie z fragmentem, w którym cytat (odsyłacz) został umieszczony¹. Połączeniem indeksu cytowań i WWW jest system CiteSeer opracowany w firmie NEC, umożliwiający wyszukiwanie tekstów naukowych na podstawie cytowań² [Bol98, Law99a]. Systemy hipertekstowe posiadają jednak specyficzne cechy, które odróżniają je od cytowań:

„(...) w WWW spotykamy nie tylko dokumenty połączone przez więcej niż jeden odsyłacz, ale również dokumenty odsyłające do siebie nawzajem. Takie przypadki oczywiście nie występują w indeksach cytowań. Do opublikowanego dokumentu nie można wprowadzić żadnych zmian. Wersja z wprowadzonymi zmianami może ukazać się wyłącznie jako nowy (inny) dokument.”³

Gdy już zostanie ustalony nośnik podobieństwa, do rozstrzygnięcia pozostaje jak informację zawartą w tym nośniku wyrazić w postaci funkcji podobieństwa. Propozycje kilkudziesięciu takich funkcji, dla różnych dziedzin, podał Spät w pracy [Spä77]. W dalszym przeglądzie ograniczono się jedynie do tych funkcji, które pojawiają się w literaturze w kontekście przetwarzania dokumentów.

Formalnie, funkcję $S: D \times D \rightarrow \mathbf{R}$ nazywamy funkcją podobieństwa⁴ pomiędzy parą elementów ze zbioru D wtedy i tylko, gdy spełniony jest warunek (wg [Soe67, Dąb78]):

$$(\forall d_i, d_j \in D) S(d_i, d_i) \geq S(d_i, d_j).$$

Warunkiem mocniejszym jest:

$$(\forall d_i, d_j \in D) S(d_i, d_i) = S(d_j, d_j) \geq S(d_i, d_j). \quad (3.1)$$

Jeżeli, prócz warunku (3.1), spełniony jest warunek symetrii:

¹ Larson w pracy [Lar96] potraktował strony WWW jak dokumenty naukowe zawierające cytowania i na podstawie tego, analizując zbiory dokumentów wyszukiwarki WebCrawler i Altavista, wyznaczył m.in. strony najbardziej popularne a także dokonał analizy cytowań stron (analizy odsyłaczy) dla dziedziny „nauka o Ziemi” (ang. *Earth Science*)

² System CiteSeer jest dostępny w sieci WWW, pod adresem: <http://citeseer.nj.nec.com/cs>

³ [Dan00a, s.4]

⁴ funkcja podobieństwa (ang. *similarity function*) jest także nazywana funkcją bliskości (ang. *nearness function*)

$$(\forall d_i, d_j \in D) S(d_i, d_j) = S(d_j, d_i), \quad (3.2)$$

wtedy taką funkcję S nazywa się **quasi-metryczną funkcją podobieństwa** (ang. *quasimetric similarity function*).

Jeżeli quasi-metryczna funkcja podobieństwa spełnia warunek trójkąta:

$$(\forall d_i, d_j, d_k \in D) S(d_i, d_k) \geq \frac{S(d_i, d_j) S(d_j, d_k)}{S(d_i, d_j) + S(d_j, d_k)}, \quad (3.3)$$

to funkcję tę nazywamy **półmetryczną funkcją podobieństwa** (ang. *half-metric similarity function*).

Półmetryczna funkcja podobieństwa nazywa się **metryczną funkcją podobieństwa** (ang. *metric similarity function*), jeżeli spełniony jest warunek:

$$(\forall d_i, d_j \in D, d_i \neq d_j) S(d_i, d_i) = S(d_j, d_j) < S(d_i, d_j). \quad (3.4)$$

Inną, wykorzystywaną miarą jest **odległość**¹ (ang. *distance*), będąca odwrotnością (choć niekoniecznie w sensie matematycznej funkcji odwrotności) podobieństwa. Im odległość jest większa, tym podobieństwo jest mniejsze i odwrotnie.

3.1.1. PODOBIEŃSTWO STRUKTURALNE

Podobieństwo strukturalne odzwierciedla informację, którą zawiera powiązanie odsyłaczami dokumentów hipertekstowych.

Najprostszą miarą podobieństwa strukturalnego między dwoma dokumentami można zbudować na bazie odsyłaczy bezpośrednio łączących dokumenty. Jeżeli istnieje odsyłacz między dwoma dokumentami (bez względu na to, w którą stronę i ile ich jest), to funkcja podobieństwa między tymi dokumentami przyjmuje wartość jeden, jeżeli nie istnieje – zero. Takie podejście zostało wykorzystane w opisywanych dalej grafowych metodach grupowania kolekcji hipertekstowych (pkt. 3.6).

Mukherjea, Foley i Hudson [Muk94] w systemie *Navigational View Builder*, przeznaczonym dla autorskich systemów hipertekstowych, do tak wyznaczonego podobieństwa dodali także rodzaj odsyłacza, przy czym założono, że rodzaj ten określa autor w trakcie tworzenia hipertekstu.

Binarne odzwierciedlenie powiązań strukturalnych hipertekstu może być zastąpione liczbą odsyłaczy między dokumentami. Według Chena [Chen97] podobieństwo strukturalne $S^s(d_i, d_j)$ między dokumentem d_i oraz d_j wyraża się wzorem:

$$S^s(d_i, d_j) = \frac{nl_{ij}}{\sum_{k=1}^N nl_{ik}} = \frac{nl_{ij}}{nl_i} \quad (3.5)$$

gdzie: nl_{ij} – liczba odsyłaczy prowadzących z dokumentu d_i do dokumentu d_j , a nl_i – liczba wszystkich odsyłaczy wychodzących z dokumentu d_i , a prowadzących do innych dokumentów w ramach kolekcji dokumentów D . Tak określone podobieństwo nie jest symetryczne, gdyż nl_{ij} nie musi się równać nl_{ji} .

W pracy [Dan00a] zaproponowano ważenie poszczególnych odsyłaczy. Podobieństwo strukturalne $S^s(d_i, d_j)$ jest wtedy sumą wag wszystkich tych odsyłaczy, które łączą dokumenty d_i i d_j (zarówno w jedną jak i drugą stronę). Przy określaniu wartości wag, można przyjąć najprostsze rozwiązanie, tzn. wartość 1 dla wszystkich odsyłaczy. Tak wyznaczone podobieństwo jest funkcją symetryczną, dla której jednak należy zastosować pewną formę normalizacji (aby zakres wartości był przedziałem $[0,1]$), poprzez normalizację całej funkcji lub odpowiednie określenie wag.

¹ w literaturze używany jest także termin „odmienność” (ang. *dissimilarity*)

Inną miarę podobieństwa można zbudować na bazie informacji, która jest zawarta w długości najkrótszej ścieżki łączącej dwa dokumenty. Im długość ścieżki jest większa, tym podobieństwo powinno być mniejsze. Problemem pozostaje tutaj postać funkcji odzwierciedlającej tę relację. W przypadku długości najkrótszych ścieżek można wyróżnić dwa podejścia:

- a) z zachowaniem kierunku odsyłaczy; funkcja podobieństwa nie jest symetryczna;
- b) z pominięciem kierunku (istnienie odsyłacza w jedną stronę jest traktowane jakby także istniał odsyłacz w drugą stronę).

Botafogo, Rivlin i Schneiderman z Uniwersytetu Maryland [Bot92, Riv94] zaproponowali, aby w przypadku, gdy ścieżka między dokumentami nie istnieje, jako jej długość przyjąć liczbę dokumentów N w kolekcji¹. Wyróżnili oni także kilka parametrów związanych ze strukturą wzajemnych powiązań, jakie można przyporządkować dokumentom hipertekstowym. Dla pojedynczego dokumentu można obliczyć sumę długości wszystkich najkrótszych ścieżek do pozostałych dokumentów w kolekcji. W przypadku, gdy początkiem ścieżek jest pewien dokument d_i , suma taka będzie wyznaczać parametr $COD(d_i)$ (ang. *converted out distance*). Natomiast wtedy, gdy dokument d_i będzie końcem ścieżek – będzie to parametr $CID(d_i)$ (ang. *converted in distance*). Korzystając z tych parametrów można obliczyć względną centralność zewnętrzną – $ROC(d_i)$ (ang. *relative out centrality*) lub wewnętrzną: $RIC(d_i)$ (ang. *relative in centrality*). Są one definiowane następująco:

$$ROC(d_i) = \frac{\sum_{j=1}^N COD(d_j)}{COD(d_i)}, \quad RIC(d_i) = \frac{\sum_{j=1}^N CID(d_j)}{CID(d_i)}$$

Duża wartość parametru $ROC(d_i)$, może sugerować, że d_i jest on dobrym „dokumentem centralnym” lub kandydatem na korzeń hierarchii dokumentów.

Omówione parametry dokumentów hipertekstowych, związane ze ścieżkami łączącymi dokumenty, można zastosować do stworzenia odpowiednich funkcji podobieństwa. Niestety, jak dotąd, propozycje te nie zostały wykorzystane w opracowywanych później metodach grupowania, klasyfikacji czy tworzenia rankingów.

Brin i Page z Uniwersytetu Stanforda przy opracowywaniu wyszukiwarki internetowej Google² zastosowali szeregowanie odpowiedzi za pomocą funkcji *PageRank* strony [Bri98]. Dla strony d_i , do której istnieją odsyłacze ze stron $d_1, \dots, d_k$, funkcja *PageRank* czyli $PR(d_i)$ jest definiowana następująco:

$$PR(d_i) = (1-a) + a \left(\frac{PR(d_1)}{nl_1} + \dots + \frac{PR(d_k)}{nl_k} \right),$$

gdzie: a – stały parametr rankingu, należący do przedziału $(0,1)$, przez autorów ustalony na 0.85; nl_i – liczba wszystkich odsyłaczy wychodzących ze strony d_i .

$ROC(d_i)$, $RIC(d_i)$ i $PageRank(d_i)$ są wyznaczane dla jednego dokumentu d_i . Chcąc wykorzystać je do określania podobieństwa między dwoma dokumentami d_i i d_j , należałoby połączyć wartości obliczone osobno dla dokumentu d_i , z tymi dla d_j .

Kolejnym elementem, który może mieć wpływ na wartość podobieństwa strukturalnego jest umiejscowienie odsyłacza, jego kontekst. W systemie ParaSite [Spe97], przy wyszukiwaniu stron WWW podobnych do danej, wykorzystywana jest informacja o tym, czy dwa odsyłacze znajdujące się na stronie d_1 , a prowadzące do dwóch różnych stron d_2 i d_3 , zostały umieszczone w bliskim

¹ Jest to wartość jakiej nie osiągnie żadna długość najkrótszej ścieżki, która maksymalnie może wynieść $N-1$.

² jest ona dostępna pod adresem: <http://www.google.com>

sąsiedztwie siebie – w tej samej „liście odsyłaczy”. Jeżeli tak, strony d_2 i d_3 traktuje się jako bardziej do siebie podobne.

3.1.2. PODOBIENSTWO TREŚCIOWE

Najczęściej wykorzystywanym nośnikiem podobieństwa treściowego jest częstość występowania terminów w poszczególnych dokumentach ze zbioru D . Po znormalizowaniu (na przykład maksymalną częstością występującą w dokumentach z D), może ona określać wagę terminu w dokumencie. Jeżeli zbiór tych wag, dla danego dokumentu, potraktujemy jako wektor nt -wymiarowy (gdzie nt – liczba różnych terminów ze wszystkich dokumentów ze zbioru D), wtedy otrzymujemy tzw. wektorowy model dokumentu. Podobieństwo treściowe $S^c(d_i, d_j)$ – między dokumentem d_i i d_j – można wyznaczyć z wykorzystaniem wektorów oraz znanych miar. Są nimi współczynniki: Dice’a, Jaccard’a i cosinusowy [Dąb78]:

a) współczynnik Dice’a:

$$S^c(d_i, d_j) = \frac{2 \sum_{k=1}^{nt} (w_{d_i,k}^t w_{d_j,k}^t)}{\sum_{k=1}^{nt} w_{d_i,k}^{t^2} + \sum_{k=1}^{nt} w_{d_j,k}^{t^2}}$$

b) współczynnik Jaccard’a:

$$S^c(d_i, d_j) = \frac{\sum_{k=1}^{nt} (w_{d_i,k}^t w_{d_j,k}^t)}{\sum_{k=1}^{nt} w_{d_i,k}^{t^2} + \sum_{k=1}^{nt} w_{d_j,k}^{t^2} - \sum_{k=1}^{nt} (w_{d_i,k}^t w_{d_j,k}^t)}$$

c) współczynnik cosinusowy:

$$S^c(d_i, d_j) = \frac{\sum_{k=1}^{nt} (w_{d_i,k}^t w_{d_j,k}^t)}{\sqrt{\sum_{k=1}^{nt} w_{d_i,k}^{t^2} \sum_{k=1}^{nt} w_{d_j,k}^{t^2}}} \quad (3.6)$$

gdzie: $w_{d_i,k}^t$ – waga k -tego terminu w dokumencie d_i .

Przedstawione miary podobieństwa stosuje się zwykle do obliczania wartości funkcji podobieństwa korzystając z pełnej treści dokumentów. Niezbędny jest więc dostęp do całego dokumentu, co w przypadku rozproszonego systemu WWW, może być problemem: gdy zachodzi potrzeba obliczenia w krótkim czasie podobieństwa dla dużej liczby rozproszonych w sieci dokumentów – stron, które trzeba sprowadzić. W metodzie STC¹ [Zam98, Zam99] analizie podlega nie cała treść dokumentów, a jedynie ich opis zwracany przez wyszukiwarki. Opis ten to zazwyczaj zawartość (początek) pola nagłówka strony *description* lub początek treści. W obu przypadkach jest on bardzo krótki (średnio ok. 100 bajtów), co jest małą wartością w porównaniu do pełnej treści strony². Jednakże jest dostępny w krótkim czasie.

¹ metoda STC jest opisana w pkt. 3.7.1.

² Analizując próbkę serwisów informacyjnych z losowo wybranych 2500 serwerów WWW, z których każdy zawierał średnio 289 stron, Lawrence i Giles [Law99b] obliczyli, że średni rozmiar strony WWW wynosi 18.7 kB (mediana: 3.9 kB) lub 7.3 kB (mediana: 0.98 kB) – w przypadku pozostawienia wyłącznie tekstu (bez znaków formatujących, komentarzy, pustych miejsc). Broder i inni, współpracujący z firmą Digital (operator wyszukiwarki Altavista), podczas eksperymentu, obejmującego 30 mln stron, obliczyli, że średnia wielkość strony¹ wynosi ok. 5 kB (bez znaków formatujących) [Bro97].

3.1.3. INNE NOŚNIKI PODOBIENSTWA. PODOBIENSTWO ZŁOŻONE.

Oprócz tekstu danego dokumentu można także wykorzystać inne nośniki podobieństwa¹, na przykład teksty dokumentów z jego najbliższego otoczenia. Otoczenie to jest wyznaczane poprzez odsyłacze hipertekstowe prowadzące z danego dokumentu. Takie podejście zaprezentował Marchiori dla stron WWW [Mar97]. Treść tekstowa danej strony jest zwiększana o treść tekstową wszystkich stron, do których można przejść po odsyłaczach prowadzących z danej strony². Pod uwagę brane są tylko strony, do których ścieżka (liczba niezbędnych odsyłaczy) jest nie dłuższa niż wartość parametru k . Dodawana treść tekstowa jest zmniejszana przez współczynnik F^i , gdzie F jest parametrem, należącym do przedziału $[0,1]$, a i - długością ścieżki prowadzącej od danej strony³. Dodawana informacja tekstowa jest jeszcze modyfikowana poprzez współczynnik związany z umiejscowieniem stron. Jeżeli strona, której treść dodajemy do danej, jest umiejscowiona na tym samym serwerze WWW, to jest ona inaczej traktowana niż strona z innego serwera (inna jest wtedy wartość parametru F). Jest to koncepcja bliska tej, jaką przedstawili Croft i Turtle w pracy [Cro93], w której przy wyszukiwaniu informacji w hipertekście i porównywaniu zgodności dokumentu z pytaniem wykorzystano także treść dokumentów z najbliższego sąsiedztwa danego dokumentu.

Pirolli, Pitkow oraz Rao w pracy [Pir96], zaproponowali wykorzystanie różnych źródeł informacji do automatycznej klasyfikacji stron WWW (podział stron na osiem ustalonych klas), pochodzących z serwisu firmy Xerox:

- a) Treści tekstowej. Użyto tutaj klasycznego modelu wektorowego (tekst jest reprezentowany w postaci wektora częstości występowania poszczególnych terminów). Dla strony obliczane jest jej podobieństwo treściowe, ale tylko do jej dzieci.
- b) Strukturze hipertekstowej. Dokonano tego przez uwzględnienie dla każdej strony:
 - liczby odsyłaczy, dla których dana strona jest startowa,
 - liczby stron, dla których istnieją odsyłacze prowadzące z nich do danej strony,
 - średniego zagnieżdżenia dzieci danej strony w strukturze serwisu (wyznaczonego poprzez liczbę znaków "/" w adresie URL strony - dziecka, co odzwierciedla jej poziom w strukturze katalogowej serwera WWW).
- c) Mierze popularności danej strony wyrażonej przez liczbę odwiedzin.

Szczególnie interesujące jest wykorzystanie liczby odwiedzin strony. Ustalenie jej jest jednak możliwe jedynie wtedy, gdy istnieje dostęp do statystycznych informacji połączeń z serwerem WWW, który daną stronę obsługuje.

Nie jest możliwe bezpośrednie wykorzystanie propozycji Pirolliego, Pitkova i Rao przy wyznaczaniu podobieństwa między parami dokumentów WWW. Nie można, na przykład określić wielkości wzajemnych powiązań hipertekstowych dla dwóch stron, gdyż obliczane jest jedynie „strukturalne miejsce” strony w ramach całego, jednego serwisu informacyjnego a nie względem drugiej strony. Wymienione rodzaje informacji można jednakże traktować jako propozycje obszarów, w których można poszukiwać elementów składowych do opracowania konkretnych funkcji podobieństwa.

¹ Przegląd różnych źródeł informacji, które mogą być wykorzystane przy wyznaczaniu podobieństwa między stronami WWW zawarto w pracy [Kaz00b].

² Marchiori jako treść tekstową przyjął ciąg bajtów, z których składa się strona. W innym podejściu, treścią tekstową może być wektor częstości występowania poszczególnych terminów. Zwiększanie treści tekstowej danej strony jest wtedy, dodaniem częstości występowania terminów w dokumentach z otoczenia

³ W przypadku potraktowania treści tekstowej, jako wektora częstości występowania terminów, dla danego dokumentu d_i , zwiększenie jego treści o treść tekstową dokumentu d_j , polega na zwiększeniu częstości występowania terminu t w d_i , czyli $tf_{d_i,t}$, w ten sposób, że: $tf_{d_i,t} = tf_{d_i,t} + F^i tf_{d_j,t}$

Na uniwersytecie MIT Weiss i inni opracowali system HyPursuit [Wei96] grupujący dokumenty (strony WWW), w którym wykorzystano podobieństwo hybrydowe (ang. *hybrid similarity*) $S^{hybrid}(d_i, d_j)$ między dwoma dokumentami d_i oraz d_j . Na podobieństwo hybrydowe ma wpływ zarówno struktura hipertekstowa – odsyłacze (w postaci podobieństwa odsyłaczy - $S^{links}(d_i, d_j)$) oraz treść (w postaci podobieństwa terminów - $S^{terms}(d_i, d_j)$). Wyraża się to w postaci wzoru:

$$S^{hybrid}(d_i, d_j) = \max(S^{terms}(d_i, d_j), S^{links}(d_i, d_j)).$$

Podobieństwo $S^{links}(d_i, d_j)$ uwzględnia trzy elementy, charakteryzujące strukturę hipertekstu:

- Bezpośrednia ścieżka (ang. *direct paths*) pomiędzy dwoma dokumentami – podobieństwo $S^{spl}(d_i, d_j)$. Autorzy założyli, że podobieństwo między dwoma dokumentami zmienia się w zależności od długości najkrótszej ścieżki między tymi dokumentami. Istnienie takiej ścieżki ustanawia pomiędzy nimi zależność semantyczną. Jeżeli długość ścieżki rośnie, słabnie więc semantyczna pomiędzy dokumentami. Podobieństwo $S^{spl}(d_i, d_j)$ jest więc odwrotnie proporcjonalne do długości najkrótszej ścieżki pomiędzy nimi. Ponieważ odsyłacze pomiędzy dokumentami są skierowane, należy rozważyć dwojaki rodzaje połączeń: $d_i \rightarrow d_j$ oraz $d_j \rightarrow d_i$. W przypadku nie istnienia ścieżki między d_j i d_i , wartość $S^{spl}(d_i, d_j)$ wynosi 0. Funkcja podobieństwa $S^{spl}(d_i, d_j)$ wyrażona jest wzorem:

$$S^{spl}(d_i, d_j) = \frac{1}{2^{(spl_{ij})}} + \frac{1}{2^{(spl_{ji})}},$$

gdzie: spl_{xy} – długość najkrótszej ścieżki (ang. *shortest path length*) z dokumentu d_x do d_y ,

- Liczba wspólnych przodków dokumentów (ang. *common ancestors*), tj. liczba tych dokumentów, z których istnieją odsyłacze do obu dokumentów, między którymi obliczamy podobieństwo ($S^{anc}(d_i, d_j)$). Koncepcja podobieństwa $S^{anc}(d_i, d_j)$ pochodzi z opisów bibliograficznych: jeżeli dwa lub więcej artykułów $a_1, a_2, \dots, a_n$ zawiera odesłania do wspólnego zbioru artykułów $D_1 = \{d_1, d_2, \dots, d_n\}$, wówczas rośnie semantyczna zależność pomiędzy artykułami ze zbioru D_1 . Ponadto, zgodnie z zasadą z pkt. a), zależność semantyczna pomiędzy dokumentami ze zbioru D_1 słabnie, jeżeli rośnie długość bezpośredniej ścieżki od przodka do potomka:

$$S^{anc}(d_i, d_j) = \sum_{a_x \in \text{wspólnych przodków}} \frac{1}{2^{(spl_{ia} + spl_{ja})}},$$

gdzie: $spl_{xy}^{\bar{z}}$ – długość najkrótszej ścieżki z d_x do d_y , bez przechodzenia przez dokument d_z .

Wartość $S^{anc}(d_i, d_j)$ jest następnie normalizowana tak, aby $S^{anc}(d_i, d_j) \in [0, 1]$. Waga obliczana dla każdego przodka a_x jest dzielona przez ogólną liczbę wszystkich przodków tego samego co a_x poziomu, gdzie poziom a_x w odniesieniu do d_i oraz d_j jest minimalną odległością (długością ścieżki) z dokumentu a_x do d_i lub d_j .

Jeżeli jedyna ścieżka wiodąca od a_i do d_j prowadzi poprzez d_i , wówczas a_i nie jest traktowany jako wspólny przodek d_i oraz d_j i $S^{anc}(d_i, d_j)$ nie jest wówczas obliczana. W takim przypadku, jeżeli d_i jest jedynym przodkiem d_j , wówczas wszyscy przodkowie $a_1, a_2, \dots, a_n$ dokumentu d_i są również przodkami d_j . Podobieństwo pomiędzy d_i oraz d_j jest wówczas obliczane zgodnie z $S^{spl}(d_i, d_j)$. Jeżeli istnieje inna ścieżka $a_1 \rightarrow d_j$, która nie przechodzi przez d_i , wtedy a_1 jest uważany za właściwego przodka d_i , a $S^{anc}(d_i, d_j)$ jest obliczana.

- c) Liczba wspólnych potomków (ang. *common descendants*), tj. dokumentów, do których prowadzą odsyłacze z obu dokumentów – podobieństwo $S^{dsc}(d_i, d_j)$. Podobieństwo $S^{dsc}(d_i, d_j)$ jest również proporcjonalne do liczby wspólnych potomków d_i i d_j . Sytuacja jest analogiczna do tej, przedstawionej w pkt. b):

$$S^{dsc}(d_i, d_j) = \sum_{a_x \in \text{wspólnych potomków}} \frac{1}{2^{(spl_{ix}^i + spl_{jx}^j)}}$$

Podobnie jak funkcja $S^{anc}(d_i, d_j)$, również i $S^{dsc}(d_i, d_j)$ jest normalizowana tak, aby $S^{dsc}(d_i, d_j) \in [0, 1]$.

Podobieństwo odsyłaczy jest liniową kombinacją podobieństw $S^{spl}(d_i, d_j)$, $S^{anc}(d_i, d_j)$, $S^{dsc}(d_i, d_j)$:

$$S^{links}(d_i, d_j) = W_d S^{dsc}(d_i, d_j) + W_a S^{anc}(d_i, d_j) + W_s S^{spl}(d_i, d_j)$$

Niestety autorzy nie opisali parametrów: W_a , W_s , W_d .

Funkcja podobieństwa terminów $S^{terms}(d_i, d_j)$ bazuje na zliczaniu terminów w obrębie dokumentu. Uwzględniono w niej zarówno zróżnicowanie terminów pomiędzy poszczególnymi dokumentami jak i między serwerami WWW, na których dokumenty te są składowane. Autorzy sugerują, że najlepsze podejście do funkcji obliczającej wagę terminów, powinno brać pod uwagę trzy aspekty:

- częstotliwość występowania terminu w danym dokumencie, tzw. częstotliwość terminu,
- liczbę dokumentów, w których dany termin występuje, tzw. częstotliwość dokumentów, (ang. *document frequency*)¹,
- współczynnik rozmiaru pliku, kompensujący wagę słowa w obszernych dokumentach (ang. *document size factor*).

Dodatkowo terminy mogą uzyskać większą wagę, jeżeli w dokumencie opatrzone są specjalnym atrybutem (tytuł, nagłówki itp.). Całkowita waga $w'_{d,k}$ terminu t_k w dokumencie d_i obliczana jest następująco:

$$w'_{d,k} = w_{d,k}^{tf} w_{d,k}^{ds} w_{d,k}^{at}, \quad w_{d,k}^{tf} = \left(0,5 + 0,5 \frac{tf_{d,k}}{\max_j \{tf_{d,j}\}} \right), \quad w_{d,k}^{ds} = \frac{1}{\sqrt{\sum_i (w_{d,k}^{at} w_{d,k}^{tf})^2}},$$

gdzie: $tf_{d,k}$ - częstotliwość występowania słowa t_k w dokumencie d_i ,

$w_{d,k}^{tf}$ - składowa wynikająca z $tf_{d,k}$,

$w_{d,k}^{ds}$ - składowa wynikająca z rozmiaru dokumentu d_i ,

$w_{d,k}^{at}$ - składowa wynikająca z atrybutu słowa, jest ona stała dla danego serwera WWW, autorzy przyjęli ją jako stałą dla wszystkich serwerów i wynoszącą: 10 dla słów zawartych w tytule, 5 – w nagłówku, adresie, słowach kluczowych, 1 – w ciele dokumentu.

¹ Ma to szczególne znaczenie w homogenicznych przestrzeniach informacyjnych, w których istnieje pewien zasób słów często pojawiających się w dokumentach. Przykładowo, termin „JAVA”, występuje bardzo często na stronach serwera firmy Sun Microsystems, jednak po uwzględnieniu częstotliwości dokumentów, będzie on miał małą wagę.

Funkcja podobieństwa terminów ($S^{terms}(d_i, d_j)$) pomiędzy dwoma dokumentami d_i oraz d_j jest wyrażona jako iloczyn skalarny pomiędzy dwoma wektorami terminów, reprezentującymi dwa dokumenty d_i, d_j :

$$S^{terms}(d_i, d_j) = \sum_{k=1}^{nt} w'_{d_i,k} w'_{d_j,k}.$$

Podobieństwo hybrydowe ma następujące wady:

- a) To czy dokumenty, dla których obliczane jest podobieństwo, łączy 1 czy 10 odsyłaczy nie ma odzwierciedlenia w wartości podobieństwa odsyłaczy - w obu przypadkach $S^{links}(d_i, d_j)$ będzie miało taką samą wartość. Jest to związane z tym, że w funkcji $S^{spl}(d_i, d_j)$ nie jest uwzględniana liczba najkrótszych ścieżek, a jedynie jej długość.
- b) Podobieństwo $S^{links}(d_i, d_j)$ nie uwzględnia w ogóle gęstości odsyłaczy, tzn. tego, jaka część z wszystkich odsyłaczy w obu dokumentach ma związek z ich wzajemnym połączeniem (poprzez ścieżkę, wspólnych przodków lub potomków). Jeżeli między dwoma dokumentami – mapami dwóch serwisów informacyjnych¹, z których każdy składa się ze 100 dokumentów – będzie istnieć jeden odsyłacz bezpośredni oraz jeden wspólny przodek-rodzic i jeden potomek-dziecko, wtedy podobieństwo $S^{links}(d_i, d_j)=1$. A przecież łączy je tylko jeden odsyłacz bezpośredni (z wszystkich 400 występujących na obu stronach) plus łącznie 4 odsyłacze do wspólnego rodzica i dziecka! Jeżeli natomiast między dokumentami będzie 10 odsyłaczy bezpośrednich i będą to jedyne odsyłacze występujące na obu stronach², wtedy przy braku wspólnych przodków i potomków podobieństwo $S^{links}(d_i, d_j) = W_s S^{spl}(d_i, d_j) < 1$.
- c) Podobieństwo terminów nie uwzględnia liczby dokumentów w całej kolekcji, w których dany termin występuje (częstotliwość dokumentów), co różni się z założeniami dla funkcji podobieństwa słów. Istnieje więc sprzeczność między założeniem a metodą.
- d) Zastosowanie funkcji *max* do wyznaczania podobieństwa hybrydowego jest kontrowersyjne, gdyż właściwie podobieństwo $S^{hybrid}(d_i, d_j)$ nie łączy $S^{links}(d_i, d_j)$ z $S^{terms}(d_i, d_j)$ – jest jednym lub drugim. Podobieństwo odsyłaczy nie koryguje (zwiększa lub zmniejsza) podobieństwa terminów i odwrotnie.
- e) Funkcja $S^{hybrid}(d_i, d_j)$ jest bardzo skomplikowana. Wyznaczenie jej wartości wymaga zastosowania odpowiednich algorytmów, na przykład dla ustalenia długości ścieżek spl_{xy} , $spl_{xy}^{\bar{}}$, co może okazać się być bardzo złożone obliczeniowo.

Oprócz funkcji *max* Chen w pracy [Chen97] zaproponował także dwie, inne funkcje do połączenia informacji tekstowej ze strukturalną: minimum i średnią arytmetyczną. W artykule nie przedstawiono jednak żadnych ocen tych funkcji.

3.1.4. PODSUMOWANIE

Funkcja podobieństwa powinna być dobrana zarówno z uwzględnieniem charakterystyki kolekcji grupowanych dokumentów, jak i stosownie do wybranej metody grupowania. Korzystając z wiedzy o strukturze hipertekstu, zaprezentowanej w pkt. 2 oraz hipotezy odesłania („Wprowadzenie”), można założyć, że funkcja podobieństwa powinna przede wszystkim uwzględniać strukturę

¹ mapa serwisu WWW to rodzaj spisu wszystkich stron należących do serwisu, tzn. lista odsyłaczy do wszystkich ważniejszych stron serwisu

² w takim przypadku można domniemywać, że są to dokumenty ściśle związane ze sobą

powiązań odsyłaczowych łączących dokumenty. Uniknie się wtedy problemów, które wiążą się z dokumentami pochodzącymi z otwartego środowiska hipermedialnego, jakim jest WWW:

- a) Dokumenty zawierają różne media, które niosą ważną informację o treści, którą jednak bardzo trudno wykorzystać do automatycznego przetwarzania.
- b) W systemach hipermedialnych tekst bywa redukowany jedynie do tytułów i nagłówków.
- c) Dokumenty mogą posiadać różne równoległe postacie, na przykład tekstową lub graficzną. W przypadku graficznej, tekst jest bardzo skrócony a nawet nie ma go w ogóle.
- d) W otwartym środowisku swoje dokumenty umieszczają bardzo różni autorzy, wykorzystujący różne słownictwo, różne rozumienie terminów oraz różny styl pisanie, co ma wpływ na przykład na słownictwo i długość tekstu. Autorzy również stosunkowo mało współpracują ze sobą.
- e) Poszczególne terminy mogą posiadać różne, zupełnie inne znaczenia (wieloznaczność terminów), na przykład termin „zamek” ma w języku polskim wiele odmiennych znaczeń.
- f) Dokumenty mogą posiadać równoległe wersje językowe, które – mimo jednakowej treści – przy porównaniu terminów wykazują nikłe podobieństwo.

Odsyłacze pozbawione są tych wad, gdyż:

- a) Nie zawierają tekstu w językach narodowych, są „międzynarodowe”.
- b) Dla każdego systemu hipertekstowego ich składnia jest określona; w przypadku WWW jest to język HTML i standard URL, określający postać adresu dokumentu w sieci Internet.
- c) Są niezależne od mediów występujących w dokumencie.

Przy opracowywaniu funkcji podobieństwa należy więc brać pod uwagę przede wszystkim strukturę semantyczną opisywaną przez odsyłacze hipertekstowe.

Niejednoznaczność terminów języka naturalnego stała się także podstawą do tworzenia indeksów cytowań, w których wykorzystywane są jedynie odesłania literaturowe zamieszczane w artykułach naukowych:

„Praca¹ staje się dodatkowo trudna z uwagi na dynamiczną naturę języka. Wprowadza się nowe terminy, zaś stare znikają, a także starym słowom nadaje się nowe znaczenie. (...) Innymi słowy, cytowanie jest precyzyjną niedwuznaczną reprezentacją tematu, która nie wymaga interpretacji i jest odporna na zmiany terminologii. Dodatkowo, cytowanie na trwałe zachowuje swoją dokładność.”²

Odpowiednikiem cytowań w pracach naukowych są odsyłacze w systemach hipertekstowych.

Wyżej zaprezentowano różne podejścia wykorzystujące strukturę hipertekstu, od prostych do bardzo złożonych. Proste metody wyznaczania podobieństwa bazującego na odsyłaczach nie uwzględniają wielu cennych elementów struktury, natomiast te najbardziej skomplikowane (Weiss i inni [Wei96]) są bardzo złożone obliczeniowo, a i tak nie wykorzystują wszystkich ważnych cech charakterystycznych dla systemów hipertekstowych. Postać funkcji podobieństwa zależnego od struktury należy więc dostosować do metody grupowania, w której będzie ona wykorzystana.

Pojawia się tutaj problem: czy przy określaniu podobieństwa bazującego na strukturze hipertekstowej uwzględniać tylko odsyłacze łączące dwa dokumenty, dla których obliczamy podobieństwo, czy także dłuższe ścieżki je łączące, jak również wspólnych przodków lub potomków (pkt. 3.1.1, 3.1.3). Jego rozwiązanie jest uzależnione od przyjętej metody grupowania. W tej, proponowanej w niniejszej pracy jest wykorzystywana wartość maksymalnego przepływu między wę-

¹ związana z indeksowaniem tematycznym na podstawie terminów, przyp. tłum.

² [Gar79, s.3], tłum. autora.

złami-dokumentami. Poprzez ten przepływ (już w samej metodzie) uwzględniane jest podobieństwo „przechodzące przez ścieżki” łączące dwa dokumenty. W związku z tym podobieństwo wynikające ze struktury powinno uwzględniać przede wszystkim wzajemne powiązania odsyłaczowe i ewentualnie najbliższych wspólnych przodków lub najbliższych wspólnych potomków, bez ścieżek. O ile jednak, w zastosowaniu metody grupowania przy wyszukiwaniu informacji, które wymaga sprowadzenia pełnej treści stron, ustalenie wspólnych potomków jest stosunkowo łatwe (wystarczy odnaleźć takie same odsyłacze w obu dokumentach¹), o tyle ustalenie wspólnych przodków jest utrudnione, gdyż konieczne jest wyszukanie i sprowadzenie nowych dokumentów – przodków.

W przeciwieństwie do drukowanych tekstów naukowych, zawierających cytowania, otwarte środowisko WWW jest zmienne w czasie. Powoduje to powstawanie niespójności w strukturze hipertekstowej – odsyłacze kierują do dokumentów już (lub jeszcze) nieistniejących². Tego typu odsyłacze nie mają oczywiście wpływu na podobieństwo określane poprzez wzajemne odesłania między dwoma dokumentami (oba dokumenty istnieją, gdyż należą do kolekcji grupowanych dokumentów). W przypadku odsyłaczy do wspólnych potomków - dzieci fakt, czy ów potomek rzeczywiście istnieje, czy może dopiero będzie istniał nie osłabia podobieństwa dokumentów. Nie jest przecież brana pod uwagę treść zawarta w potomku, a jedynie to, że dwa dokumenty odnoszą się do tego samego, innego dokumentu.

Nie powinno się jednak rezygnować z wykorzystania innych – poza strukturą – elementów dokumentów, niosących informację o temacie. Dotyczy to zwłaszcza treści tekstowej. Ponieważ jednak struktura jest traktowana jako zasadniczy nośnik, więc tekst może jedynie wpłynąć na poprawę (zwiększenie) podobieństwa. Można tutaj wykorzystać rezultaty badań z dziedziny porównywania tekstów bazujących na występujących w dokumentach terminach, tj. częstość występowania terminów w dokumentach i całej kolekcji, długość dokumentów itp.

W wypadku braku podobieństwa wynikającego ze struktury, należy je oprzeć na innych miarach (tekście), gdyż w dużych, otwartych systemach hipertekstowych (WWW) nie jest możliwe, aby istniało niezerowe podobieństwo struktury, między wszystkimi, tematycznie związanymi dokumentami³.

Niektóre dokumenty hipertekstowe, takie jak strony WWW, mają swoją specyfikę związaną z organizacją elementów opisujących zawartość. Dotyczy to przede wszystkim: tytułu dokumentu, jego opisu oraz słów kluczowych. Przy obliczaniu podobieństwa wynikającego z treści tekstowej należałoby w szczególny sposób uwzględnić terminy występujące w tych elementach.

3.2. METODY I ALGORYTMY GRUPOWANIA. WYMAGANIA OBLICZENIOWE I PAMIĘCIOWE

W procesie grupowania (w przeciwieństwie do klasyfikacji) liczba grup dokumentów U nie jest z góry zadana, a kryteria przyporządkowywania elementów do grup mogą zmieniać się dynamicznie⁴. Wymaga to dużych wymagań obliczeniowych, ale niesie też nowe możliwości rozpoznawania takiego zbioru grup, który odzwierciedlałby możliwie najlepiej specyfikę przetwarzanych danych. Wygenerowana struktura grup zależy od zastosowanej metody grupowania, dla której może istnieć możliwość wyboru algorytmu lub sposobów implementacji. Wybór metody decyduje o wyniku końcowym, tzn. o tym, ile grup zostanie utworzonych i jakie one będą. Natomiast wybór

¹ W przypadku WWW odsyłacze zawarte są w opisie strony, wraz z tekstem.

² Według badań przeprowadzonych Etzioni i Selberga [Etz95] aż 15% odsyłaczy podawanych jako odpowiedzi przez wyszukiwarki internetowe jest już nieaktualna.

³ Autorzy nie są w stanie umieszczać na każdej stronie odsyłaczy do wszystkich dokumentów o podobnej treści.

⁴ Na przykład, maksymalna wielkość grupy i współczynniki metody mogą być wyznaczane na podstawie liczby dokumentów.

algorytmu (będąc implementacją metody) wpływa przede wszystkim na szybkość działania metody.

Wymagania na zasoby niezbędne do uruchomienia danej metody grupowania stają się istotnym problemem w przypadkach, gdy rozmiar danych do przetworzenia jest duży. Głównym składnikiem, wpływającym na czas przetwarzania, są obliczenia podobieństw pomiędzy parami dokumentów. W przypadku najprostszych metod grupowania, złożoność czasowa – obliczeniowa (ang. *time requirements*) jest co najmniej rzędu $O(NU)$. Dla bardziej złożonych metod, które na przykład wymagają obliczenia całej macierzy podobieństw¹, złożoność czasowa rośnie do rzędu $O(N^2)$. Złożoność pamięciowa (ang. *storage* lub *space requirements*) jest rzędu $O(N)$, jeżeli zapamiętywany jest zbiór dokumentów wejściowych. W przypadku wykorzystywania macierzy podobieństw złożoność pamięciowa jest rzędu $O(N^2)$. Dla bardzo dużych kolekcji dokumentów (duża wartość N), macierz podobieństw musi być zapamiętana na dysku (a nie w pamięci operacyjnej komputera). Jeśli nie ma możliwości składowania macierzy (na przykład na dysku brak miejsca²), niezbędne jest alternatywne rozwiązanie. Polega ono na obliczaniu poszczególnych podobieństw za każdym razem, gdy są one potrzebne do ustalenia bieżącej, najbardziej podobnej pary dokumentów. To z kolei zwiększa złożoność czasową o rząd N^2 [Ras92].

„Z powodu dużych wymagań czasowych i pamięciowych wiele prac nad grupowaniem w wyszukiwaniu informacji ograniczało się do małych zbiorów dokumentów, zazwyczaj tylko kilkuset. Jednakże zwiększenie możliwości obliczeniowych i pojemności pamięci komputerów oraz zastosowanie efektywnych algorytmów w implementacji niektórych metod grupowania i poszukiwania najbliższego sąsiada uczyniło możliwym przeprowadzanie grupowania nawet na bardzo dużych zbiorach dokumentów. Jak wskazują badania, istnieje wysoki stopień równoległości w obliczaniu wartości podobieństw, w związku z czym równoległa architektura sprzętowa pozwala potencjalnie na dalszą poprawę efektywności przetwarzania”^{3,4}.

3.3. PODZIAŁ METOD GRUPOWANIA DOKUMENTÓW

Metody grupowania są zazwyczaj klasyfikowane ze względu na otrzymywane przy ich zastosowaniu wyniki, tj. strukturę grup. W związku z tym, biorąc pod uwagę **cel grupowania**, można wyróżnić metody (wg [Kaz00a]):

- hierarchiczne,
- niehierarchiczne.

Proste metody niehierarchiczne dzielą zbiór N obiektów na U grup, w których nie jest dopuszczalne pokrycie – nakładanie (ang. *overlapping, covering*). Tego rodzaju metody nazywane są metodami podziału – dzielącymi (ang. *partitioning*). Każdy element należy do grupy, do której jest najbardziej podobny. Może ona być reprezentowana przez centroid⁵ lub przez tzw. przedsta-

¹ Macierz podobieństw jest kwadratową macierzą, w której pamiętane są wartości podobieństw, dla każdej pary dokumentów d_i i d_j . Jeżeli podobieństwo jest funkcją symetryczną, wtedy wystarczy zbudować dolną (lub górną) macierz trójkątną.

² Sytuacja taka może wystąpić przy grupowaniu setek milionów dokumentów zebranych w bazach wyszukiwarki internetowej.

³ [Ras92, s.424], tłum. autora.

⁴ Porównanie grupowania hierarchicznego przy przetwarzaniu równoległym i liniowym można znaleźć w pracy Allena, Obry i Littmana [All93]. Dla równoległych obliczeń szybkość była nawet prawie 10-krotnie większa.

⁵ Centroidem może być na przykład połączenie tekstów wszystkich dokumentów należących do grupy. W modelu wektorowym, natomiast, będzie to zwykle wektor, dla którego suma odległości od wszystkich dokumentów z grupy jest najmniejsza.

wiciela grupy¹, pokazującego charakterystykę elementów należących do grupy. Inne metody dopuszczają przyporządkowanie tego samego dokumentu do kilku grup, czyli tzw. pokrywanie się grup. Biorąc powyższe pod uwagę, można wyróżnić [Kaz00a]:

- metody pokrycia (zachodzących na siebie grup),
- metody podziału (rozdzielnych grup).

Metody hierarchiczne są bardziej złożone i w efekcie ich działania powstają zagnieżdżone zbiory danych. Ze względu na **sposób grupowania**, metody hierarchiczne mogą być [Kaz00a]:

- skupiające² (ang. *agglomerative*); polegają na łączeniu dokumentów ($N-1$ połączeń), zaczynając od zbioru nie pogrupowanych dokumentów;
- dzielące³ (ang. *divisive*); na początku wszystkie elementy znajdują się w jednej grupie, która przechodzi $N-1$ podziałów na mniejsze grupy.

3.4. METODY NIEHIERARCHICZNE

Większość ich jest z natury heurystyczna, ponieważ wymaga wcześniejszej decyzji, o ilości grup, ich rozmiarze, kryteriach przynależności do grupy oraz reprezentacji grup. Duża liczba możliwych podziałów N elementów na U grup czyni optymalne rozwiązanie bardzo trudnym do uzyskania. W związku z tym w metodach niehierarchicznych celem jest zwykle znalezienie możliwie dobrego przybliżenia. Osiąga się to zazwyczaj poprzez dzielenie kolekcji na pewne, wstępne grupy, a następnie zmianę przyporządkowania dokumentów, aż do osiągnięcia zadanych kryteriów. Liczba operacji wykonywanych w metodach niehierarchicznych (NU) jest najczęściej mniejsza niż w metodach hierarchicznych (N^2), gdyż $U \ll N$. Tak więc metody niehierarchiczne mogą być wykorzystywane dla bardzo dużych zbiorów dokumentów. Większość wczesnych metod grupowania była niehierarchiczna, gdyż możliwości obliczeniowe maszyn były wtedy mocno ograniczone. Przykładem jest projekt SMART opisany w pracy [Sal75a].

3.4.1. METODY JEDNEGO PRZEBIEGU

Metody jednego przebiegu (ang. *single pass methods*) są bardzo proste, ponieważ wymagają, aby zbiór dokumentów był przetwarzany tylko jeden raz. Ogólny algorytm jest dla nich następujący [Ras92]:

1. Niech pierwszy dokument d_1 będzie reprezentantem grupy Z_1 ,
2. Dla kolejnego dokumentu d_i oblicz jego podobieństwo S do reprezentanta każdej istniejącej grupy,
3. Jeżeli maksymalna wartość podobieństwa S , obliczona w kroku 2, jest większa niż pewna stała wartość progowa τ , dodaj dokument d_i do odpowiedniej grupy (dla której wartość S jest największa) i oblicz na nowo jej reprezentanta; w przeciwnym wypadku użyj d_i do utworzenia nowej grupy,
4. Powrót do kroku 2.

Wadą metody jednego przebiegu jest to, iż często grupy utworzone jako pierwsze są większe, niż grupy wygenerowane na końcu. Ponadto formowane grupy są mocno zależne od porządku, w

¹ W modelu wektorowym przedstawicielem (reprezentantem) może być taki dokument należący do grupy, dla którego odległość do każdego innego dokumentu z grupy jest najmniejsza.

² w literaturze, na przykład [Kuch82], można spotkać także tłumaczenie „metody aglomeracyjne”

³ na ich określenie używa się także terminu „metody podziału”

jakim jest przetwarzany zbiór dokumentów. Z tego względu metoda ta, przed uruchomieniem metody ponownego przydziału jest czasami używana do wstępnego przetwarzania.

Jednym z przykładów realizacji metody jednego przebiegu jest algorytm k -środków (ang. *k-means*). Polega on na wyborze k środków – centroidów (najczęściej są to losowo wybrane dokumenty) i przyporządkowywaniu poszczególnych dokumentów do najbardziej do nich podobnego centroidu¹. Zwykle, co zadaną liczbę przyporządkowań, następuje ponowne określenie środków – centroidów [Kaz00a]. Metoda ta ma tę zasadniczą wadę, że liczba klas jest w niej ustalona z góry: k . Wulfekuhler i Punch [Wul97] zastosowali metodę k -środków do grupowania słów pochodzących ze stron WWW (proces grupowania był powtarzany 10 razy).

3.4.2. METODY PONOWNEGO PRZYDZIAŁU

Idea działania metod ponownego przydziału (ang. *reallocation methods*) polega na dokonaniu pewnego, wstępnego podziału dokumentów, a następnie przeniesieniu dokumentów z grupy do grupy, w celu poprawienia tego podziału. Ogólny algorytm działania takiej metody jest następujący [Ras92]:

1. Wybierz U centroidów (reprezentantów) grup,
2. Od $i=1$ do N , przypisz dokument d_i do najbardziej podobnego centroidu (reprezentanta),
3. Od $j=1$ do U , oblicz na nowo centroid grupy Z_j ,
4. Powtarzaj kroki 2, 3 aż nie ma już lub jest bardzo niewiele zmian przypisania dokumentów do grup podczas kolejnego przebiegu.

Metody jednego przebiegu i ponownego przydziału były używane do wyszukiwania informacji od początku prac w dziedzinie grupowania, klasycznym przykładem jest projekt SMART opisany w pracach [Sal75a, Dąb78]. Wymagania czasowe i pamięciowe metod ponownego przydziału są znacznie mniejsze niż dla metod hierarchicznych i można przy ich użyciu przetwarzać znacznie większe zbiory danych. Obecnie, ze względu na wzrost możliwości obliczeniowych oraz poprawę efektywności algorytmów zwykle bardziej preferowane są jednak metody hierarchiczne.

3.4.3. METODA „ROZRZUC I ZBIERZ”

Metoda „rozrzuc i zbierz” (ang. *scatter/gather*) została opracowana w ośrodku naukowym firmy Xerox [Cut92, Cut93]. Obejmuje ona więcej niż tylko proces grupowania. Ma na celu wspomóc użytkownika w wyszukiwaniu dokumentów, zwłaszcza wtedy, gdy nie potrafi on sformułować precyzyjnego zapytania, bądź nie szuka informacji na konkretny temat, a tylko przegląda kolekcję dokumentów w poszukiwaniu „czegoś interesującego”.

Idea metody polega na tym, żeby podzielić kolekcję dokumentów na pewne grupy tematyczne i przedstawić opis powstałych grup użytkownikowi (faza rozrzucania). Analizując zawartość grup na podstawie ich opisu, użytkownik może wybrać te grupy, które uzna za interesujące (spełniające jego potrzeby informacyjne). W ten sposób powstaje tzw. podkolekcja dokumentów (faza zbierania). Proces ten może być powtarzany wielokrotnie, aż do momentu uzyskania przez użytkownika zadowalającego go zbioru dokumentów.

Rozrzucanie zostało zrealizowane przy użyciu algorytmu grupowania niehierarchicznego przez tzw. rozdrabnianie (ang. *fractionation*). Algorytm ten polega na wydzieleniu z kolekcji pewnej liczby k centroidów. W tym celu cały zbiór N dokumentów jest dzielony na N/m podkolekcji, o ustalonej liczbie dokumentów m (przy czym m jest większe od k). Następnie, w każdej podkolekcji stosowana jest hierarchiczna metoda skupiająca średniej grupowej (pkt. 3.5.4), aż do momentu uzyskania k grup (a ściślej, centroidów tych grup). Kolejny krok polega na podzieleniu

¹ Podprogram, napisany w języku Fortran, realizujący grupowanie metodą k -środków jest dostępny pod adresem: <http://lib.stat.cmu.edu/apstat/136>

kolekcji na grupy przez przyporządkowanie wszystkich dokumentów z kolekcji do otrzymanych centroidów. Do tego celu zastosowano algorytm najbliższego sąsiedztwa (ang. *assign-to-nearest*), który polega na przypisywaniu poszczególnych dokumentów do najbliższego centroidu¹. Dla poprawienia jakości grupowania dodany został mechanizm udoskonalania (ang. *refinement*), na który składają się trzy metody: powtórzeniowy algorytm najbliższego sąsiedztwa (ang. *iterated assign-to-nearest*), metoda podziału (ang. *split*) oraz metoda łączenia (ang. *join*).

Powtórzeniowy algorytm najbliższego sąsiedztwa polega na tym, że dla danej grupy obliczany jest jej centroid na podstawie tylko części dokumentów o największym, wzajemnym podobieństwie. Następnie, wszystkie dokumenty są ponownie przydzielane do nowo powstałych centroidów, tworząc w ten sposób nowe, zmodyfikowane grupy (koncepcja podobna do metody ponownego przydziału – pkt. 3.4.2).

Metoda podziału polega na podzieleniu na dwie podgrupy tych grup, w których nie jest spełnione kryterium jednorodności, na przykład średnie podobieństwo dokumentów parami lub średnie podobieństwo dokumentów do centroidu grupy nie przekracza przyjętego progu.

Z kolei, w metodzie łączenia agregowane są te grupy, które nie są dostatecznie odległe od siebie. Kryterium odległości stanowią tzw. słowa kluczowe (ang. *topical words*) dla grupy, tzn. te, które najczęściej występują w dokumentach z grupy. Jeżeli liczba wspólnych słów kluczowych w dwóch grupach przekracza przyjęty próg, grupy te są łączone. Słowa kluczowe wykorzystano także do tworzenia opisów słownych utworzonych grup. Dzięki temu, użytkownikowi prezentowane jest streszczenie każdej grupy (ang. *cluster digest*), na podstawie którego może on dokonać „zbierania” interesujących go tematów.

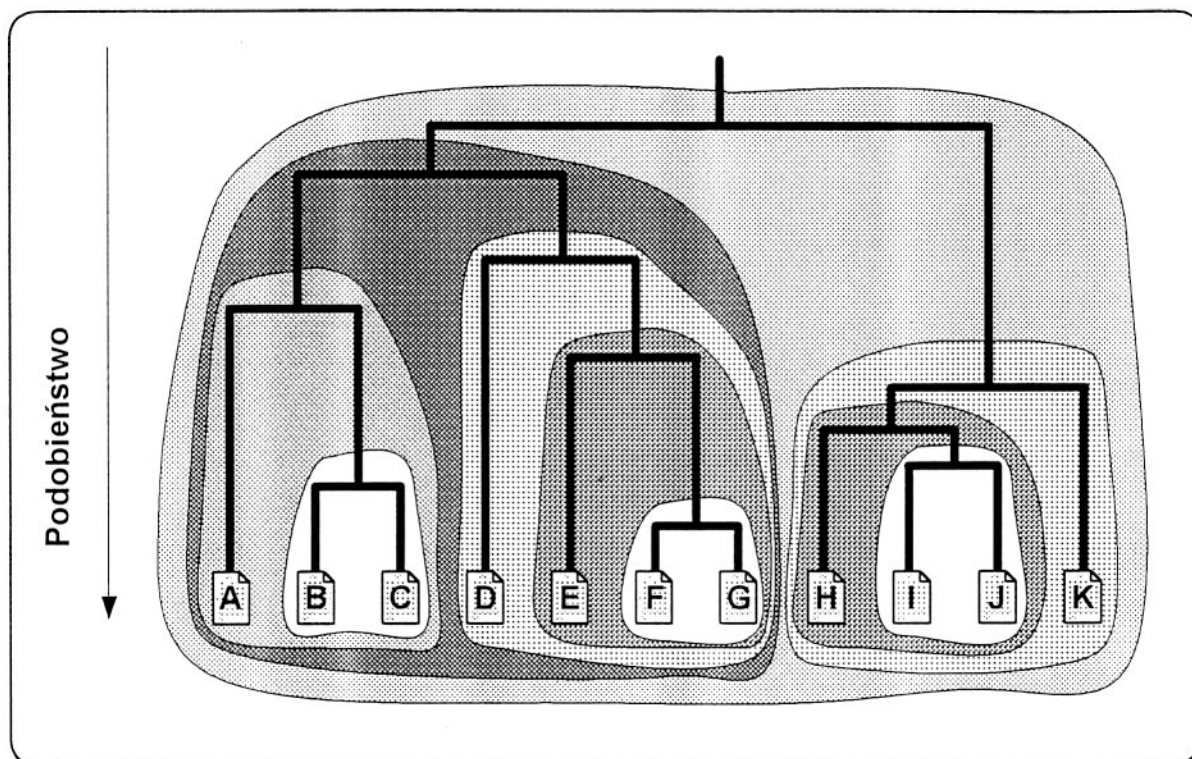
Według autorów problem stanowi złożoność obliczeniowa metody. Dzielenie kolekcji, znajdowanie centroidów, grupowanie oraz poprawa grupowania wymagają zastosowania efektywnych algorytmów tak, aby system mógł reagować w czasie, który byłby do zaakceptowania dla użytkownika.²

3.5. METODY HIERARCHICZNE

W początkowej fazie rozwoju metod grupowania szeroko stosowano i rozwijano metody hierarchiczne w różnych dziedzinach, jednak nie dotyczyło to wyszukiwania informacji. Powodem takiego stanu rzeczy były mocno ograniczone możliwości ówczesnych maszyn obliczeniowych. Wraz ze wzrostem wydajności komputerów i efektywności algorytmów, prace nad grupowaniem w wyszukiwaniu informacji skoncentrowały się na hierarchicznych metodach skupiających (ang. *hierarchical agglomerative clustering methods* - HACM). W efekcie zastosowania metod z grupy HACM powstaje hierarchia - dendrogram (rys. 3.1).

¹ Jest to w istocie odmiana algorytmu *k*-środków (pkt. 3.4.1), chociaż autorzy o tym nie wspominają.

² Metoda „rozrzucić i zbierać” została przetestowana na zbiorze 5000 dokumentów (zajmujących łącznie ponad 30MB) będących tekstami wiadomości z serwisu gazety *New York Times* [Cut92]. Następne badania przeprowadzono dla 700tys. dokumentów kolekcji TREC/Tipster-2 (2.1 GB) [Cut93] oraz zbioru miliona dokumentów TREC/Tipster-3 (ponad 3 GB).



Rys. 3.1. Dendrogram.

Najczęściej wykorzystywane metody skupiające¹ (wg [Ras92]), to:

- a) metoda najbliższego sąsiedztwa² (ang. *single link, nearest neighbor*),
- b) metoda najdalszego sąsiedztwa (ang. *complete link, furthest neighbor*),
- c) metoda średniej grupowej (ang. *group average link*³),
- d) metoda Warda,
- e) metoda centroidalna (ang. *centroid method*⁴),
- f) metoda medianowa (ang. *median method*⁵).

Z badań porównawczych przeprowadzonych przez Burgina [Bur95] dla 4 pierwszych, wymienionych wyżej metod wynika, że w zastosowaniach z dziedziny wyszukiwania informacji, najmniej przydatna jest metoda najbliższego sąsiedztwa. Potwierdza to wnioski z prac Griffithsa i innych [Gri84, Gri86].

3.5.1. OGÓLNY ALGORYTM DLA METOD HIERARCHICZNYCH

Większość hierarchicznych metod skupiających może być opisana następującym algorytmem:

1. Rozpoznań dwa najbliższe dokumenty i utwórz z nich grupę.

¹ Nazwy metod hierarchicznych zostały zaczerpnięte z pracy Kucharskiego [Kuch82]. Tam także znajdują się procedury, napisane w języku Algol, realizujące wybrane algorytmy dla poszczególnych metod.

² „W literaturze polskiej metoda ta jest znana pod nazwą *taksonomii wrocławskiej*” [Kuch82, s.34]

³ W terminologii angielskiej używany jest także termin: *unweighted pair-group method using average* – UPGMA.

⁴ W terminologii angielskiej używany jest także termin: *unweighted pair-group method using the centroid average* – UPGMC.

⁵ W terminologii angielskiej używany jest także termin: *weighted pair-group method using centroid average* – WPGMC.

2. Rozpoznaj i połącz dwa, kolejne, najbliższe sobie obiekty, przy czym obiektem może być zarówno nieprzynależny do żadnej grupy dokument jak i cała grupa¹.
3. Jeżeli pozostaje więcej niż jedna grupa, powrót do kroku 1.

Poszczególne metody skupiające różnią się sposobem definiowania najbardziej podobnej pary obiektów oraz sposobem reprezentacji obiektu-grupy². Lance i Williams [Lan66] zaproponowali ogólną formułę dla obliczania odległości pomiędzy nową grupą, reprezentowaną przez centroid $Z_{i,j}$ (powstałą z połączenia dwóch grup Z_i oraz Z_j) a pozostałymi istniejącymi grupami. Wykorzystuje ona odległość opartą na wartościach odległości sprzed utworzenia nowej grupy³. Odległość $d_{Z_{i,j}Z_k}$ pomiędzy nową grupą $Z_{i,j}$, a każdą inną, istniejącą po połączeniu Z_k , wyraża się wzorem:

$$d_{Z_{i,j}Z_k} = a_1 d_{Z_iZ_k} + a_2 d_{Z_jZ_k} + b d_{Z_iZ_j} + g \text{ card}(d_{Z_iZ_k} - d_{Z_jZ_k})$$

Równanie to może być modyfikowane i tym samym dopasowane do różnych metod skupiających poprzez dobór wartości współczynników a_1 , a_2 , b oraz g . Tabela 3.1 zawiera postaci parametrów Lance-Williamsa oraz centroidów (reprezentantów grupy), dla wspomnianych wyżej, hierarchicznych metod skupiających⁴.

Metoda HACM	Parametry Lance-Williamsa	Centroid grupy
Metoda najbliższego sąsiedztwa	$a_1 = a_2 = \frac{1}{2}, \quad b = 0, \quad g = -\frac{1}{2}$	-
Metoda najdalszego sąsiedztwa	$a_1 = a_2 = \frac{1}{2}, \quad b = 0, \quad g = \frac{1}{2}$	-
Metoda średniej grupowej	$a_1 = \frac{m_i}{m_i + m_j}, \quad a_2 = \frac{m_j}{m_i + m_j}, \quad b = 0, \quad g = 0$	-
Metoda Warda	$a_1 = \frac{m_i + m_k}{m_i + m_j + m_k}, \quad a_2 = \frac{m_j + m_k}{m_i + m_j + m_k},$ $b = -\frac{m_k}{m_i + m_j + m_k}, \quad g = 0$	$C_{Z_{i,j}} = \frac{m_i C_{Z_i} + m_j C_{Z_j}}{m_i + m_j}$
Metoda medianowa	$a_1 = a_2 = \frac{1}{2}, \quad b = -\frac{1}{4}, \quad g = 0$	$C_{Z_{i,j}} = \frac{C_{Z_i} + C_{Z_j}}{2}$
Metoda centroidalna	$a_1 = \frac{m_i}{m_i + m_j}, \quad a_2 = \frac{m_j}{m_i + m_j}, \quad b = \frac{m_i m_j}{(m_i + m_j)^2},$ $g = 0$	$C_{Z_{i,j}} = \frac{m_i C_{Z_i} + m_j C_{Z_j}}{m_i + m_j}$

Tab. 3.1. Parametry równania Lance-Williams'a: m_i jest liczbą elementów w grupie Z_i .

¹ W kroku tym może następować połączenie dwóch dokumentów, dwóch grup albo przyłączenie dokumentu do istniejącej grupy.

² W niektórych metodach reprezentantem grupy jest centroid, który ulega zmianie przy każdorazowym dodaniu nowego obiektu do grupy. W innych metodach (najbliższego, najdalszego sąsiedztwa, średniej grupowej) reprezentant grupy jest uzależniony od innego obiektu, do którego obliczane jest podobieństwo grupy. Jest to, na przykład, najbliższy (najbardziej podobny) danemu obiektowi dokument z grupy.

³ Dzięki temu możliwe jest proste przeliczenie macierzy podobieństw, jeśli jest to wymagane w danej metodzie.

⁴ Opis programów napisanych w języku C, z pakietu XploRe, realizujących ogólny algorytm dla metod HACM, jest dostępny pod adresem: <http://www.galaxy.gmu.edu/~xplore/help/agglom.html>, a koncepcja metod HACM pod adresem <http://www.galaxy.gmu.edu/~xplore/tutorials/clustnode3.html>. Źródła analogicznego programu napisane w języku Fortran znajdują się pod adresem: <http://lib.stat.cmu.edu/multi/hcl>

Andenberg wyróżnił trzy ogólne podejścia do implementacji metod skupiających [And73]. W podejściu ze składowaniem macierzy wartości powiązań pomiędzy elementami (ang. *stored matrix approach*) zapamiętywana jest macierz $N \times N$, zawierająca wartości wszystkich różnic pomiędzy parami dokumentów. Równanie Lance-Williams'a pozwala na ponowne obliczanie różnic pomiędzy centroidami przy użyciu tylko zapamiętanych wartości. Złożoność czasowa jest rzędu $O(N^2)$ i wzrasta do $O(N^3)$, jeżeli macierz jest przeszukiwana sekwencyjnie. Złożoność pamięciowa jest rzędu $O(N^2)$. W podejściu ze składowaniem danych o elementach (ang. *stored data approach*), złożoność pamięciowa wynosi zaledwie $O(N)$, ale wymaga przeliczania różnic pomiędzy parami elementów, dla każdego połączenia, co prowadzi do złożoności czasowej rzędu $O(N^3)$. W podejściu z posortowaną macierzą wartości powiązań (ang. *sorted matrix approach*) macierz jest obliczana (złożoność rzędu $O(N^2)$), a następnie sortowana ($O(N^2 \log N^2)$) przed tworzeniem hierarchii grup ($O(N^2)$). Zbiór danych nie wymaga zapamiętywania, a macierz podobieństwa jest przetwarzana sekwencyjnie, co minimalizuje ilość operacji dostępu do dysku. Jednakże to podejście jest odpowiednie tylko dla metod najbliższego i najdalszego sąsiedztwa, które nie wymagają przeliczania podobieństw podczas procesu grupowania [Ras92].

Poza ogólnym algorytmem istnieją także algorytmy specyficzne dla poszczególnych metod skupiających. Dla niektórych z nich powstały algorytmy optymalne pod względem złożoności czasowej (rzęd $O(N^2)$) oraz pamięciowej (rzęd $O(N)$). Jest to kilka algorytmów dla metody najbliższego sąsiedztwa, algorytm Defay'a dla metody najdalszego sąsiedztwa oraz algorytm znajdowania najbliższego sąsiada dla metody Warda [Ras92]¹.

3.5.2. METODA NAJBLIŻSZEGO SĄSIEDZTWA

Metoda najbliższego sąsiedztwa polega na łączeniu w każdym kroku pary najbliższych (o największej wartości wzajemnego podobieństwa), wcześniej niepowiązanych dokumentów (tzn. nie należących do tej samej grupy). Ponieważ odległość pomiędzy grupami jest zdefiniowana jako odległość pomiędzy dwoma najbardziej podobnymi dokumentami, z których każdy należy do innej grupy (wartość minimalnej odległości, która wyznacza najbliższego sąsiada), nie jest potrzebny centroid lub reprezentant grupy². Niewymagane jest także przeliczanie macierzy podobieństw podczas przetwarzania³.

Metodę tę wyjaśnijmy dla zbioru dokumentów z rys. 3.1. W pierwszym kroku algorytmu łączone są dwa najbardziej podobne do siebie dokumenty: F i G i tworzą one grupę {F,G}. W drugim kroku, w pierwszej iteracji, najbardziej podobne są dokumenty B i C (grupa {B,C}). W drugiej iteracji będą to: I oraz J (grupa {I,J}). Następnie, najbardziej podobne do siebie są dokumenty H i J. Ponieważ J jest już w grupie {I,J}, więc zostaje utworzona grupa nadrzędna dla tej grupy, czyli {H,I,J}. Dalej, najbardziej podobne są dokumenty K oraz I. Utworzona więc zostanie grupa nadrzędna do {H,I,J}, tzn. {H,I,J,K}. W kolejnych iteracjach wygenerowane zostaną grupy: {E,F,G}, {A,B,C}, {D,E,F,G}, {A,B,C,D,E,F,G} oraz grupa zawierająca wszystkie dokumenty.

Może się tak zdarzyć, że najbardziej podobnymi będzie więcej niż dwa obiekty. W takim wypadku, w zależności od realizacji algorytmu, mogą albo wszystkie one zostać połączone, albo tylko dwa z nich.

Metodę najbliższego sąsiedztwa zastosował Maarek i Shaul do zbudowania hierarchii zakładek (ang. *bookmarks*) do stron WWW, korzystając przy tym z podobieństwa treści tekstowych stron [Maa96]. Po utworzeniu hierarchii, następował proces jej burzenia, do czego wykorzystano technikę plastrów (ang. *slicing technique*). Technika ta polega na podzieleniu zbioru wartości podo-

¹ Równoległe algorytmy realizujące poszczególne metody HACM przedstawił Olson w pracy [Ols95].

² Właściwie jako reprezentanta, względem danego dokumentu, nie należącego do grupy, można traktować dokument z grupy, który jest najbliższy danemu dokumentowi.

³ W metodach, w których wykorzystywana jest macierz podobieństw i centroidy, konieczne jest zwykle przeliczenie macierzy podobieństw z uwzględnieniem zmienionych w kolejnym kroku centroidów.

bieństw dokumentów na przedziały - plastry. Grupy, które należą do tego samego plastra, są następnie łączone.

Wiele spośród algorytmów rozwiniętych dla tej metody nie odpowiada zastosowaniu w wyszukiwaniu informacji, gdzie zbiory dokumentów charakteryzują się dużą liczbą elementów oraz wysoką różnorodnością [Ras92]. Najbardziej znane algorytmy to: algorytm SLINK [Sib73]¹, algorytm minimalnego drzewa rozpinającego² Prima-Dijkstry [Dij78]³ oraz algorytm van Rijsbergena [Rij71]. Wszystkie one mają złożoność obliczeniową rzędu $O(N^2)$ oraz pamięciową, rzędu $O(N)$ – nie jest konieczne zapamiętywanie w nich macierzy podobieństw. Metoda najbliższego sąsiedztwa wydaje się być więc interesującą zarówno pod względem złożoności obliczeniowej, jak i pamięciowej.

Metoda najbliższego sąsiedztwa jest szeroko stosowana, aczkolwiek ma tendencję do formowania dużych, słabo odseparowanych od siebie grup.

3.5.3. METODA NAJDALSZEGO SĄSIEDZTWA

Metoda najdalszego sąsiedztwa polega na łączeniu w kolejnych krokach dwóch obiektów (obiektem jest pojedynczy dokument lub cała grupa) z wykorzystaniem maksymalnej odległości. W przypadku dwóch pojedynczych dokumentów, maksymalna odległość między nimi jest wartością podobieństwa. Jeżeli obliczana jest odległość między dwoma grupami, wtedy ważna jest największa między nimi odległość, z których jeden należy do pierwszej grupy, a drugi - do drugiej. W ten sposób wyznacza się dla wszystkich par grup odległość dwóch najdalszych sąsiadów. Połączeniu w kolejnym kroku podlegają te dwie grupy (dokumenty), dla których odległość maksymalna jest najmniejsza.

Metoda najdalszego sąsiedztwa nazywana jest kompletną (ang. *complete*), ponieważ wszystkie dokumenty w danej grupie są połączone ze sobą z pewnym, minimalnym podobieństwem. W efekcie tworzone są małe grupy z mocno związanymi dokumentami.

Hierarchia grup tworzona przez metodę najdalszego sąsiedztwa (małe grupy o stosunkowo mocno związanych elementach), okazała się bardzo dobra dla celów wyszukiwania dokumentów [Voo86a]. Niestety, metoda ta jest trudna do zastosowania dla dużych zbiorów danych, ponieważ nie znaleziono algorytmu bardziej efektywnego niż ogólny algorytm dla metod skupiających, w podejściu ze składowaniem danych o dokumentach lub macierzy podobieństw [Ras92]. Mimo tego, znajduje ona zastosowanie w opracowywanych metodach grupowania dokumentów hipertekstowych, czego przykładem jest praca Salampasisa i Taita [Sal99]. Metoda najdalszego sąsiedztwa jest wykorzystywana także do grupowania stron WWW, jak na przykład w systemie HyPersuit [Wei96] oraz w systemie Jasper⁴ opracowanym w firmie British Telecom [Dav96].

Najlepiej znanym algorytmem dla metody najdalszego sąsiedztwa jest CLINK, opracowany przez Defay'a [Def77]. Ze względu na problemy z uzyskaniem wszystkich poziomów hierarchii, dla zastosowań z dziedziny wyszukiwania informacji bardziej użyteczny wydaje się algorytm Voorhees [Voo86b]; potwierdzają to badania wykonane przez El-Hamdouchi i Willetta [Ham89].

¹ Program w języku Fortran realizujący algorytm SLINK został opisany w pracy [Ras92, s.431]

² Zbiór dokumentów jest traktowany jako graf, w którym węzłami są dokumenty, a wagi krawędzi odpowiadają wartościom podobieństwa między dokumentami, łączonymi przez krawędzie. Minimalne drzewo rozpinające dla takiego grafu (ang. *minimal spanning tree*), to drzewo rozpinające (szkielet grafu) – łączące wszystkie węzły, dla którego suma wag krawędzi jest minimalna.

³ Program w języku Fortran realizujący algorytm Prima-Dijkstry został opisany w pracy [Ras92, s.433]

⁴ dla obliczenia podobieństwa wykorzystano w nim współczynnik Dice'a

3.5.4. METODA ŚREDNIEJ GRUPOWEJ

W metodzie średniej grupowej podobieństwo pomiędzy dwoma grupami jest określone przez średnią wartość wszystkich podobieństw (odległości) pomiędzy parami dokumentów, z których każdy należy do innej z tych grup. Dzięki temu w przeciwieństwie do metod najbliższego i najdalszego sąsiedztwa, w których wyznaczana była para najbardziej lub najmniej oddalonych dokumentów – wszystkie dokumenty z obu grup mają wpływ na wartość odległości między grupami. W wyniku powstaje hierarchia pośrednia pomiędzy grupami zawierającymi dokumenty luźniej związane ze sobą z metody najbliższego sąsiedztwa oraz grupami o mocno powiązanych dokumentach z metody najdalszego sąsiedztwa.

W aplikacjach metody wykorzystywany był zmodyfikowany ogólny algorytm dla metod skupiających. Działał on jednak z dużą złożonością czasową rzędu $O(N^3)$ przy podejściu ze składowaniem danych oraz pamięciową rzędu $O(N^2)$, przy podejściu z zapamiętywaniem macierzy wartości powiązań. Jest to więc rozwiązanie nieefektywne dla dużych kolekcji dokumentów [Ras92]. Metodę średniej grupowej zastosowano do grupowania stron WWW w systemie WebYacht [Chan99].

Voorhees zaproponowała potencjalnie bardziej efektywny algorytm: podobieństwo pomiędzy wybranymi dokumentami jest obliczane na podstawie dwóch właściwie wyważonych wektorów reprezentujących te dokumenty [Voo86b]. Centroid grupy jest średnią wektorów wszystkich dokumentów należących do tej grupy. W tym przypadku podobieństwo pomiędzy centroidem grupy i danym dokumentem jest równe średniemu podobieństwu pomiędzy dokumentem i wszystkimi dokumentami w grupie. W związku z tym, centroidy mogą być używane do obliczania podobieństw pomiędzy grupami.

3.5.5. METODA WARDA

W metodzie Warda [War63] do mierzenia wartości powiązań pomiędzy dokumentami używana jest odległość euklidesowa. Grupowane dokumenty reprezentowane są przez wektory wag terminów o postaci $d_i = (w'_{d,i,1}, w'_{d,i,2}, \dots, w'_{d,i,nt})$ w przestrzeni nt -wymiarowej. Wartość $w'_{d,k}$ to waga terminu k w dokumencie d_i . Grupy¹ są łączone tak, aby w każdym kroku zmiana wartości średniego podobieństwa dokumentów wewnątrz grup (wariancja) była minimalna. Dla dwóch grup Z_i, Z_j , wzrost I_{ij} wariancji jest określony następująco:

$$I_{ij} = \frac{m_i m_j}{m_i + m_j} \Delta_{ij}^2,$$

gdzie: m_i, m_j jest liczbą elementów (dokumentów) odpowiednio w grupie Z_i i Z_j , a Δ_{ij} jest odległością euklidesową:

$$\Delta_{ij} = \sqrt{\sum_{k=1}^{nt} (w'_{d_i,k} - w'_{d_j,k})^2}.$$

Centroid grupy, po połączeniu dwóch obiektów Z_i oraz Z_j , o centroidach C_{Z_i} i C_{Z_j} , jest określony jako:

$$\frac{m_i C_{Z_i} + m_j C_{Z_j}}{m_i + m_j}$$

Jak wykazano, matematyczne własności metody Warda pozwalają zastosować do jej implementacji algorytm obustronnie najbliższego sąsiada (ang. *reciprocal nearest neighbor algorithm* - RNN) [Mur83]. Rasmussen podaje, że algorytm ten skupia elementy raczej w pewnych ograniczonych

¹ pojedynczy dokument jest traktowany jako grupa 1-elementowa

obszarach niż w prostym porządku rosnącej wartości różnic między nimi [Ras92]. Jest on nazywany algorytmem pojedynczej grupy (ang. *single cluster algorithm* - SCA), ponieważ w każdej iteracji jest wykonywane tylko jedno łączenie.

Jako rezultat zastosowania metody Warda otrzymuje się jednorodne grupy i symetryczną hierarchię [Ras92].

3.5.6. METODA CENTROIDALNA I MEDIANOWA

W metodzie centroidalnej każda grupa jest formowana i reprezentowana przez współrzędne centroidu. W każdym kroku łączona jest para grup najbardziej podobnych, przy czym podobieństwo określa się przez porównanie centroidów a nie poszczególnych dokumentów.

W metodzie centroidalnej często występuje proces „przyłączania” (ang. *chaining*), oznaczający dołączanie – w kolejnych krokach algorytmu – do dużej grupy kolejnych, pojedynczych dokumentów. W wyniku tego otrzymujemy bardziej rodzaj rankingu, niż zrównoważonej hierarchii grup.

Metoda medianowa jest podobna do poprzedniej, ale centroidy dwóch łączonych grup są ważone rozmiarem grupy. Dzięki ważeniu metoda medianowa jest bardziej odporna na efekt „przyłączania”.

W wyniku zmienności centroidów po każdym kroku, w obu tych metodach, w niektórych przypadkach może się okazać, że wartości podobieństw, na podstawie których dokumenty lub grupy są łączone, wcale nie muszą być przy każdym kroku malejące, a co więcej kolejność tych wartości nie odpowiada tym, z końcowej hierarchii. W efekcie, w niektórych miejscach hierarchii mogą wystąpić tzw. „odwrócenia” (ang. *reversals*).

3.6. GRAFOWE METODY GRUPOWANIA DOKUMENTÓW HIPERTEKSTOWYCH

3.6.1. METODA K-KRAWĘDZIOWYCH SKŁADOWYCH

Metoda grupowania oparta na wydzielaniu k -krawędziowych składowych (ang. *k-edge-components*) została zaproponowana przez Matulę w pracy [Mat77]. Następnie, Botafogo rozszerzył ją i zaadoptował dla środowiska hipertekstowego [Bot93]. W metodzie tej wykorzystuje się wyłącznie wzajemne odsyłacze hipertekstowe między dokumentami, w ogóle nie uwzględniając treści dokumentów. Na podstawie kolekcji dokumentów hipertekstowych tworzony jest niezorientowany i nieważony graf. Dokumenty traktowane są jako węzły tego grafu. Z odsyłaczy hipertekstowych, natomiast, generowane są niezorientowane i nieważone krawędzie, w ten sposób, że ze wszystkich odsyłaczy łączących dane dwa dokumenty powstaje jedna, pozbawiona wagi krawędź niezorientowana (usuwany jest zwrot odsyłacza oraz tracona jest informacja o ich liczbie).

Z grafu wydzielane są k -krawędziowe składowe. k -krawędziowa składowa $G'(V', E')$ (k -krawędziowo spójna) grafu $G(V, E)$ jest maksymalnym podgrafem sprzężonym z $G(V, E)$ ¹, posiadającym taką własność, że $G'(V', E')$ jest k -krawędziowo spójny² i nie jest $k-1$ -krawędziowo spójny. Oznacza to, istnieje taki podział k -krawędziowej składowej $G'(V', E')$ (rozdzielenie na dwie

¹ Podgraf $G'(V', E')$ grafu $G(V, E)$, jest podgrafem sprzężonym z $G(V, E)$, jeśli E' zawiera wszystkie krawędzie z E , łączące pary węzłów, należące do V' . Podgraf $G'(V', E')$ sprzężony z $G(V, E)$, posiadający pewną własność, jest maksymalny ze względu tę własność, jeżeli nie istnieje inny podgraf $G''(V'', E'')$ sprzężony z $G(V, E)$, taki że $G'(V', E') \subset G''(V'', E'')$ i $G''(V'', E'')$ także posiada daną własność [Mat77, s.102].

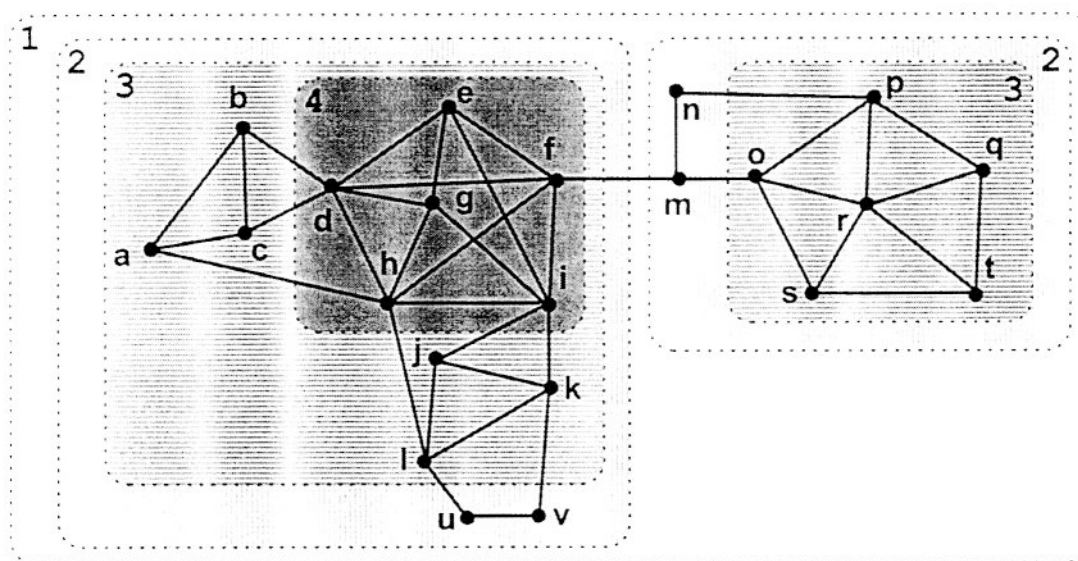
² Graf $G(V, E)$ jest k -krawędziowo spójny, jeżeli minimalna liczba krawędzi, których usunięcie powoduje utratę spójności grafu $G(V, E)$, wynosi k [Kul86, s.79, Bot93, s.118].

składowe), który zawiera k krawędzi łączących węzły składowych, a nie istnieje podział $k-1$ krawędziowy. Bazuje to na podstawowych pojęciach spójności krawędziowej i twierdzeniu Mengera.

Następnie definiowana jest k -krawędziowa agregacja, która jest zbiorem powiązanych węzłów, tworzących k -krawędziową składową, ale niebędących $(k+i)$ -krawędziową składową dla żadnego $i \in \mathbb{N}_+$.

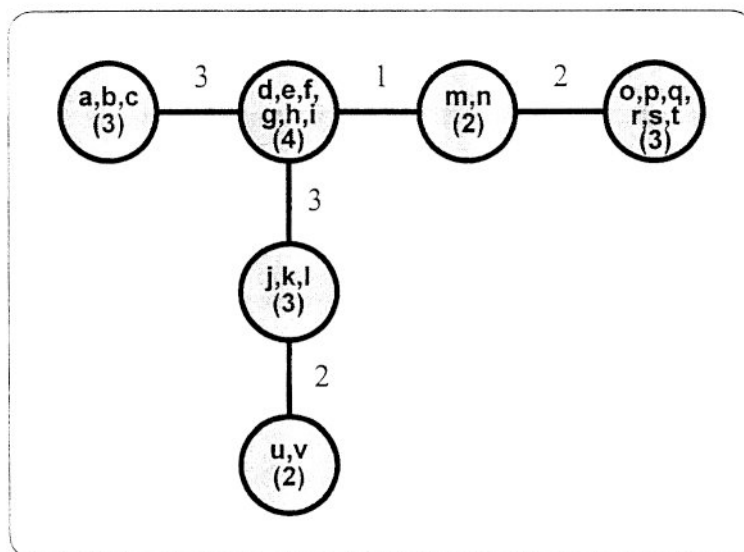
W celu wydzielenia agregacji należy:

- Zidentyfikować k -krawędziowe składowe. W grafie z rys. 3.2 zaznaczono je linią kropkowaną. Istnieją w nim 1-, 2-, 3- oraz 4-krawędziowe składowe.
- Połączyć węzły, które należą do mocniejszych składowych. Węzły $\{d, e, f, g, h, i\}$ znajdują się w 4-krawędziowej składowej, a nie znajdują się w żadnej mocniejszej składowej (5- lub więcej), czyli zostanie z nich utworzona jedna 4-krawędziowa agregacja.
- Usunąć mocniejsze składowe i związać węzły pozostałe w słabszej składowej. Po usunięciu 4-krawędziowej składowej $\{d, e, f, g, h, i\}$, w 3-krawędziowej składowej pozostały 2 agregacje: $\{a, b, c\}$ oraz $\{j, k, l\}$. Gdyby potraktować je osobno, wtedy tworzyłyby one 2-krawędziowe składowe. Ponieważ jednak pochodzą z 3-krawędziowej składowej, zatem nazwie się je 3-krawędziowymi agregacjami. Inne agregacje w tym grafie to 2-krawędziowe: $\{u, v\}$, $\{m, n\}$ oraz 3-krawędziowa: $\{o, p, q, r, s, t\}$.



Rys. 3.2. Graf z wydzielonymi k -krawędziowymi składowymi [Bot93].

Z wydzielonych agregacji można utworzyć graf zagregowany (ang. *aggregate graph*) – rys. 3.3. Liczby w nawiasach oznaczają numer agregacji, na przykład (2) - oznacza 2-krawędziową agregację. Liczby przyporządkowane krawędzom grafu zagregowanego są równoznaczne z liczbą krawędzi w grafie wejściowym, jaka łączy dane agregacje.

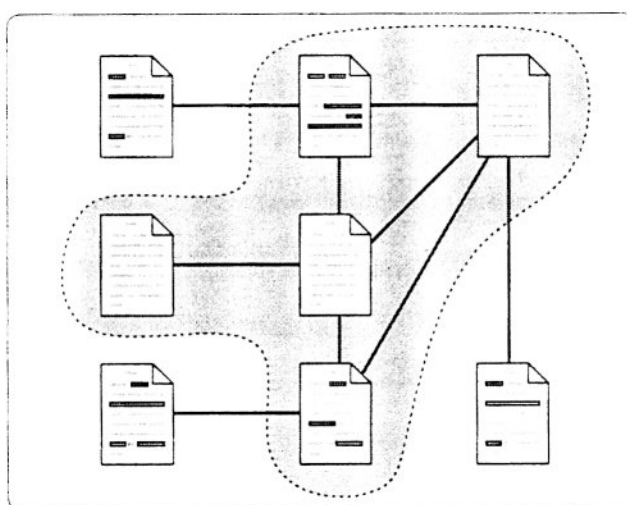


Rys. 3.3. Graf zagregowany powstały z grafu z rys. 3.2.

Metoda k -krawędziowych składowych ma zasadniczą wadę: nie uwzględnia wag krawędzi grafu. Mogą one odzwierciedlać mniejsze lub większe podobieństwo między dokumentami, wynikające na przykład z treści tekstowej lub liczby wzajemnych odsyłaczy.

3.6.2. METODA K -KLANÓW

Metoda k -klanów została zaproponowana przez Terveena, Hilla oraz Amento w pracy [Ter99]. Opracowano i zastosowano ją z myślą o miejscach WWW – całych serwisach informacyjnych (ang. *web sites*) a nie o pojedynczych stronach, tzn. węzłem grafu jest cały serwis informacyjny pochodzący z danej domeny internetowej, a krawędziami wszystkie odsyłacze prowadzące na zewnątrz z wszystkich stron należących do serwisu.



Rys. 3.4. 2-klan.

k -klan jest podgrafem, w którym:

- Każdy węzeł podgrafu jest połączony z każdym innym węzłem podgrafu ścieżką o długości k lub mniejszej.
- Wszystkie ścieżki łączące węzły podgrafu przebiegają jedynie przez węzły tego podgrafu.

Autorzy wykorzystywali jedynie 2-klany, czyli takie podgrafy, w których każda para węzłów, do nich należących, połączona jest ścieżką przebiegającą przez węzły tego podgrafu o maksymalnej długości 2 krawędzi. Na rys. 3.4 zawarty został przykład takiego 2-klanu. Liczba dwa ma na celu

uwzględnienie trzech zasadniczych rodzajów relacji między dwoma dokumentami d_i i d_j , związanych z innym węzłem d_k :

- a) $d_k \rightarrow d_i$ oraz $d_k \rightarrow d_j$. Oznacza to wspólne cytowanie przez inny dokument (ang. *co-citation*), tzn. jeżeli dwa dokumenty d_i oraz d_j są cytowane przez trzeci dokument d_k to wszystkie trzy są związane tematycznie.
- b) $d_k \leftarrow d_i$ oraz $d_k \leftarrow d_j$. Oznacza wspólne cytowanie innego dokumentu. Jeżeli dwa dokumenty d_i oraz d_j odsyłają do innego, trzeciego dokumentu d_k , wtedy wszystkie trzy są związane tematycznie.
- c) $d_i \rightarrow d_k$ oraz $d_k \rightarrow d_j$. Oznacza to, ograniczoną do dwóch krawędzi, przechodność (ang. *transitivity*), którą można interpretować jako: „jeżeli dokument d_i jest na dany temat i cytuje d_k , wtedy d_k wydaje się także być na dany temat; jeżeli d_k cytuje d_j , wtedy d_j wydaje się (choć w stopniu nieco mniejszym) także być na dany temat”.

Wymienione trzy relacje są minimalnymi 2-klanami. Należy zauważyć, że 2-klany są zdefiniowane dla grafu niezorientowanego, tzn. dwa węzły d_i i d_j są połączone, jeżeli istnieje odsyłacz z d_i do d_j lub z d_j do d_i . Uzasadnieniem takiej operacji są obserwacje autorów, w których zauważono wiele serwisów WWW - indeksów, posiadających jedynie odsyłacze do innych serwisów (ang. *index sites*) a także takie, do których inne miejsca odsyłają, traktując je jako opisujące jakieś zagadnienie (ang. *content sites*). Gdyby wymagano zawsze istnienia odsyłaczy w obie strony, tego typu miejscom nie nadano by odpowiedniego dla nich znaczenia.

W metodzie k -klanów – podobnie jak w metodzie k -krawędziowych składowych – nie uwzględniono wag krawędzi grafu.

3.7. NIEGRAFOWE METODY GRUPOWANIA DOKUMENTÓW HIPERTEKSTOWYCH

3.7.1. METODA STC DRZEWA FRAZ

Metoda drzewa fraz – STC (*Suffix Tree Clustering*) zaproponowana została przez Zamira i Etzioni w pracach [Zam98, Zam99] i zaimplementowana w systemie Grouper¹ działającym w systemie WWW. Jej zastosowanie, które przedstawili i przetestowali autorzy polega na tematycznym, niehierarchicznym grupowaniu odpowiedzi z wyszukiwarek internetowych. Metodę można podzielić na dwa zasadnicze etapy [Zam98]:

- a) Utworzenie grup bazowych (ang. *base clusters*) w postaci drzewa fraz (ang. *suffix tree*). Polega na wydzieleniu fraz (ciągu wyrazów, który w szczególności może składać się tylko z jednego słowa²) z wszystkich dokumentów i utworzeniu zbioru odwróconego, zawierającego frazę oraz listę numerów dokumentów, w której ta fraza występuje. Frazy mogą zachodzić na siebie, a nawet zawierać się w sobie nawzajem, na przykład osobną frazą może być „Ala ma kota” „nikt nie ma kota”, „ma kota”, „kota”.

Dla każdej frazy i (grupy bazowej B_i)³ obliczana jest wartości funkcji s :

$$s(B_i) = \text{card}(B_i) \cdot f(\text{len}_i),$$

gdzie: $\text{card}(B_i)$ - liczba dokumentów zawierających frazę i , len_i - długość frazy i (liczba wyrazów, bez wyrazów ze stop listy), f – pewna nieliniowa funkcja zależna od wartości len_i , która

¹ dostępnym pod adresem: <http://zhadum.cs.washington.edu/zamir/cluster.html>

² Przewidziano także zastosowanie stop listy dla słów najczęściej występujących, na przykład *previous*, *java*, *frames*, *mail* itp.

³ grupa B_i to zbiór dokumentów zawierających frazę i

zmniejsza znaczenie fraz jednowyrazowych, dla fraz od 2- do 6-wyrazowych jest liniowa, a dla fraz powyżej 6 wyrazów – stała.

- b) Utworzenie nieważonego, nieorientowanego grafu, którego węzłami są grupy bazowe. Istnienie krawędzi w grafie, łączącej grupę B_m oraz B_n , oznacza, że liczba dokumentów wspólnych w obu grupach, czyli takich, które zawierają zarówno frazę m jak i frazę n ($\text{card}(B_m \cap B_n)$), w stosunku do liczby dokumentów w każdej z tych grup bazowych, jest nie mniejsza niż pewien parametr (autorzy ustalili go na 0.5). Oznacza to, że krawędź między grupą B_m oraz B_n istnieje wtedy, gdy:

$$\frac{\text{card}(B_m \cap B_n)}{\text{card}(B_m)} > 0.5 \text{ oraz } \frac{\text{card}(B_m \cap B_n)}{\text{card}(B_n)} > 0.5.$$

Z tak utworzonego grafu wydzielane są składowe, czyli maksymalne podgrafy spójne. Pojedyncza składowa (suma logiczna zbiorów dokumentów zawartych w grupach bazowych należących do składowej) stanowi końcową grupę dokumentów.

Metoda STC ma dużą tendencję do zachodzenia grup na siebie. W przeprowadzonych przez autorów eksperymentach $\frac{3}{4}$ dokumentów znalazło się w więcej niż 1 grupie, a średnia liczba grup, w których znalazł się dokument wyniosła 2.1. Zdaniem autorów jest to zaleta metody. Twierdzą oni, że tematy objęte przez grupy mogą na siebie zachodzić.

Metoda drzew fraz posiada następujące wady:

- Duże zachodzenie na siebie grup powoduje, że w przypadku zastosowania do grupowania odpowiedzi z wyszukiwarek, użytkownik otrzymuje średnio ponad 2 razy większą kolekcję stron, co znacznie zmniejsza jego chęć do przeglądania dokumentów i zwiększa jego poczucie chaosu.
- Metoda jest wrażliwa na kolejność występowania słów w zdaniu (słowa podane w różnej kolejności utworzą dwie odrębne frazy) co, szczególnie w przypadku języka polskiego w którym panuje duża dowolność szyku wyrazów, może prowadzić do sfałszowanych wyników.
- Brak uwzględniania długości dokumentów. Dwa krótkie dokumenty posiadające tą samą na przykład 3-wyrazową wspólną frazę, są tak samo podobne do siebie, jak dwa długie dokumenty z jedną 3-wyrazową wspólną frazą. Wada ta uwidacznia się, jeżeli zastosuje się metodę do dokumentów pełnotekstowych. Eksperymenty przeprowadzone przez autorów dotyczyły kolekcji krótkich opisów dokumentów.
- Brak uwzględnienia częstości powtarzania się fraz w dokumencie. Jedno wystąpienie danej frazy jest tak samo ważne jak 100 jej wystąpień.
- W metodzie nie uwzględniono struktury hipertekstowej – odsyłaczy łączących dokumenty.
- Brak w niej parametrów, które umożliwiłyby sterowanie liczbą i wielkością grup. Na jedyny parametr (przyjęta przez autorów wartość 0.5) metoda jest - jak podano - mało wrażliwa.
- Metoda jest niehierarchiczna i dlatego nie jest możliwe ustalenie wzajemnych powiązań między grupami (nie można określić, która grupa jest szczegółowsza, a która ogólniejsza).

3.7.2. METODA K-WYCINKÓW

Metoda grupowania syntaktycznego stron WWW (ang. *syntactic clustering*), którą można nazwać metodą k -wycinków (ang. *k-shingles*), została zaproponowana przez Brodera i innych w pracy [Bro97]. Porównywane w niej są wyłącznie treści tekstowe stron, tzn. dodatkowe znaki takie, jak: formatowania języka HTML, ustawienia wielkości liter, logo, podpisy autora itp., są pomijane. Badane są dwa elementy świadczące o związkach między dwoma dokumentami: podobieństwo (ang. *resemblance*) i zawieranie się jednego w drugim (ang. *containment*).

Dla każdego dokumentu tworzony jest jego szkic (ang. *sketch*), który stanowi podstawę dla dalszego porównywania. Do utworzenia szkicu wykorzystuje się tzw. wycinki, które są fragmentami

tekstu o zadanej liczbie słów. Każdy dokument d_i ma przyporządkowany zbiór wszystkich, różnych sekwencji wycinków $K(d_i, k)$ o zadanej długości k , tzw. k -wycinków. Przykładowo, dla ciągu słów (Ala ma kota. Ala ma kota. Ala ma), zbiór wycinków o długości 3 jest następujący: {(ala, ma, kota), (ma, kota, ala), (kota, ala, ma)}. Szkic $K(d_i, k)$ jest więc listą wszystkich różnych wycinków o długości k , występujących w dokumencie d_i . Dla zadanego rozmiaru wycinka k podobieństwo S dwóch dokumentów d_i i d_j wyraża się wzorem:

$$S(d_i, d_j) = \frac{\text{card}(K(d_i, k) \cap K(d_j, k))}{\text{card}(K(d_i, k) \cup K(d_j, k))}.$$

Wartość zawierania się dokumentu d_i w d_j , czyli $C(d_i, d_j)$, jest wyrażona za pomocą:

$$C(d_i, d_j) = \frac{\text{card}(K(d_i, k) \cap K(d_j, k))}{\text{card}(K(d_i, k))}.$$

Algorytm grupowania jest następujący:

1. Obliczanie szkicu $K(d_i, k)$, dla każdego dokumentu d_i .
2. Utworzenie listy wycinków wraz z numerami dokumentów, w których się one pojawiły. Lista jest sortowana po wycinkach.
3. Utworzenie listy trójek: numery dwóch dokumentów d_i i d_j oraz liczba nk_{ij} wspólnych wycinków, występujących w dokumentach d_i i d_j , czyli: (i, j, nk_{ij}) .
4. Dokumenty, dla których parami podobieństwo S przekracza przyjęty próg, są łączone i tworzą jedną grupę.

Rozwinięciem metody jest zastosowanie tzw. super wycinków (ang. *super shingles*). Są one wtedy sortowane, a następnie znajdowane zostają dla nich super wycinki, tzn. wycinki z wycinków. Powstaje w ten sposób tzw. meta-szkic dokumentu (szkic ze szkicu). Przy odpowiednio dobranej wielkości wycinka można przyjąć, że dwa dokumenty są podobne, jeżeli mają co najmniej jeden wspólny super wycinek. Dzięki temu wyeliminowany zostaje trzeci etap powyższego algorytmu.

Wadą tej modyfikacji jest to, że stosując super wycinki, tracimy dokładność porównania dokumentów. Pojawia się także problem, w przypadku dwóch, krótkich dokumentów, dla których nie jest tworzonych zbyt wiele wycinków, przez co można nie uzyskać wspólnych super wycinków.

Ponadto super wycinki nie pozwalają wykryć zawierania się dokumentów. Jeżeli utworzymy super wycinki dla dokumentu d_i zawierającego inny - d_j , w $K(d_i, k)$ będą zawarte wycinki z d_j (bo $d_j \subset d_i$) oraz inne (pochodzące z dodatkowego tekstu). W związku z tym, super wycinki dla obu dokumentów d_i oraz d_j , będą się różnić między sobą.

Metoda ma duże wymagania pamięciowe i obliczeniowe. Mimo to autorzy przeprowadzili eksperymenty na zbiorze 30mln stron WWW, pochodzących z bazy wyszukiwarki Altavista. Zastosowano zwykłe wycinki. Przyjmując próg 0.5 dla podobieństwa S , otrzymano 3.6mln grup zawierających łącznie 12.3mln dokumentów. Z tego 2.1mln grup zawierało jedynie strony identyczne.

3.7.3. METODA KONCENTRATORÓW I AUTORYTETÓW

Metoda koncentratorów (ang. *hub*) i stron – autorytetów (ang. *authority page*) została zaproponowana przez Kleinberga i następnie rozwinięta we współpracy z ośrodkiem naukowym firmy IBM [Kle99, Chak98a, Gib98]. Zastosowano ją w systemie ARC (*automatic resource compiler*), służącym do, w pełni zautomatyzowanego, wyszukiwania stron WWW.

Strony - autorytety to takie, które zawierają dużo szczegółowych informacji na zadany temat. Koncentratory to strony, na których znajduje się duża liczba odsyłaczy do autorytetów. Idea metody opiera się na tym, że dobry koncentrator zawiera odsyłacze do wielu dobrych stron - autorytetów; odsyłacze do dobrej strony - autorytetu są umieszczone w wielu dobrych koncentratorach. Zbiór dokumentów do przetwarzania uzyskiwany jest w następujący sposób: najpierw pobierane

są strony zbioru podstawowego z wyszukiwarki Altavista, a następnie konstruowany jest tzw. zbiór rozszerzony w ten sposób, że dodawane są do niego wszystkie dokumenty, które zawierają odsyłacze do dokumentów ze zbioru podstawowego (przodkowie - rodzice) oraz te, do których prowadzą odsyłacze z dokumentów zbioru podstawowego (potomkowie - dzieci). Operacja ta jest przeprowadzana jeden [Kle99, Gib98] lub dwa razy [Chak98a], przez co w rezultacie otrzymujemy zbiór dokumentów znajdujących się w odległości do dwóch odsyłaczy od siebie. Jak wynika z przeprowadzonych testów, dla zbioru podstawowego 200 stron, liczność dokumentów w zbiorze rozszerzonym waha się od kilkuset do 3000. Jest to podejście podobne do tego, jakie przedstawili Carrière i Kazman w pracy [Car97], którzy przy wyszukiwaniu stron WWW uwzględniali także te strony, na które wskazują odsyłacze ze zbioru podstawowego (dzieci).

Dla każdego z dokumentów hipertekstowych d_i w przetwarzanym zbiorze obliczana jest wartość koncentracji $h(d_i)$ oraz wartość autorytetu $a(d_i)$. W ten sposób uzyskiwane są dwie listy dokumentów: autorytetów i koncentratorów o największych wartościach odpowiedniego parametru (autorzy ograniczali te listy do 10 [Gib98] lub 15 dokumentów [Kle99, Chak98a]).

Proces obliczania wartości $h(d_i)$ oraz $a(d_i)$ przebiega następująco:

- a) Tworzone są dwa wektory jednostkowe a oraz h o długości równej liczbie wszystkich dokumentów. Pierwszy wektor zawiera wartości dokumentów jako autorytetów, a drugi – jako koncentratorów. Następnie tworzona jest macierz W , która zawiera wartości wag powiązań pomiędzy każdymi dwoma dokumentami lub 0, jeżeli takie powiązanie nie występuje.
- b) Wagi powiązań odwzorowują potencjalne wartości odsyłaczy. Jeżeli w okolicy odsyłacza (ang. *anchor window*) do dokumentu d_i , w koncentratorze występuje dużo terminów z zadanego pytania, to d_i zawiera prawdopodobnie dużo informacji na dany temat, czyli jest dobrym autorytetem. Wagi obliczane są w następujący sposób:

$$w_{ij} = 1 + nt_{ij},$$

gdzie: w_{ij} – waga odsyłacza z d_i do d_j , nt_{ij} – liczba terminów z pytania, które wystąpiły w okolicy odsyłacza $d_i \rightarrow d_j$.

- c) Wykonywanych jest k przebiegów następujących obliczeń:

$$a = h W, \quad h = a W^T,$$

gdzie: W^T – transponowana macierz W .

Kolejne przebiegi mają na celu wyodrębnienie najlepszych koncentratorów i stron - autorytetów. Jak stwierdzają autorzy, $k = 5$ jest wystarczające dla osiągnięcia dobrych rezultatów.

Mając zbiory najlepszych koncentratorów i autorytetów, można wydzielić z nich grupy najbardziej sobie bliskich dokumentów, tzn. koncentratory odsyłające do tych samych autorytetów wraz z tymi autorytetami. Kleinberg [Kle99] proponuje tutaj zastosowanie metod podziału grafu opisane w literaturze.

Wadą metody koncentratorów i opisów jest wykorzystanie w zasadzie tylko informacji strukturalnej (tylko odsyłacze)¹ a także podział stron na tylko 2 kategorie. Wydaje się być więc ona dobrą dla porządkowania (rankingu) dokumentów, ale nie najlepszą dla grupowania.

Pewną próbę rozwiązania tego problemu zaprezentowali Bharat i Henzinger w pracy [Bha98b], wykorzystując treść tekstową dokumentów. Za pomocą współczynnika kosinusowego (3.6), obliczali oni podobieństwo między pytaniem a dokumentem, poprawiając w ten sposób dokładność odpowiedzi.

¹ jedyna informacja o treści tekstowej związana jest z liczbą nt_{ij} wystąpień terminów z pytania w poszczególnych dokumentach

3.7.4. INNE METODY GRUPOWANIA DOKUMENTÓW HIPERTEKSTOWYCH

Bernstein i inni [Ber91] zaproponowali wpisanie kolekcji hipertekstowej do struktury podobnej do tablicy, którą nazwali *linkplot*. Poprzez „wpatrywanie się” w tą strukturę próbowali zidentyfikować grupy punktów ściślej ze sobą związanych. Jednak identyfikacja grup, silnie zależy od obserwatora tablicy *linkplot*. Poza tym, zmieniając ponumerowanie węzłów hipertekstowych – dokumentów, można otrzymać zupełnie inny podział na grupy

Guillaume i Murtagh [Gui00] opracowali metodę grupowania hipertekstowych dokumentów XML (*Extensible Markup Language*), którą przetestowali dla języka AML (*Astronomical Markup Language*). W standardzie XML zawarty jest specjalny język opisu odsyłaczy XLink (*XML Linking Language*), w którym każde odesłanie może zawierać rozbudowany opis, na przykład rolę jaką ono pełni w systemie [XLINK00]. Dla kolekcji hipertekstowej budowany jest nieorientowany graf. Węzły odpowiadają w nim dokumentom, a krawędzie istnieniu odsyłaczy między dokumentami łączonymi przez krawędź. Krawędzie są ważone w oparciu o opisy zawarte w odsyłaczach. Do grafu stosowany jest następnie algorytm szumowego dzielenia (ang. *noising partitioning algorithm*), którego ulepszoną wersję zaprezentował Sudhakar i Murthy w pracy [Sud97]. Polega on na wprowadzaniu „szumu” do zbioru danych (zbioru wag krawędzi) i rozwiązywaniu problemu lokalnego minimum. Proces ten jest powtarzany, przy czym przy kolejnych iteracjach ilość „szumu” jest zmniejszana.

Pirolli i Pitkow w pracy [Pit97] do grupowania stron WWW zaproponowali wykorzystanie algorytmu znanego z analizy cytowań (ang. *co-citation analysis*). Najpierw, z całej kolekcji, wydzielana jest lista dokumentów najczęściej cytowanych (do których prowadzi największa liczba odsyłaczy). Następnie, z tej listy tworzone są pary współcytowanych dokumentów (mają wspólnych rodziców), przy czym do dalszego przetwarzania przekazywane są jedynie te pary, które przekroczą zadany próg liczby wspólnych cytowań. Ostatnim krokiem jest utworzenie grup dokumentów, które posiadają cytowania do tych samych par z ww. listy.

4. PODOBIENSTWO DOKUMENTÓW HIPERTEKSTOWYCH

4.1. ZAŁOŻENIA DLA FUNKCJI PODOBIENSTWA

Konstruując funkcję podobieństwa dokumentów hipertekstowych przyjęto następujące założenia:

1. Podobieństwo między dwoma dokumentami wynika przede wszystkim ze struktury hipertekstowej. Uzyskuje się to poprzez uwzględnienie:
 - a) liczby odsyłaczy łączących oba dokumenty;
 - b) liczby wszystkich odsyłaczy, które wychodzą z obu dokumentów; pozwala to na ważenie odsyłaczy łączących dokumenty; waga jest większa wtedy, gdy są to jedyne odsyłacze występujące w tych dokumentach; mniejsza - gdy są one jednymi z wielu;
 - c) liczby i rodzaju bezpośrednich wspólnych potomków i przodków (w ramach kolekcji i poza nią);
 - d) rodzaju odsyłaczy według podziału na semantyczne i nawigacyjne przyjmując, że odsyłacze semantyczne lepiej niż nawigacyjne odzwierciedlają podobieństwo tematyczne między dokumentami.
2. Treść tekstowa wzmacnia podobieństwo obliczone na podstawie struktury. Nie może go jednak osłabić, tzn. jeżeli istnieje niezerowe podobieństwo wynikające ze struktury, to przy braku podobieństwa tekstów, wartość funkcji podobieństwa jest większa od zera¹. W przypadku zerowej wartości podobieństwa wynikającego ze struktury, należy uwzględnić treść tekstową.
3. Podobieństwo tekstów uwzględnia:
 - a) częstość występowania wspólnych terminów w obu dokumentach;
 - b) częstość występowania wspólnych terminów w całej kolekcji dokumentów;
 - c) liczbę dokumentów, w których wspólne terminy występują;
 - d) długość tekstów obu dokumentów;
 - e) miejsce występowania terminów (dotyczy to specjalnych części dokumentów, takich jak tytuł czy słowa kluczowe).
4. Zbiorem wartości funkcji podobieństwa jest przedział $[0,1]$.

4.2. PIERWOTNE PODOBIENSTWO STRUKTURALNE

Podobieństwo wynikające z powiązań hipertekstowych między dokumentami określa funkcja **pierwotnego podobieństwa strukturalnego**. Ponieważ nie istnieje żadna, uznana propozycja dla funkcji podobieństwa wykorzystującego strukturę hipertekstu, zaproponowano jej 4 różne postacie (wersje).

Oznaczenia występujące we wzorach na funkcje podobieństwa:

- a) $S^{ps}(d_i, d_j)$ – wartość pierwotnego podobieństwa strukturalnego między dokumentem d_i oraz d_j .

¹ Dzięki temu można uzyskać niezerowe podobieństwo między np. dwoma dokumentami w różnych wersjach językowych lub między dokumentami multimedialnymi.

- b) L_{ij}^s – zbiór odsyłaczy semantycznych (typ odsyłacza $t^L=sem$), które wychodzą z dokumentu d_i i prowadzą do dokumentu d_j :

$$L_{ij}^s = \{l_{ij}^{r,t} \in L, d_i, d_j \in D: i \neq j, t^L = sem\},$$

gdzie:

D – dziedzina, czyli kolekcja dokumentów (element kolekcji hipertekstowej $HT(D,L)$ – def. 2.2);

L – zbiór odsyłaczy kolekcji hipertekstowej $HT(D,L)$ – def. 2.2;

l_{ij}^{r,t^L} – r -ty odsyłacz prowadzący z dokumentu d_i do dokumentu d_j , typu t^L – def. 2.1.

- c) nl_{ij}^s – liczba odsyłaczy semantycznych z d_i do d_j ; liczba elementów zbioru L_{ij}^s .
- d) L_{ij}^n – zbiór odsyłaczy nawigacyjnych (typ odsyłacza $t^L=nav$), które wychodzą z dokumentu d_i i prowadzą do dokumentu d_j :

$$L_{ij}^n = \{l_{ij}^{r,t} \in L, d_i, d_j \in D: i \neq j, t^L = nav\},$$

- e) nl_{ij}^n – liczba odsyłaczy nawigacyjnych z d_i do d_j , liczba elementów w zbiorze L_{ij}^n
- f) nl'_{ij} – liczba wszystkich odsyłaczy, które wychodzą z dokumentu d_i i prowadzą do dokumentu d_j :

$$nl'_{ij} = nl_{ij}^n + nl_{ij}^s = card(L_{ij}^n) + card(L_{ij}^s).$$

- g) λ – współczynnik regulujący znaczenie odsyłaczy nawigacyjnych, $\lambda \in [0,1]$.
- h) nl_{ij} – względna liczba odsyłaczy z d_i do d_j :

$$nl_{ij} = nl_{ij}^s + \lambda nl_{ij}^n, \quad (4.1)$$

- i) L_i^s – zbiór wszystkich odsyłaczy semantycznych ($t^L=sem$) wychodzących z dokumentu d_i , a prowadzących do innych dokumentów w ramach danej kolekcji D :

$$L_i^s = \bigcup_{\substack{d_j \in D. \\ d_j \neq d_i}} L_{ij}^s,$$

- j) nl_i^s – liczba wszystkich odsyłaczy semantycznych wychodzących z dokumentu d_i i prowadzących do innych dokumentów w ramach danej kolekcji D . $nl_i^s = card(L_i^s)$.
- k) L_i^n – zbiór wszystkich odsyłaczy nawigacyjnych ($t^L=nav$) wychodzących z dokumentu d_i i prowadzących do innych dokumentów w ramach danej kolekcji D :

$$L_i^n = \bigcup_{\substack{d_j \in D. \\ d_j \neq d_i}} L_{ij}^n,$$

- l) nl_i^n – liczba wszystkich odsyłaczy nawigacyjnych, wychodzących z dokumentu d_i , prowadzących do innych dokumentów w ramach danej kolekcji D . $nl_i^n = card(L_i^n)$.
- m) nl_i – liczba wszystkich odsyłaczy wychodzących z dokumentu d_i , a prowadzących do innych dokumentów w ramach danej kolekcji D :

$$nl_i = nl_i^s + nl_i^n = card(L_i^s) + card(L_i^n). \quad (4.2)$$

Nie występuje tutaj współczynnik λ – jak w (4.1), tzn. odsyłacze semantyczne i nawigacyjne traktowane są z jednakową wagą.

- n) nl_{max} – maksymalna liczba odsyłaczy łączących dwa dokumenty, należące do kolekcji D :

$$nl_{max} = \max\{nl_{ij} + nl_{ji} : i \neq j, d_i, d_j \in D\}. \quad (4.3)$$

- o) $L_i^{\rightarrow}$ – zbiór odsyłaczy wychodzących z dokumentu d_i , należącego do kolekcji D i prowadzących do dowolnych innych dokumentów (w ramach kolekcji lub poza nią).

$$L_i^{\rightarrow} = \{l_{ij}^{r,s} \in L, d_i \in D, t^L \in T^L : (\exists d_j \in D^E \cup D^-) i \neq j\},$$

gdzie (def. 2.1):

D^E – zbiór dokumentów tekstowych należących do środowiska hipertekstowego,

D^- – zbiór dokumentów nieistniejących, spoza środowiska hipertekstowego, do którego prowadzą odsyłacze z dokumentów należących do tego środowiska.

- p) $nl_i^{\rightarrow}$ – liczba odsyłaczy wychodzących z dokumentu d_i , należącego do kolekcji D i prowadzących do dowolnych innych dokumentów (w ramach kolekcji lub poza nią).
 $nl_i^{\rightarrow} = \text{card}(L_i^{\rightarrow})$.

- q) $L_{ij}^{n\Rightarrow}$ – zbiór nawigacyjnych, pośrednich odsyłaczy cytowania z dokumentu d_i względem dokumentu d_j , tzn. takich pośrednich odsyłaczy cytowania (def. 2.3), dla których typ $t^L = \text{nav}$:

$$L_{ij}^{n\Rightarrow} = \{l_{ik}^{r,s} \in L, d_i \in D, d_k \in D^E \cup D^-, t^L = \text{nav} : (\exists d_j \in D) (\exists l_{jk}^{r',s'} \in L) i \neq j, i \neq k, j \neq k\}.$$

- r) $nl_{ij}^{n\Rightarrow}$ – liczba nawigacyjnych, pośrednich odsyłaczy cytowania z dokumentu d_i względem dokumentu d_j , czyli $nl_{ij}^{n\Rightarrow} = \text{card}(L_{ij}^{n\Rightarrow})$

- s) $L_{ij}^{s\Rightarrow}$ – zbiór semantycznych, pośrednich odsyłaczy cytowania z dokumentu d_i względem dokumentu d_j :

$$L_{ij}^{s\Rightarrow} = \{l_{ik}^{r,s} \in L, d_i \in D, d_k \in D^E \cup D^-, t^L = \text{sem} : (\exists d_j \in D) (\exists l_{jk}^{r',s'} \in L) i \neq j, i \neq k, j \neq k\}.$$

- t) $nl_{ij}^{s\Rightarrow}$ – liczba semantycznych, pośrednich odsyłaczy cytowania z dokumentu d_i , względem dokumentu d_j , czyli $nl_{ij}^{s\Rightarrow} = \text{card}(L_{ij}^{s\Rightarrow})$.

- u) $L_{ij}^{\Rightarrow}$ – zbiór wszystkich pośrednich odsyłaczy cytowania, wychodzących z dokumentu d_i , względem dokumentu d_j :

$$L_{ij}^{\Rightarrow} = \{l_{ik}^{r,s} \in L, d_i \in D, d_k \in D^E \cup D^- : (\exists d_j \in D) l_{jk}^{r',s'} \in L\}.$$

- v) $nl'_{ij}^{\Rightarrow}$ – liczba wszystkich, pośrednich odsyłaczy cytowania wychodzących z dokumentu d_i , względem dokumentu d_j , czyli $nl'_{ij}^{\Rightarrow} = nl_{ij}^{s\Rightarrow} + nl_{ij}^{n\Rightarrow} = \text{card}(L_{ij}^{\Rightarrow})$.

- w) $nl_{ij}^{\Rightarrow}$ – względna liczba wszystkich, pośrednich odsyłaczy cytowania wychodzących z dokumentu d_i , względem dokumentu d_j :

$$nl_{ij}^{\Rightarrow} = nl_{ij}^{s\Rightarrow} + \lambda nl_{ij}^{n\Rightarrow}. \quad (4.4)$$

- x) $L_{ij}^{n\Leftarrow}$ – zbiór nawigacyjnych, pośrednich odsyłaczy współcytowania dokumentu d_i względem dokumentu d_j , tzn. takich odsyłaczy współcytowania (def. 2.4), dla których typ $t^L = \text{nav}$:

$$L_{ij}^{n\Leftarrow} = \{l_{ki}^{r,t} \in L^E, d_i \in D, d_k \in D^E, t^L = nav : (\exists d_j \in D) (\exists l_{kj}^{r',t'} \in L^E) i \neq j, i \neq k, j \neq k\}.$$

y) $nl_{ij}^{n\Leftarrow}$ – liczba nawigacyjnych, pośrednich odsyłaczy współcytowania dokumentu d_i względem dokumentu d_j , czyli $nl_{ij}^{n\Leftarrow} = card(L_{ij}^{n\Leftarrow})$

z) $L_{ij}^{s\Leftarrow}$ – zbiór semantycznych odsyłaczy współcytowania dokumentu d_i względem dokumentu d_j :

$$L_{ij}^{s\Leftarrow} = \{l_{ki}^{r,t} \in L^E, d_i \in D, d_k \in D^E, t^L = sem : (\exists d_j \in D) (\exists l_{kj}^{r',t'} \in L^E) i \neq j, i \neq k, j \neq k\}.$$

aa) $nl_{ij}^{s\Leftarrow}$ – liczba semantycznych, pośrednich odsyłaczy współcytowania dokumentu d_i , względem dokumentu d_j , czyli $nl_{ij}^{s\Leftarrow} = card(L_{ij}^{s\Leftarrow})$.

bb) $L_{ij}^{\Leftarrow}$ – zbiór wszystkich pośrednich odsyłaczy współcytowania dokumentu d_i , względem dokumentu d_j :

$$L_{ij}^{\Leftarrow} = \{l_{ki}^{r,t} \in L^E, d_i \in D, d_k \in D^E : (\exists d_j \in D) l_{kj}^{r',t'} \in L^E\}.$$

cc) $nl_{ij}'^{\Leftarrow}$ – liczba wszystkich pośrednich odsyłaczy współcytowania dokumentu d_i , względem dokumentu d_j , czyli: $nl_{ij}'^{\Leftarrow} = nl_{ij}^{s\Leftarrow} + nl_{ij}^{n\Leftarrow} = card(L_{ij}^{\Leftarrow})$.

dd) $nl_{ij}^{\Leftarrow}$ – względna liczba wszystkich pośrednich odsyłaczy współcytowania dokumentu d_i , względem dokumentu d_j :

$$nl_{ij}^{\Leftarrow} = nl_{ij}^{s\Leftarrow} + \lambda nl_{ij}^{n\Leftarrow}. \quad (4.5)$$

ee) $L_{ij}^{\Rightarrow}$ – zbiór wszystkich odsyłaczy wychodzących z tych dokumentów, z których wychodzą także pośrednie odsyłacze współcytowania dokumentu d_i , względem dokumentu d_j :

$$L_{ij}^{\Rightarrow} = \{l_{kq}^{r,t} \in L^E, d_k \in D^E, d_q \in D^E : (\exists d_i \in D) l_{ki}^{r',t'} \in L^E \wedge (\exists d_j \in D) l_{kj}^{r',t'} \in L^E\}.$$

Oczywiście: $L_{ij}^{\Rightarrow} = L_{ji}^{\Rightarrow}$.

ff) $nl_{ij}^{\Rightarrow}$ – liczba wszystkich odsyłaczy wychodzących z tych dokumentów, z których wychodzą także pośrednie odsyłacze współcytowania dokumentu d_i względem dokumentu d_j . $nl_{ij}^{\Rightarrow} = card(L_{ij}^{\Rightarrow})$.

gg) μ – współczynnik regulujący znaczenie pośrednich odsyłaczy cytowania i współcytowania, $\mu \in [0, 1]$.

Widoczny w oznaczeniach podział odsyłaczy wynika z analizy przedstawionej w rozdziale drugim. W systemach hipertekstowych (w tym także w systemie WWW) istnieją dwa zasadnicze rodzaje odsyłaczy:

- Nawigacyjne (organizacyjne). Na stronach WWW umieszczane są najczęściej na początku lub na końcu i nie posiadają sąsiedztwa tekstowego
- Semantyczne (znaczeniowe). Na stronach WWW umieszczane są zwykle w środku strony albo w otoczeniu tekstu na początku lub końcu strony.

Współczynnik λ występuje przy liczbie odsyłaczy nawigacyjnych we wzorach na względne liczby odsyłaczy: nl_{ij} , $nl_{ij}^{\Rightarrow}$, $nl_{ij}^{\Leftarrow}$ (4.1), (4.4), (4.5). Dzięki λ możliwe jest regulowanie znaczenia odsyłaczy nawigacyjnych (w nl_{ij} , $nl_{ij}^{\Rightarrow}$, $nl_{ij}^{\Leftarrow}$ odsyłacze nawigacyjne i semantyczne są traktowane na równi). W przypadku $\lambda=1$ tracimy zupełnie rozróżnienie na odsyłacze nawigacyjne i semantyczne. Dla $\lambda=0$ odsyłacze nawigacyjne są pomijane. Wydaje się, że współczynnik λ powinien być

raczej bliższy jeden niż zero. Przemawia za tym spostrzeżenie, że odsyłacze nawigacyjne także niosą ze sobą informację o powiązaniach tematycznych między dokumentami (oczywiście waga takiej informacji jest mniejsza niż w przypadku odsyłaczy semantycznych). Wynika to z tego, że autor poprzez umieszczenie odsyłacza nawigacyjnego przewiduje, iż użytkownik zainteresowany danym dokumentem również będzie zainteresowany dokumentem, do którego prowadzi ten odsyłacz [Pit97]. Oba dokumenty są więc podobne z punktu widzenia potrzeb informacyjnych użytkownika.

Poniżej zostały przedstawione cztery wersje funkcji pierwotnego podobieństwa strukturalnego, oznaczone kolejnymi, wielkimi literami alfabetu, tzn. A, B, C oraz D¹. Wszystkie (prócz wersji D) bazują na odsyłaczach łączących dokumenty w ramach kolekcji (w ramach zbioru D), tzn. dokument początkowy i końcowy odsyłacza należy do zbioru D .

Chcąc zwiększyć przejrzystość zapisu, oznaczmy funkcję pierwotnego podobieństwa strukturalnego między dokumentami $d_i, d_j \in D$, obliczoną wg wersji A, jako $S_A^{ps}(d_i, d_j)$, wg wersji B – jako $S_B^{ps}(d_i, d_j)$, wg wersji C – $S_C^{ps}(d_i, d_j)$, a wg wersji D – $S_D^{ps}(d_i, d_j)$.

Wersja A.

Funkcja $S_A^{ps}(d_i, d_j)$ opiera się na znormalizowanej, średniej arytmetycznej liczby odsyłaczy prowadzących z jednego dokumentu do drugiego, dla których obliczamy wartość pierwotnego podobieństwa strukturalnego. Normalizacja dokonuje się przez podzielenie przez liczbę wszystkich odsyłaczy wychodzących z obu dokumentów do innych dokumentów w ramach kolekcji D . Funkcję $S_A^{ps}(d_i, d_j)$ można także traktować jako średnią arytmetyczną podobieństw między d_i i d_j oraz w drugą stronę: między d_j i d_i , dla funkcji Chena z pracy [Chen97] - wzór (3.5). Dzięki temu uzyskuje się symetrię funkcji:

$$S_A^{ps}(d_i, d_j) = \begin{cases} \frac{1}{2} \left(\frac{nl_{ij}}{nl_i} + \frac{nl_{ji}}{nl_j} \right), & nl_i > 0, nl_j > 0 \\ \frac{1}{2} \frac{nl_{ji}}{nl_j}, & nl_i = 0, nl_j > 0 \\ \frac{1}{2} \frac{nl_{ij}}{nl_i}, & nl_i > 0, nl_j = 0 \\ 0, & nl_i = 0, nl_j = 0 \end{cases}$$

$S_A^{ps}(d_i, d_j) \in [0, 1]$, ponieważ $nl_i \geq nl_{ij} \geq 0$, a $nl_j \geq nl_{ji} \geq 0$.

Wadą wersji A jest to, że w przypadku, gdy nie istnieją odsyłacze w drugą stronę (tzn. $nl_{ij}=0$ i $nl_{ji}>0$ lub $nl_{ij}>0$ i $nl_{ji}=0$), a sytuacja taka bardzo często się zdarza w przypadku stron WWW, wtedy jeden z ułamków w pierwszym wierszu ma wartość 0, zaś drugi jest dzielony przez 2. W efekcie, jeżeli istnieje odsyłacz między dokumentem d_i oraz d_j (a nawet wtedy, gdy jest ich dużo), lecz nie ma odsyłacza w drugą stronę, to i tak wartość pierwotnego podobieństwa strukturalnego nie może przekroczyć $\frac{1}{2}$.

Po uwzględnieniu współczynnika λ , wzór (4.1), otrzymujemy:

¹ Wstępne wersje funkcji podobieństwa opublikowano w pracy [Kaz99c].

$$S_A^{ps}(d_i, d_j) = \begin{cases} \frac{1}{2} \left(\frac{nl_{ij}^s + \lambda nl_{ij}^n}{nl_i} + \frac{nl_{ji}^s + \lambda nl_{ji}^n}{nl_j} \right), & nl_i > 0, nl_j > 0 \\ \frac{1}{2} \frac{nl_{ji}^s + \lambda nl_{ji}^n}{nl_j}, & nl_i = 0, nl_j > 0 \\ \frac{1}{2} \frac{nl_{ij}^s + \lambda nl_{ij}^n}{nl_i}, & nl_i > 0, nl_j = 0 \\ 0, & nl_i = 0, nl_j = 0 \end{cases}$$

Jeżeli $nl_{ij}^s = nl_{ji}^s = 0$, czyli gdy odsyłacze łączące d_i oraz d_j ze sobą są odsyłaczami nawigacyjnymi, wtedy $S_A^{ps}(d_i, d_j)$ ma wartość λ razy mniejszą niż wtedy, gdy wszystkie odsyłacze są semantyczne.

Wersja B.

Funkcja $S_B^{ps}(d_i, d_j)$ jest zmodyfikowaną postacią funkcji $S_A^{ps}(d_i, d_j)$:

$$S_B^{ps}(d_i, d_j) = \begin{cases} \frac{nl_{ij} + nl_{ji}}{nl_i + nl_j}, & \text{dla } nl_i + nl_j > 0 \\ 0, & \text{dla } nl_i + nl_j = 0 \end{cases}$$

Podobnie jak w przypadku $S_A^{ps}(d_i, d_j)$, zbiór wartości funkcji $S_B^{ps}(d_i, d_j)$ to przedział $[0, 1]$.

Dzięki wyeliminowaniu współczynnika $\frac{1}{2}$ przed ułamkiem uzyskuje się to, że pierwotne podobieństwo strukturalne może nawet osiągnąć wartość jeden, mimo tego, że nie będą istnieć odsyłacze w obie strony między dokumentem d_i oraz d_j .

Uwzględniając mniejsze znaczenie odsyłaczy nawigacyjnych w stosunku do semantycznych, czyli stosując wzór (4.1) do powyższego, otrzymujemy:

$$S_B^{ps}(d_i, d_j) = \begin{cases} \frac{(nl_{ij}^s + \lambda nl_{ij}^n) + (nl_{ji}^s + \lambda nl_{ji}^n)}{nl_i + nl_j}, & \text{dla } nl_i + nl_j > 0 \\ 0, & \text{dla } nl_i + nl_j = 0 \end{cases}$$

Wersja C.

W wersji A oraz B funkcji pierwotnego podobieństwa strukturalnego nie były uwzględnione gęstości odsyłaczy. Jeżeli między dokumentem d_i oraz d_j istnieje jeden odsyłacz semantyczny ($nl_{ij}=1$) i w drugą stronę także jest jeden odsyłacz semantyczny ($nl_{ji}=1$), oraz z każdego z tych dokumentów nie wychodzą inne odsyłacze ($nl_i=nl_j=1$), wtedy $S_A^{ps}(d_i, d_j) = S_B^{ps}(d_i, d_j) = 1$. Dokumenty są maksymalnie podobne do siebie. Tyle samo wyniosą wartości pierwotnego podobieństwa strukturalnego dla obu wersji A i B (tzn. 1) w sytuacji, gdy między dokumentami jest po pięć odsyłaczy ($nl_{ij}=nl_{ji}=5$) i ponownie nie wychodzą z nich inne odsyłacze ($nl_i=nl_j=5$).

Gęstość odsyłaczy uwzględniono w definicji funkcji $S_C^{ps}(d_i, d_j)$:

$$S_C^{ps}(d_i, d_j) = \begin{cases} \frac{nl_{ij} + nl_{ji}}{nl_{max}}, & nl_{max} > 0 \\ 0, & nl_{max} = 0 \end{cases}$$

Normalizacja (mianownik ułamka) następuje tutaj nie względem liczby odsyłaczy wychodzących z dokumentów – między którymi liczymy podobieństwo – lecz względem maksymalnej liczby odsyłaczy łączących jakiegokolwiek dwa dokumenty w kolekcji, czyli nl_{max} .

Zbiór wartości funkcji $S_C^{ps}(d_i, d_j)$ to przedział $[0, 1]$, ponieważ $nl_{max} \geq nl_{ij} + nl_{ji}$ (wzór 4.3).

Jeżeli $nl_{max} = 0$, to w kolekcji D nie występują odsyłacze między dokumentami i kolekcja ta w ogóle nie posiada charakteru hipertekstowego.

Stosując wzór (4.1) do powyższego otrzymujemy:

$$S_C^{ps}(d_i, d_j) = \begin{cases} \frac{(nl_{ij}^s + \lambda nl_{ij}^n) + (nl_{ji}^s + \lambda nl_{ji}^n)}{nl_{max}}, & nl_{max} > 0 \\ 0, & nl_{max} = 0 \end{cases}$$

Wersja D.

Poprzednie wersje postaci funkcji pierwotnego podobieństwa strukturalnego, tj. A, B oraz C uwzględniały jedynie (w liczniku) odsyłacze łączące ze sobą bezpośrednio dokument d_i oraz d_j . Jednakże – już wiele lat temu – przy opracowywaniu różnych funkcji podobieństwa, związanych ze wzajemnym cytowaniem w tekstach naukowych, dostrzeżono, że dokumenty, które cytują te same, inne dokumenty nawet, jeżeli nie cytują siebie nawzajem, są do siebie podobne. Można założyć, że podobna hipoteza jest słuszna dla środowiska hipertekstowego. Rozróżnijmy dwa przypadki:

1. Ponieważ dwa dokumenty d_i oraz d_j wskazują na ten sam inny dokument d_k , więc d_i oraz d_j są do siebie podobne (wspólni potomkowie - dzieci).
2. Dokument d_k wskazuje zarówno na dokument d_i jak i d_j , więc dokumenty d_i oraz d_j są do siebie podobne (wspólni przodkowie - rodzice); dokumenty są współcytowane.

Pierwszy przypadek wynika z istnienia pośrednich odsyłaczy cytowania (def. 2.3) – zbiór $L_{ij}^{\Rightarrow}$ oraz $L_{ji}^{\Rightarrow}$. Drugi natomiast to pośrednie odsyłacze współcytowania (def. 2.4) – zbiory $L_{ij}^{\Leftarrow}$ i $L_{ji}^{\Leftarrow}$.

Na rys. 4.1 przedstawiona jest kolekcja dokumentów tekstowych (zbiór D) wraz z odsyłaczami (zbiór L), czyli kolekcja hipertekstowa $HT(D, L)$. Przykładem pierwszego przypadku na tym rysunku jest dokument 1 (d_i) oraz 6 (d_j), które wskazują na ten sam dokument 2 (d_k). Może zaistnieć również taka sytuacja, w której dokument d_k nie należy do kolekcji, czyli $d_k \in (D^E \cup D^-) \setminus D$. Na przykład, odsyłacze z dokumentu 10 oraz 11 prowadzą do dokumentu 22, leżącego poza zbiorem D grupowanej kolekcji. Drugi przypadek to 8 (d_k), z którego prowadzą odsyłacze zarówno do dokumentu 4 (d_i), jak i 9 (d_j). Tutaj także może zaistnieć sytuacja, w której dokument wskazujący na te, między którymi obliczane jest podobieństwo, czyli d_k , leży poza zbiorem D , jednak nie może to być element zbioru D^- . Przykładem tego jest dokument 23, nienależący do kolekcji D , posiadający odsyłacze do dokumentu 6 oraz 10.

Intuicyjnie można domniemywać, że podobieństwo wynikające tylko z istnienia pośrednich odsyłaczy cytowania i współcytowania powinno mieć mniejsze znaczenie (wagę) niż podobieństwo, które jest związane z odsyłaczami bezpośrednio łączącymi dwa dokumenty (cytowanie siebie nawzajem). W związku z tym proponuje się wprowadzenie współczynnika ważności pośrednich odsyłaczy cytowania i współcytowania μ , zmniejszającego ich wagę (znaczenie) w stosunku do odsyłaczy bezpośrednio łączących dwa dokumenty.

Przy opracowywaniu funkcji pierwotnego podobieństwa strukturalnego $S_D^{ps}(d_i, d_j)$ oparto się na wersji B. Jako element normalizacji liczby odsyłaczy pośrednich cytowania i bezpośrednich łącznie przyjęto liczbę wszystkich odsyłaczy wychodzących z dokumentów d_i oraz d_j , czyli odpowiednio $nl_i^{\rightarrow}$ i $nl_j^{\rightarrow}$. Różnica między nl_i i nl_j , które występują w wersji B, a $nl_i^{\rightarrow}$ i $nl_j^{\rightarrow}$ polega na tym, że te ostatnie uwzględniają także odsyłacze prowadzące do dokumentów spoza kolekcji (zbiór dokumentów końcowych tych odsyłaczy to $D^E \cup D^-$). Do normalizacji pośrednich odsyłaczy

współcytowania wykorzystano liczbę wszystkich odsyłaczy wychodzących ze wspólnych dla d_i i d_j przodków-rodziców (zbiór $L_{ij}^{\leftrightarrow}$ o liczności $nl_{ij}^{\leftrightarrow}$). W efekcie powstał wzór:

$$S'^{ps}(d_i, d_j) = \begin{cases} \frac{nl_{ij} + nl_{ji} + \mu(nl_{ij}^{\rightarrow} + nl_{ji}^{\rightarrow} + nl_{ij}^{\leftarrow} + nl_{ji}^{\leftarrow})}{nl_i^{\rightarrow} + nl_j^{\rightarrow} + nl_{ij}^{\leftrightarrow}}, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} + nl_{ij}^{\leftrightarrow} > 0 \\ 0, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} + nl_{ij}^{\leftrightarrow} = 0 \end{cases} \quad (4.6)$$

Stosując do powyższego wzory (4.1), (4.4) i (4.5), regulujące znaczenie odsyłaczy nawigacyjnych, otrzymujemy:

$$S'^{ps}(d_i, d_j) = \begin{cases} \frac{nl_{ij}^s + nl_{ji}^s + \lambda(nl_{ij}^n + nl_{ji}^n) + \mu(nl_{ij}^{s\rightarrow} + nl_{ji}^{s\rightarrow} + nl_{ij}^{s\leftarrow} + nl_{ji}^{s\leftarrow}) + \lambda(nl_{ij}^{n\rightarrow} + nl_{ji}^{n\rightarrow} + nl_{ij}^{n\leftarrow} + nl_{ji}^{n\leftarrow})}{nl_i^{\rightarrow} + nl_j^{\rightarrow} + nl_{ij}^{\leftrightarrow}}, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} + nl_{ij}^{\leftrightarrow} > 0 \\ 0, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} + nl_{ij}^{\leftrightarrow} = 0 \end{cases}$$

W większości środowisk hipertekstowych, w tym także w systemie WWW, pełna informacja o odsyłaczach jest umieszczana w treści dokumentu. Mając dostęp do pełnej treści posiada się więc także możliwość wykorzystania informacji semantycznej, którą niosą ze sobą odsyłacze. W związku z tym ustalenie wspólnych potomków-dzieci, poprzez porównanie adresów węzłów docelowych odsyłaczy nie nastręcza większych problemów¹. Aby jednak wyznaczyć wspólnych przodków-rodziców i obliczyć liczby odpowiednich odsyłaczy z nich wychodzących ($nl_{ij}^{\leftarrow}$, $nl_{ji}^{\leftarrow}$, $nl_{ij}^{\rightarrow}$), należy mieć dostęp do treści owych przodków. Jest to możliwe wtedy, gdy grupowaną kolekcją jest pewien zamknięty system hipertekstowy, np. system autorski lub pojedynczy serwis informacyjny WWW.

Poważny problem pojawia się jednak wtedy, gdy kolekcją hipertekstową $HT(D, L)$ będzie zbiór stron zwróconych przez wyszukiwarki internetowe. Jak uzyskać w takim przypadku informację o treści przodków? Wymaga to przecież ustalenia, które dokumenty z całego środowiska hipertekstowego (całego systemu WWW) wskazują na dane dwie strony, należące do kolekcji. Należy więc dotrzeć do stron (dokumenty d_k), które w swojej treści zawierają odsyłacze do stron, dla których obliczane jest podobieństwo (adresy URL strony d_i oraz d_j). W tym celu należy zadać wyszukiwarkom pytanie q_1 składające się iloczynu logicznego adresów URL strony d_i ($adresURL_i$) oraz d_j ($adresURL_j$), traktowanych jako słowa kluczowe²:

$$q_1 = adresURL_i \wedge adresURL_j.$$

Już na tym etapie pojawia się trudność związana z różną postacią adresów URL. Na stronie mogą występować adresy względne, które dopiero w połączeniu z adresem serwera dają pełny adres URL. W adresie może występować (albo nie) nazwa pliku na serwerze WWW. W tym drugim przypadku serwer przyjmuje pewne wartości domyślne³. Istnieją także alternatywne nazwy zare-

¹ W przypadku WWW pewną trudność sprawiają różne postacie adresów URL, np. odesłania względne, które wymagają połączenia ich z adresem serwera.

² Odpowiedzi wyszukiwarek będą niestety przybliżone, gdyż nie obejmują one całego środowiska hipertekstowego. Wg badań Bharata i Brodera [Bha98a] pojedyncza wyszukiwarka nie pokrywa więcej niż 50% całego systemu WWW. Zgodnie z badaniami przeprowadzonymi w 1999r. przez Lawrence'a oraz Gilesa pokrycie pojedynczej wyszukiwarki jest mniejsze niż 40%, zaś więcej niż 16% ogólnie dostępnych stron nie jest zaindeksowanych przez żadną wyszukiwarkę [Law99b, Sul00].

³ zwykle jest to index.html lub default.html, ale nie zawsze

jestrowane w serwerach nazw domen (DNS) – różne dla tego samego serwera WWW¹. W konsekwencji należałoby zadać nie pytanie q_1 a raczej zbiór pytań zawierający wszystkie możliwe kombinacje postaci adresu $adresURL_i$ z wszystkimi możliwymi postaciami adresu $adresURL_j$.

Po uzyskaniu odpowiedzi z wyszukiwarki należy pobrać z sieci Internet treść stron zwróconych jako odpowiedź wyszukiwarek, aby potwierdzić, czy rzeczywiście posiadają one odsyłacze do dwóch interesujących nas stron.

Jeżeli niezbędnym jest uzyskanie macierzy podobieństw, to wtedy koniecznym, jest zadanie po jednym pytaniu skierowanym do wyszukiwarek, dla każdej pary dokumentów z kolekcji (problem różnych postaci adresów URL został tutaj pominięty). Daje to łącznie $\binom{N}{2} = \frac{N(N-1)}{2}$ pytań do

wyszukiwarek, gdzie: N to liczba dokumentów w kolekcji D . Zachodzi tutaj także konieczność uzyskania treści wielu stron, zwróconych jako odpowiedzi na te pytania. Można to optymalizować, tj. zadawać pytanie o każdą stronę ($q_2 = adresURL_i$) czyli zadać tylko N pytań i wśród treści stron, których adresy są podane w odpowiedzi, szukać tych, które zawierają adres $adresURL_j$. Nie jest jednak pewne, czy będzie to rozwiązanie szybsze niż zadanie pierwszego pytania q_1 , gdyż prawdopodobnie konieczne będzie pobranie treści większej liczby stron.

W związku z powyższym postanowiono zredukować wzór (4.6) poprzez usunięcie z niego liczb tych odsyłaczy, które są związane ze wspólnymi przodkami – współcytowaniem. W wyniku tego, opracowano wersję D funkcji pierwotnego podobieństwa strukturalnego:

$$S_D^{ps}(d_i, d_j) = \begin{cases} \frac{nl_{ij} + nl_{ji} + \mu(nl_{ij}^{\rightleftharpoons} + nl_{ji}^{\rightleftharpoons})}{nl_i^{\rightarrow} + nl_j^{\rightarrow}}, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} > 0 \\ 0, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} = 0 \end{cases} \quad (4.7)$$

Podobnie jak w przypadku pozostałych wersji, zbiór wartości funkcji $S_D^{ps}(d_i, d_j)$ to przedział $[0, 1]$.

Po uwzględnieniu (4.1) oraz (4.4), otrzymujemy:

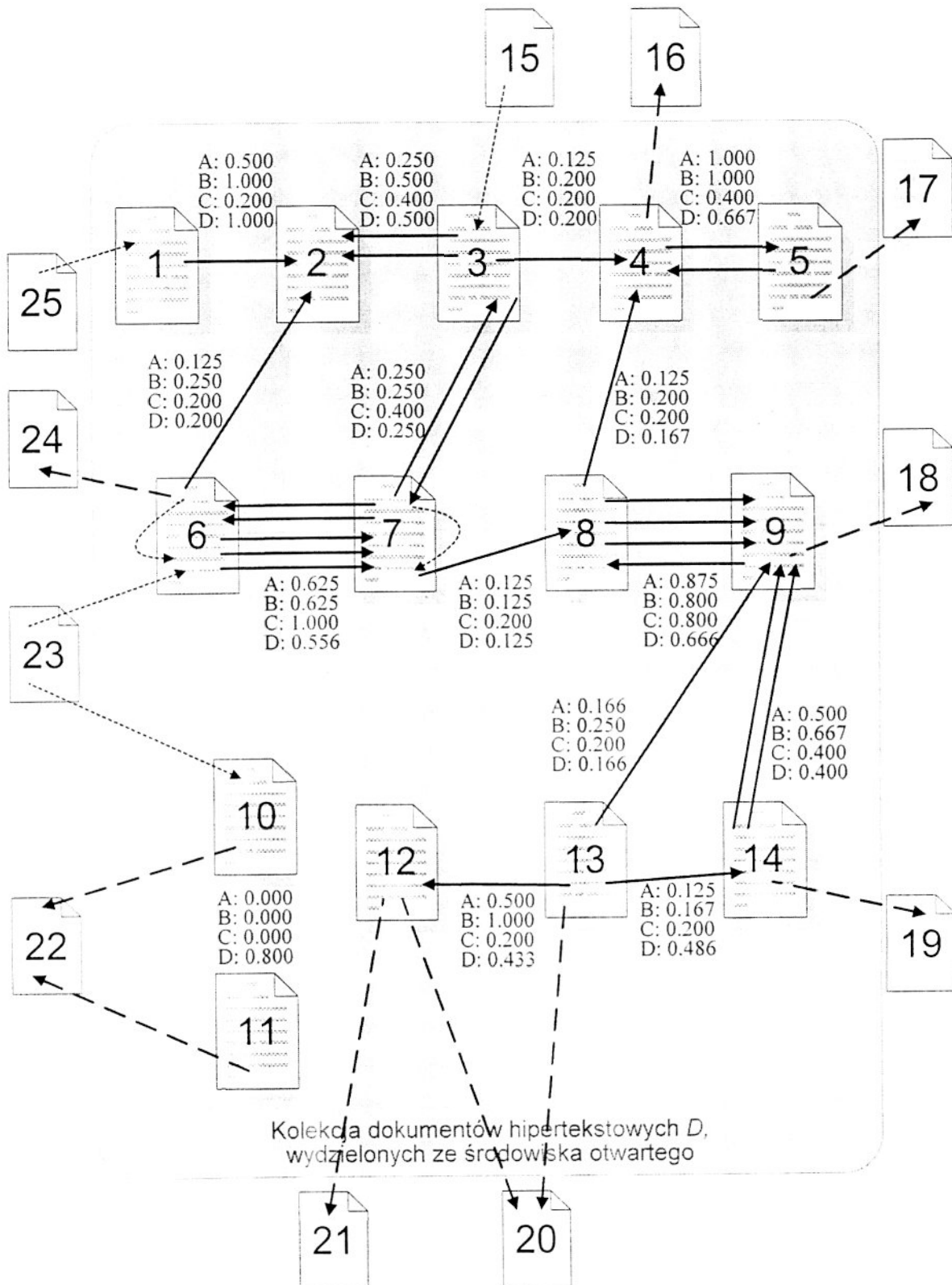
$$S_D^{ps}(d_i, d_j) = \begin{cases} \frac{(nl_{ij}^s + \lambda nl_{ij}^n) + (nl_{ji}^s + \lambda nl_{ji}^n) + \mu(nl_{ij}^{s\rightleftharpoons} + \lambda nl_{ij}^{n\rightleftharpoons} + nl_{ji}^{s\rightleftharpoons} + \lambda nl_{ji}^{n\rightleftharpoons})}{nl_i^{\rightarrow} + nl_j^{\rightarrow}}, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} > 0 \\ 0, & \text{dla } nl_i^{\rightarrow} + nl_j^{\rightarrow} = 0 \end{cases}$$

W dalszej części pracy rozważana będzie jedynie wersja D w postaci wzoru 4.7, tzn. bez uwzględniania wspólnych przodków-rodziców.

4.3. WŁASNOŚCI PIERWOTNEGO PODOBIENSTWA STRUKTURALNEGO

Na rys. 4.1 przedstawiono kolekcję dokumentów hipertekstowych, dla których obliczono parami wartości funkcji podobieństwa dla różnych wersji tej funkcji (A, B, C oraz D). Założono, że wszystkie odsyłacze mają charakter semantyczny, czyli $nl_{ij}^n = nl_{ji}^n = 0$, co oznacza, że $nl_{ij} = nl_{ij}^s$, zaś $nl_{ji} = nl_{ji}^s$. Przy ustalaniu wartości podobieństwa dla wersji C konieczne było określenie wartości nl_{max} , która dla tej kolekcji wynosi 5 (liczba odsyłaczy między dokumentami 6 i 7). Dla wersji D, obliczeń dokonano przy współczynniku $\mu = 0.8$.

¹ Na przykład, wszystkie adresy: <http://www.gsmsend.com/index.html>, <http://www.gsmsend.com/>, <http://gsmsend.com>, <http://www.gs.com>, <http://www.gsmsend.com/index.htm>, <http://www.oldgsmsend.com/index.html> wskazują na ten sam dokument w sieci WWW.



Rys. 4.1. Kolekcja D dokumentów hipertekstowych z obliczonymi wartościami funkcji pierwotnego podobieństwa strukturalnego.

Odsyłacze bezpośrednie (łącznie dwa dokumenty, należące do kolekcji D) narysowane są linią ciągłą. Odsyłacze oznaczone linią przerywaną to te, które biorą udział jedynie w obliczaniu podobieństwa $S_D^{ps}(d_i, d_j)$ i prowadzą do dokumentów nie należących do kolekcji. Są to odsyłacze: (4,16), (5,17), (6,24), (9,18), (10,22), (11,22), (12,21), (12,20), (13,20), (14,19).

Odsyłacze, które w ogóle nie były uwzględniane przy obliczaniu wartości funkcji pierwotnego podobieństwa strukturalnego (dla żadnej wersji) zostały oznaczone linią kropkowaną. Dotyczy to

odsyłaczy prowadzących do innych części tego samego dokumentu (z d_i do d_i) oraz prowadzące z dokumentów nie należących do kolekcji D do dokumentów do kolekcji należących. Są to odsyłacze: (6,6), (7,7), (15,3), (25,1), (23,6), (23,10).

Dla funkcji podobieństwa obliczanej wg wersji D, w kolekcji z rys. 4.1, istnieje $S_D^{ps}(d_i, d_j) > 0$, dla niektórych takich par (d_i, d_j) , przy których podobieństwo liczone według pozostałych wersji jest równe 0. Jest to związane z tym, że nie ma odsyłaczy bezpośrednio łączących dokumenty w takiej parze ($nl_{ij}=nl_{ji}=0$). Między dokumentem 10 oraz 11 wartość pierwotnego podobieństwa strukturalnego $S_D^{ps}(10,11)=0.8$, zaś $S_A^{ps}(10,11)=S_B^{ps}(10,11)=S_C^{ps}(10,11)=0$. Istnieje więcej podobnych, niezerowych pierwotnych podobieństw strukturalnych liczonych wg wersji D, niezaznaczonych na rysunku: $S_D^{ps}(1,3)=0.267$, $S_D^{ps}(1,6)=0.267$; $S_D^{ps}(3,5)=0.267$, $S_D^{ps}(3,6)=0.533$; $S_D^{ps}(3,8)=0.2$; $S_D^{ps}(5,8)=0.267$; $S_D^{ps}(7,9)=0.32$; $S_D^{ps}(8,13)=0.4$; $S_D^{ps}(8,14)=0.571$. Dla dokumentów 3 oraz 8 pośrednie odsyłacze cytowania prowadzą nawet do dwóch, różnych, innych dokumentów 4 oraz 7.

Jeżeli dla dwóch dokumentów d_i i d_j nie istnieją bezpośrednio¹ ($nl_{ij}=nl_{ji}=0$) ani pośrednio² ($nl_{ij}^{\Rightarrow}=nl_{ji}^{\Rightarrow}=nl_{ij}^{\Leftarrow}=nl_{ji}^{\Leftarrow}=0$) odsyłacze je łączące, wtedy $S_A^{ps}(d_i, d_j)=S_B^{ps}(d_i, d_j)=S_C^{ps}(d_i, d_j)=0$.

Jeżeli obliczane jest podobieństwo danego dokumentu d_i do niego samego, wtedy otrzymujemy:

$$(\forall d_i \in D) S^{ps}(d_i, d_i) = 0. \quad (4.8)$$

Wynika to z tego, że odsyłacze odnoszące się do innych części tego samego dokumentu są pomijane w procesie obliczania pierwotnego podobieństwa strukturalnego, dla wszystkich jego wersji. W związku z tym liczniki ułamków mają wartość 0. Niespełniona jest więc nierówność (3.1).

Pierwotne podobieństwo strukturalne spełnia warunek symetrii (3.2), dla wszystkich wersji, oraz (także dla wszystkich wersji) nie spełnia warunku trójkąta (3.3).

Pierwotne podobieństwo strukturalne między dwoma dokumentami d_i oraz d_j uwzględnia jedynie odsyłacze łączące te dokumenty i ewentualnie odesłania do wspólnych potomków - dzieci. W przypadku, gdy między dwoma dokumentami nie istnieją odsyłacze, zaś inne odsyłacze wychodzące z tych stron nie prowadzą do wspólnych potomków, pierwotne podobieństwo strukturalne jest równe zero. Jest to odmienne podejście niż to, zaprezentowane przez Weissa i innych [Wei96], w którym uwzględniane są ścieżki łączące dokumenty. Spełnia ono jednak warunek 1e założeń.

W przeciwieństwie do propozycji podobieństwa strukturalnego zawartych w pkt. 3.1, pierwotne podobieństwo strukturalne uwzględnia gęstość odsyłaczy w dokumentach, dla których dokonuje się obliczeń.

Pierwotne podobieństwo strukturalne spełnia więc punkt 1 założeń. Wersje A, B, C spełniają podpunkty a), b), d), zaś wersja D dodatkowo podpunkt d).

Zbiorem wartości pierwotnego podobieństwa strukturalnego jest przedział $[0,1]$.

4.4. ZMODYFIKOWANE PODOBIENSTWO STRUKTURALNE

Pierwotne podobieństwo strukturalne odzwierciedla strukturę hipertekstu poprzez uwzględnienie liczby odsyłaczy łączących dokumenty. Podobieństwo to jest niezbędne dla nadania wag krawędziom grafu ważonego³. Aby powstały graf nie był grafem pełnym (w którym istnieją krawędzie między każdą parą węzłów), zastosowano próg τ istotności podobieństwa dokumentów,

¹ tj. dla wersji A, B oraz C

² dla wersji D

³ proces ważenia krawędzi grafu został opisany szczegółowo w dalszej części pracy (pkt. 5.2)

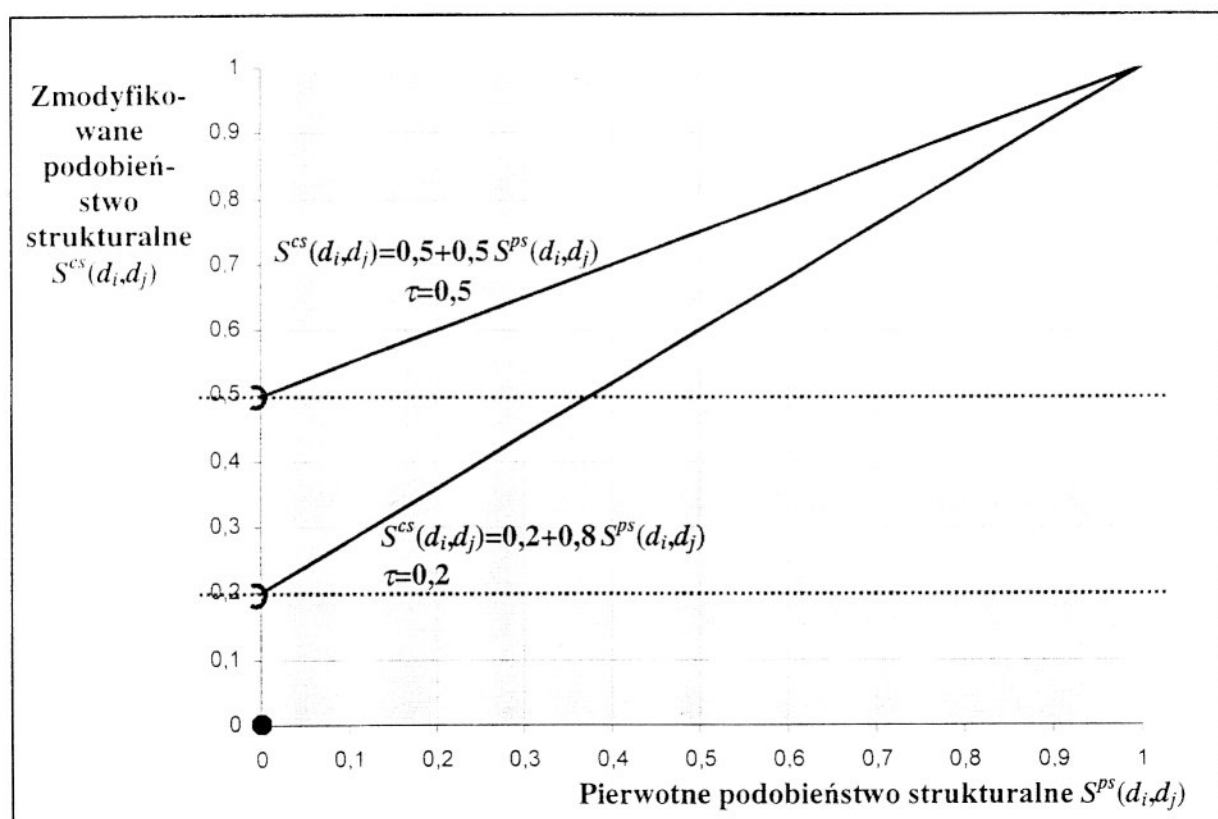
$\tau \in [0,1]$. Krawędź w grafie ważonym będzie istnieć tylko wtedy, gdy podobieństwo między dokumentami odpowiadającymi węzłom grafu połączonym przez tę krawędź przekroczy próg τ .

Jednym z założeń metody grupowania na podstawie drzewa maksymalnych przepływów jest to, że funkcja podobieństwa powinna uwzględniać przede wszystkim strukturę hipertekstową (założenie 1, pkt. 4.1). Ponieważ jednak wartości pierwotnego podobieństwa strukturalnego $S^{ps}(d_i, d_j) \in [0,1]$, więc może się zdarzyć, że $S^{ps}(d_i, d_j) \leq \tau$. W przypadku dokumentów - spisów treści lub indeksów wartość $S^{ps}(d_i, d_j)$ jest bliska zero. Może więc nawet wystąpić taka sytuacja, że będą istniały odsyłacze między dokumentami, a nie będzie krawędzi między nimi w grafie ważonym. Aby wyeliminować takie przypadki, przekształcono funkcję pierwotnego podobieństwa strukturalnego, przenosząc zbiór jej wartości z przedziału $[0,1]$ do $\{0\} \cup (\tau, 1]$, za pomocą funkcji liniowej.

$$S^{cs}(d_i, d_j) = \begin{cases} \tau + S^{ps}(d_i, d_j)(1 - \tau), & \text{dla } S^{ps}(d_i, d_j) > 0 \\ 0, & \text{dla } S^{ps}(d_i, d_j) = 0 \end{cases}$$

Wartości $S^{ps}(d_i, d_j)$ z przedziału $(0,1]$ są przenoszone do przedziału $(\tau, 1]$, zaś dla $S^{ps}(d_i, d_j) = 0$, nie następuje przeniesienie. Tak powstała funkcję nazywamy **zmodyfikowanym podobieństwem strukturalnym** (ang. *converted structural similarity*) i oznaczamy przez $S^{cs}(d_i, d_j)$. Modyfikacja podobieństwa strukturalnego wynika więc ściśle ze specyfiki metody grupowania na podstawie drzewa maksymalnych przepływów.

Wykres zależności $S^{cs}(d_i, d_j)$ od $S^{ps}(d_i, d_j)$, dla dwóch różnych wartości współczynnika progowego $\tau = 0.2$ oraz $\tau = 0.5$, przedstawiono na rys. 4.2. Do obu wykresów należy środek układu współrzędnych.



Rys. 4.2. Wykresy zależności zmodyfikowanego podobieństwa strukturalnego $S^{cs}(d_i, d_j)$ od pierwotnego podobieństwa strukturalnego $S^{ps}(d_i, d_j)$, dla dwóch różnych wartości współczynnika τ .

Funkcja zmodyfikowanego podobieństwa strukturalnego ma następujące właściwości:

- Jeżeli obliczamy podobieństwo danego dokumentu do niego samego, wtedy, na podstawie (4.8), otrzymujemy:

$$(\forall d_i \in D) S^{cs}(d_i, d_i) = 0, \quad (4.9)$$

czyli nie jest spełniony warunek (3.1).

- b) Funkcja jest symetryczna, tzn. spełnia warunek (3.2), co wynika z symetryczności funkcji pierwotnego podobieństwa strukturalnego.
- c) Funkcja nie spełnia – podobnie jak funkcja pierwotnego podobieństwa strukturalnego warunku trójkąta (3.3).

4.5. PODOBIENSTWO TREŚCIOWE

Drugim elementem podobieństwa dokumentów hipertekstowych, prócz podobieństwa struktury wyrażonego przez zmodyfikowane podobieństwo strukturalne, jest **podobieństwo treści**, nazwane także **podobieństwem treściowym** (ang. *content similarity*) lub **podobieństwem tekstowym**. W niniejszej pracy proponuje się wykorzystanie, ogólnie znanego, podejścia do podobieństwa treści, poprzez zastosowanie modelu wektorowego, w którym każdy dokument składa się z wielowymiarowego wektora wag poszczególnych terminów. Wagi te związane są z częstością występowania terminów w dokumentach. Podobieństwo treściowe, które oznaczamy przez $S^c(d_i, d_j)$, na podstawie pracy Saltona [Sal88], można wyrazić poprzez:

$$S^c(d_i, d_j) = \frac{\sum_{k=1}^{nt} w'_{d_i,k} w'_{d_j,k}}{\sqrt{\left(\sum_{k=1}^{nt} w'^2_{d_i,k} \right) \left(\sum_{k=1}^{nt} w'^2_{d_j,k} \right)}}. \quad (4.10)$$

gdzie: $w'_{d_i,k}$ – waga terminu k w dokumencie d_i ,

nt – liczba wszystkich terminów występujących w dokumentach d_i oraz d_j łącznie. Można także wartość nt traktować jako liczbę wszystkich terminów z wszystkich dokumentów z całej kolekcji, gdyż terminy nie występujące w którymś z dokumentów d_i lub d_j , i tak mają odpowiednią wagę równą zero.

Jest to typowa formuła podobieństwa cosinusowego (3.6), używana w modelu wektorowym. Model ten został opisany przez Saltona m.in. w pracach [Sal75a, Sal75b, Sal83].

Wartości tak określonego podobieństwa zawierają się w przedziale $[0,1]$. Normalizacja cosinusowa (mianownik) jest najczęściej używaną normalizacją w modelu wektorowym [Sin96].

Wagę terminu k w dokumencie d_j , czyli $w'_{d_j,k}$ można wyrazić wzorem:

$$w'_{d_j,k} = \frac{tf_{d_j,k} \log \frac{N}{nd_k}}{\sqrt{\sum_{i=1}^{nt_j} \left(tf_{d_j,i} \log \frac{N}{nd_i} \right)^2}}, \quad (4.11)$$

gdzie: $tf_{d_j,k}$, $tf_{d_j,i}$ – liczba wystąpień (częstość) terminu odpowiednio, k lub i w dokumencie d_j ,

N – liczba wszystkich dokumentów w kolekcji, czyli $N = \text{card}(D)$,

nd_k , nd_i – liczba dokumentów z kolekcji, w których występuje termin odpowiednio k lub i ,

nt_j – liczba różnych terminów zawartych w dokumencie d_j .

Podobieństwo treściowe wyznaczone przez (4.10) oraz (4.11) uwzględnia:

- a) Długość dokumentów, gdyż we wzorze (4.11) wartość mianownika zależy od długości dokumentu d_j . Im dokument dłuższy (im zawiera więcej różnych terminów, duże nt_j), tym większy jest mianownik. Oznacza to, że w dokumentach krótkich, zawierających mało terminów, wagi tych terminów są duże.
- b) Częstość występowania danego terminu w danym dokumencie poprzez wartość $tf_{d,k}$ w liczniku wzoru (4.11). Im częściej dany termin występuje w danym dokumencie, tym jest ważniejszy.
- c) Częstość występowania poszczególnych terminów w całej kolekcji, tzn. terminy, które występują w dużej liczbie dokumentów¹, będą miały mniejszą wagę w każdym z nich, gdyż $\log \frac{N}{nd_k}$, dla tych terminów, będzie miał małą wartość.
- d) Częstość występowania danego terminu w obu dokumentach, dla których liczone jest podobieństwo. Dokonuje się to w liczniku wzoru (4.10). Jeżeli wagi danego terminu w obu dokumentach są duże, wtedy licznik wzoru (4.10) będzie duży, czyli dany termin silniej, niż inne, będzie wpływał na podobieństwo obu dokumentów. Wagi te mogły być duże tylko pod warunkiem, iż nie występowały one w zbyt wielu innych dokumentach, tzn. element $\log \frac{N}{nd_k}$ we wzorze (4.11) musiał być wystarczająco duży.

Na podstawie tego można stwierdzić, że podobieństwo treściowe opisane przez (4.10) oraz (4.11) spełnia warunki 3a)-d) założeń (pkt. 4.1).

Wagi terminu w dokumencie, czyli $w'_{d,k}$, należą do przedziału $[0,1]$, gdyż mianownik we wzorze (4.11) jest nie mniejszy od licznika (zawiera licznik). W związku z tym, także $S^c(d_i, d_i) \in [0,1]$.

Zastosowanie (4.10) oraz (4.11), powoduje, że nie zachodzi konieczność stosowania tzw. stop listy, czyli zbioru terminów, które są pomijane. Terminy ze stop listy są pomocnicze i bardzo popularne w danym języku, zaś ich wpływ na podobieństwo treściowe nie jest znaczący.

Jeżeli dokumentami, między którymi obliczamy podobieństwo treściowe, będzie ten sam jeden dokument, wtedy wartość licznika we wzorze (4.10) będzie równa jego mianownikowi, czyli:

$$(\forall d_i, d_j \in D) S^c(d_i, d_i) = 1 \geq S^c(d_i, d_j). \quad (4.12)$$

W związku z tym spełniona jest nierówność (3.1).

Funkcja podobieństwa treściowego jest także symetryczna, tzn. spełniona jest równość (3.2).

Funkcja podobieństwa treściowego jest także – w przeciwieństwie do podobieństwa strukturalnego – półmetryczną, tzn. spełnia dodatkowo warunek trójkąta (3.3).

4.6. PRZEKSZTAŁCENIA PODOBIENSTWA TREŚCIOWEGO UWZGLĘDNIAJĄCE SPECYFIKĘ STRON WWW

Strony systemu WWW posiadają pewną specyfikę w porównaniu do innych dokumentów hipertekstowych. Językiem opisującym ich zawartość jest HTML, w którym wydzielono dwie główne części dokumentu:

- a) nagłówek (ang. *header*),
- b) ciało (ang. *body*).

¹ Dla języka polskiego będą to np. słowa: „i”, „lub”, „że”, itd., zaś dla języka angielskiego „the”, „and”, „of”, itd.

W skład nagłówka (część HEAD) wchodzi (lub raczej może wchodzić) kilka elementów bardzo istotnych dla określania zawartości treściowej stron WWW. Trzeba tutaj wymienić przede wszystkim (za [HTM98b]):

- a) **Tytuł** – element nagłówka o nazwie TITLE. Zawartość tego elementu wyświetla się zwykle w górnym pasku tytułowym okna przeglądarki. Element ten jest standardowym dla języka HTML w wersji 4.0. Jego składnia wygląda następująco:

`<TITLE>Tytuł dokumentu</TITLE>.`

- b) **Opis** dokumentu HTML (ang. *description*), przewidziany na słowne, krótkie określenie zawartości całej strony (całego dokumentu). Jest on nie widoczny dla użytkownika podczas przeglądania dokumentu. Stanowi część elementu META, który – choć nie należy do standardu – to jednak jest zalecany przez World Wide Web Consortium i przewidziany dla potrzeb wyszukiwania informacji. Jego składnia jest następująca:

`<META name="description" content="Opis zawartości">.`

- c) **Słowa kluczowe** (ang. *keywords*) opisują treść dokumentu hipertekstowego. Są one (podobnie jak opis) nie widoczne dla użytkownika, podczas przeglądania dokumentu w przeglądarce. Także wchodzi w skład zalecanego, choć nienależącego do standardu, elementu META, który ma zastosowanie głównie dla potrzeb wyszukiwania informacji. Składnia tego elementu wygląda następująco:

`<META name="keywords" content="słowa kluczowe">.`

Wymienione elementy stron WWW przeznaczone są do wykorzystania przez wyspecjalizowane systemy, np. wyszukiwarki internetowe. Słowa kluczowe oraz opis strony umieszczony w nagłówku dokumentu są bardzo ważną informacją o zawartości strony. Według badań przeprowadzonych w firmie NEC, 34.2 % wszystkich stron WWW posiada znaczniki keywords lub description [Law99b].

Częstość występowania (liczba wystąpień) terminu k w dokumencie d_j , czyli $tf_{d,j,k}$, niezbędna we wzorze (4.11), może być wyrażona poprzez:

$$tf_{d,j,k} = tf_{d,j,k}^{BODY},$$

gdzie: $tf_{d,j,k}^{BODY}$ – częstość występowania (liczba wystąpień) terminu k w treści zasadniczej dokumentu d_j tzn. tej, która jest widoczna dla użytkownika, np. w oknie przeglądarki i która jest zawarta w części BODY opisu strony.

Zasadnym jednak wydaje się uwzględnienie wyżej wymienionych, niewidocznych dla użytkownika elementów opisu strony HTML. Wartość $tf_{d,j,k}$ można by zatem wyrazić wzorem:

$$tf_{d,j,k} = tf_{d,j,k}^{BODY} + tf_{d,j,k}^{TITLE} + tf_{d,j,k}^{DESCRIPTION} + tf_{d,j,k}^{KEYWORDS},$$

gdzie: $tf_{d,j,k}^{TITLE}$ – częstość występowania terminu k w tytule dokumentu d_j , tzn. w zawartości elementu <TITLE> nagłówka opisu strony, wyżej opisanego pogrubioną kursywą;

$tf_{d,j,k}^{DESCRIPTION}$ – częstość występowania terminu k w opisie dokumentu d_j , tzn. w zawartości elementu <META name="description"> z nagłówka opisu strony; wyżej wyrażonego poprzez pogrubioną kursywę;

$tf_{d,j,k}^{KEYWORDS}$ – częstość występowania terminu k w słowach kluczowych dokumentu d_j ; inaczej jest to liczba wystąpień terminu k w zawartości elementu <META name="keywords"> z nagłówka opisu strony d_j ; wyżej opisanego pogrubioną kursywą;

Terminy, występujące w wyżej wymienionych polach nagłówka strony WWW, niosą jednak skondensowaną informację o treści całej strony. W związku z tym należy je uwzględnić przy obliczaniu podobieństwa treściowego. Można to uczynić poprzez modyfikację wzoru (4.11), tj. zmianę wagi tych terminów, które znalazły się w polach nagłówka: tytuł, opis lub słowa kluczowe.

Modyfikacja polega na zmianie sposobu obliczania częstości występowania terminu k w dokumencie d_j i wprowadzeniu zmienionej funkcji $tf'_{d_j,k}$ zamiast $tf_{d_j,k}$. Można ją wyrazić wzorem:

$$tf'_{d_j,k} = tf_{d_j,k}^{BODY} + \alpha tf_{d_j,k}^{TITLE} + \beta tf_{d_j,k}^{DESCRIPTION} + \gamma tf_{d_j,k}^{KEYWORDS}, \quad (4.13)$$

gdzie: $\alpha, \beta, \gamma \geq 0$ to współczynniki umożliwiające określenie znaczenia terminów pochodzących z odpowiednio: tytułu, opisu dokumentu, słów kluczowych.

Dla $\alpha, \beta, \gamma > 1$ następuje uwypuklenie ważności (wagi) terminów należących do elementów: tytuł, opis i słowa kluczowe z nagłówka opisu strony.

Zaproponowane w postaci wzoru (4.13) uwzględnienie specyficznych cech stron WWW wydaje się być lepsze od przedstawionego przez Weissa i innych [Wei96] (pkt. 3.1.3). W tym drugim przypadku poprzez wagę $w_{d_j,k}^{tt}$ uwzględniane były wszystkie terminy pochodzące z całego nagłówka HTML. Nie jest to dobre rozwiązanie, gdyż w nagłówku występuje bardzo wiele pól niemających nic wspólnego z treścią dokumentu, a jedynie związanych z organizacją strony, np. opis programu użytego do utworzenia strony.

Podstawiając $tf'_{d_j,k}$ ze wzoru (4.13), zamiast $tf_{d_j,k}$ we wzorze (4.11), uzyskujemy zmodyfikowaną wagę terminu k w dokumencie d_j , czyli $w'_{d_j,k}$:

$$w'_{d_j,k} = \frac{tf'_{d_j,k} \log \frac{N}{nd_k}}{\sqrt{\sum_{i=1}^{nt_i} \left(tf'_{d_j,i} \log \frac{N}{nd_i} \right)^2}}.$$

Podobnie jak $w'_{d_j,k}$, wartość $w'_{d_i,k}$, należy do przedziału $[0,1]$.

Podstawiając zmodyfikowaną wagę terminu w dokumencie $w'_{d_j,k}$ do wzoru (4.10) otrzymujemy **zmodyfikowane podobieństwo treściowe**, które oznaczmy przez $S^{cc}(d_i, d_j)$. Można to wyrazić poprzez:

$$S^{cc}(d_i, d_j) = \frac{\sum_{k=1}^{nt} w'_{d_i,k} w'_{d_j,k}}{\sqrt{\sum_{k=1}^{nt} w'_{d_i,k}^2 \sum_{k=1}^{nt} w'_{d_j,k}^2}}.$$

Zmodyfikowane podobieństwo treściowe spełnia wszystkie warunki zawarte w punkcie 3 założeń dla funkcji podobieństwa (pkt. 4.1).

Podobnie jak $S^c(d_i, d_j)$, funkcja $S^{cc}(d_i, d_j)$ jest funkcją półmetryczną, tzn.:

a) dla $d_i = d_j$, otrzymujemy, za (4.12):

$$(\forall d_i \in D) S^{cc}(d_i, d_i) = 1 \geq S^c(d_i, d_j), \quad (4.14)$$

b) jest symetryczna (3.2),

c) spełnia warunek trójkąta (3.3).

Nagłówek strony HTML może zawierać także wiele innych, niewymienionych wyżej znaczników. Jednak nie powinny one być brane pod uwagę przy wyznaczaniu podobieństwa treści doku-

mentów, gdyż dotyczą przede wszystkim opisu elementów aktywnych występujących na stronie, oraz sposobów formatowania, kodowania, przetwarzania i prezentacji strony użytkownikowi.

4.7. PODOBIENSTWO KOMPLEKSOWE

W poprzednich punktach zdefiniowano:

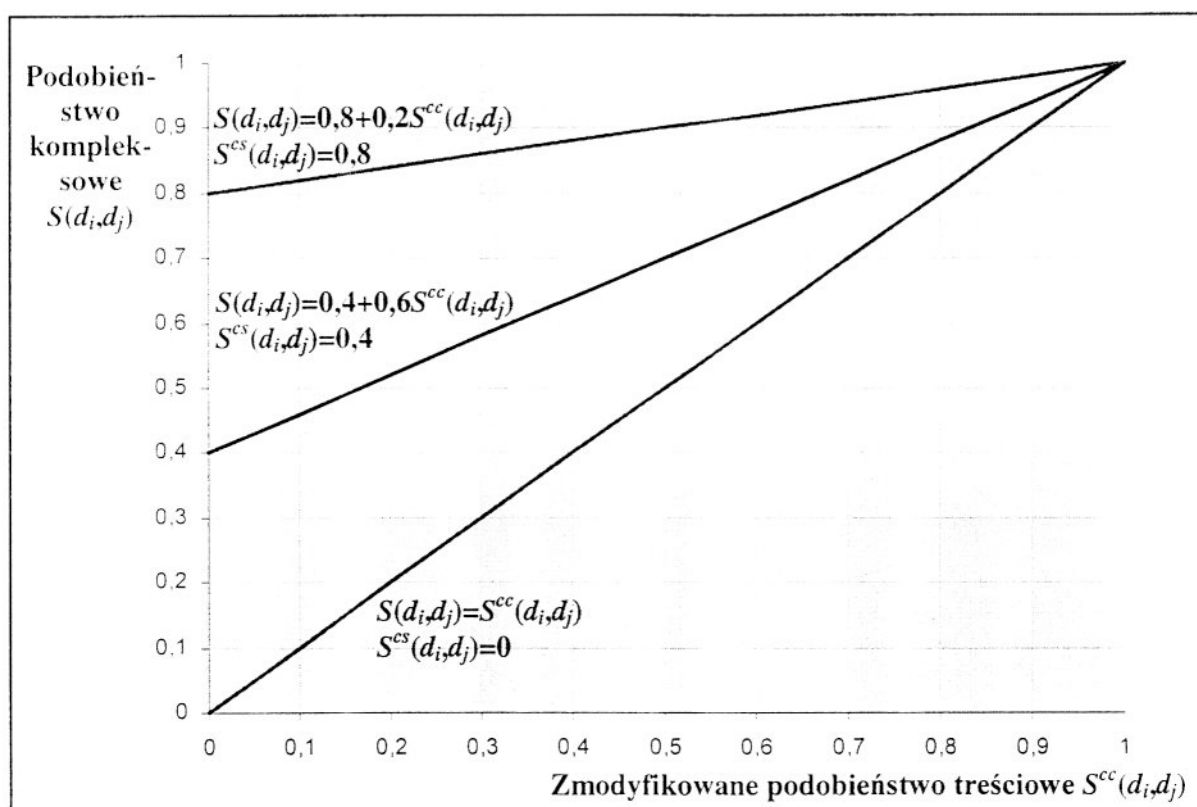
- Pierwotne podobieństwo strukturalne $S^{ps}(d_i, d_j)$ przekształcone w zmodyfikowane podobieństwo strukturalne $S^{cs}(d_i, d_j)$, które informuje o tym, że dokumenty są do siebie podobne ze względu na odsyłacze ich łączące.
- Podobieństwo treściowe $S^{cc}(d_i, d_j)$ przekształcone w zmodyfikowane podobieństwo treściowe $S^{cc}(d_i, d_j)$, które wyraża podobieństwo treści dokumentów. W sposób szczególny uwzględniono przy tym elementy specyficzne dla stron WWW.

Podobieństwo dokumentów $S(d_i, d_j)$, nazwane **podobieństwem kompleksowym** (ang. *complex similarity*) uwzględnia oba te elementy:

$$S(d_i, d_j) = S^{cs}(d_i, d_j) + S^{cc}(d_i, d_j) (1 - S^{cs}(d_i, d_j)), \quad d_i, d_j \in D. \quad (4.15)$$

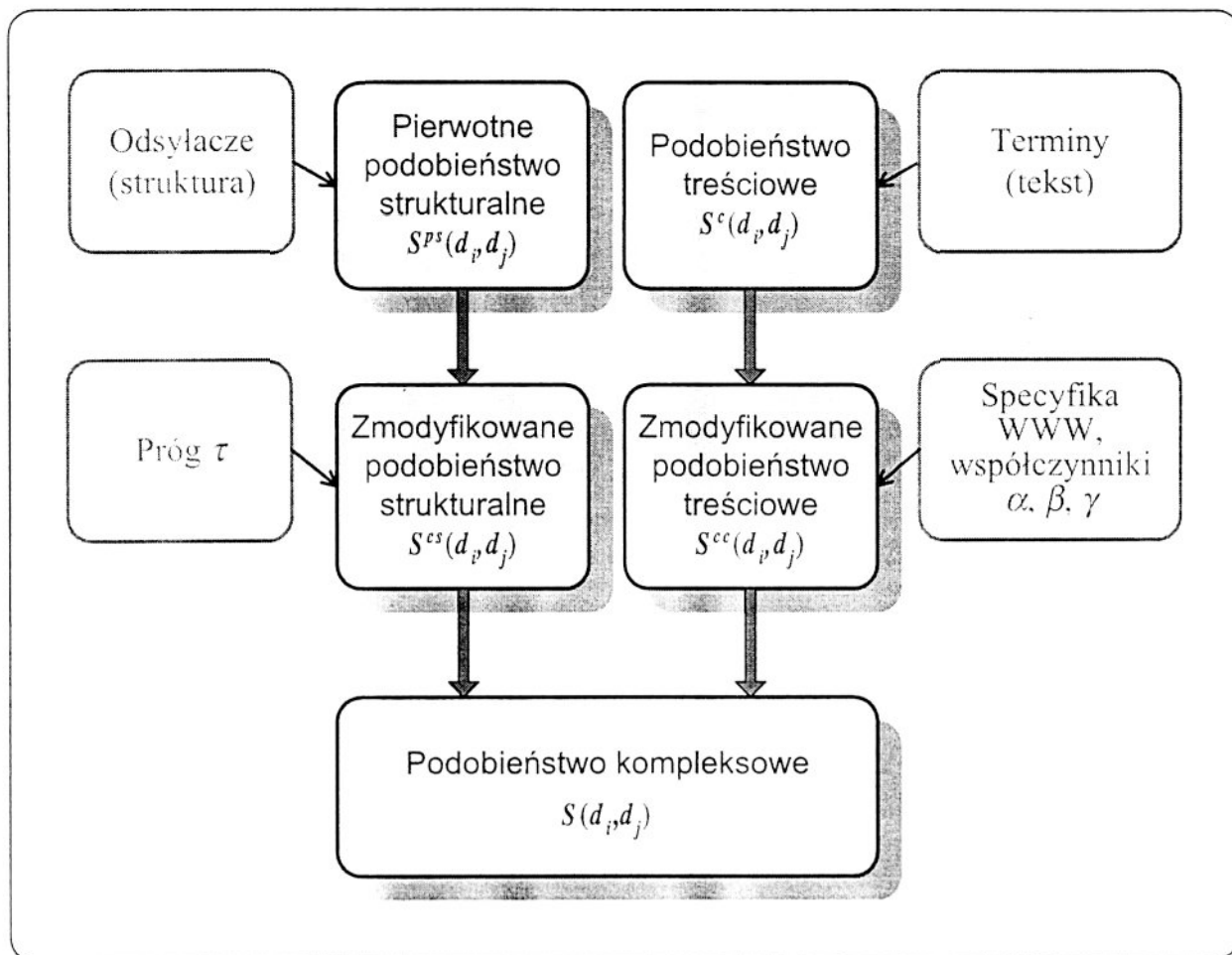
Wartość $S(d_i, d_j)$ wynosi co najmniej wartość $S^{cs}(d_i, d_j)$. Zmodyfikowane podobieństwo strukturalne $S^{cs}(d_i, d_j)$ w podobieństwie kompleksowym $S(d_i, d_j)$ może być zwiększone przez zmodyfikowane podobieństwo treściowe, nawet do wartości 1 (dla $S^{cc}(d_i, d_j)=1$).

Na rys. 4.3 przedstawiono wykresy zależności funkcji podobieństwa kompleksowego od zmodyfikowanego podobieństwa treściowego, dla 3 różnych wartości zmodyfikowanego podobieństwa strukturalnego, tj. dla $S^{cs}(d_i, d_j)=0.8$, $S^{cs}(d_i, d_j)=0.6$ oraz $S^{cs}(d_i, d_j)=0$.



Rys. 4.3. Wykres zależności wartości podobieństwa kompleksowego $S(d_i, d_j)$ funkcji podobieństwa treściowego $S^{cc}(d_i, d_j)$, przy ustalonym zmodyfikowanym podobieństwie strukturalnym.

Schemat obliczania podobieństwa kompleksowego zilustrowano na rys. 4.4.



Rys.4.4. Schemat obliczania podobieństwa kompleksowego.

Jeżeli zmodyfikowane podobieństwo treściowe $S^{cc}(d_i, d_j) = 0$, wtedy otrzymujemy:

$$S(d_i, d_j) = S^{cs}(d_i, d_j).$$

Nawet wtedy, gdy dokumenty nie są w ogóle podobne do siebie na podstawie treści, to $S(d_i, d_j)$ nie jest zerowa, gdy istnieją odsyłacze łączące dokumenty. Ponieważ wtedy $S^{cs}(d_i, d_j) \in (\tau, 1]$, więc także $S(d_i, d_j) \in (\tau, 1]$.

Gdy nie ma odsyłaczy między dwoma dokumentami¹ d_i oraz d_j , wtedy $S^{cs}(d_i, d_j) = 0$, a w konsekwencji otrzymujemy:

$$S(d_i, d_j) = S^{cc}(d_i, d_j), \quad \text{dla } S^{cs}(d_i, d_j) = 0, \quad d_i, d_j \in D. \quad (4.16)$$

Wartości podobieństwa kompleksowego $S(d_i, d_j)$ mieszczą się w takim przypadku w przedziale $[0, 1]$.

Podobieństwo kompleksowe między danym dokumentem, a nim samym wynosi:

$$(\forall d_i \in D) S(d_i, d_i) = 0 + 1 = 1.$$

Wynika to ze wzorów (4.9) oraz (4.14). W związku z tym funkcja podobieństwa kompleksowego spełnia warunek (3.1).

Oczywistą własnością tej funkcji jest jej symetryczność, która wynika z tego, że symetryczne są wszystkie składniki sumy we wzorze (4.15).

¹ a w przypadku wersji D dla pierwotnego podobieństwa strukturalnego, gdy nie istnieją także pośrednie odsyłacze cytowania, wychodzące z tych dokumentów

Ponieważ funkcja zmodyfikowanego podobieństwa strukturalnego nie spełnia warunku trójkąta, więc także nie spełnia go funkcja podobieństwa kompleksowego, gdyż dla $S^{cc}(d_i, d_j) = 0$, mamy że $S(d_i, d_j) = S^{cs}(d_i, d_j)$.

Funkcja podobieństwa kompleksowego jest quasi-metryczną funkcją podobieństwa i nie jest funkcją półmetryczną.

5. METODA GRUPOWANIA DOKUMENTÓW HIPERTEKSTOWYCH NA PODSTAWIE DRZEWA MAKSYMALNYCH PRZEPŁYWÓW

5.1. OGÓLNY OPIS METODY GRUPOWANIA

Proponowana w niniejszej pracy metoda grupowania na podstawie drzewa maksymalnych przepływów ma na celu przekształcenie kolekcji hipertekstowej w hierarchie grup podobnych do siebie dokumentów. Jej zasadnicza idea metody opiera się na założeniu, że w grafie ważonym, powstałym z przeniesienia podobieństwa między parami dokumentów na wagi krawędzi łączących dokumenty w grafie, wartość maksymalnego przepływu między dwoma węzłami – dokumentami jest odzwierciedleniem podobieństwa semantycznego między tymi dokumentami. Efektem grupowania są hierarchie, w których dokumenty ściślej związane ze sobą znajdują się w grupach niżżej położonych w hierarchii. Miernikiem związania jest wartość maksymalnego przepływu między dowolnymi parami dokumentów z grupy¹.

W metodzie grupowania na podstawie drzewa maksymalnych przepływów można wyróżnić kilka zasadniczych etapów, które zostały schematycznie przedstawione na rys. 5.1. Grupowanie przeprowadzane jest dla kolekcji hipertekstowej wydzielonej ze środowiska hipertekstowego, czyli nieuporządkowanego zbioru dokumentów tekstowych wraz z odsyłaczami (def. 2.1, 2.2). Przygotowanie danych wejściowych stanowi pierwszy etap metody. Źródło kolekcji jest dowolne, a jedynym warunkiem jest to, że są to powiązane odsyłaczami dokumenty tekstowe. Zbiorem wejściowym może być wybrany zbiór stron z całego systemu WWW, będący na przykład odpowiedzią wyszukiwarki internetowej na zadane przez użytkownika pytanie. Innym przykładem danych wejściowych mogą być strony WWW zgromadzone przez agenta informacyjnego, znającego preferencje użytkownika (profil użytkownika). Kolekcją wejściową może być także zbiór powiązanych ze sobą wiadomości dostarczanych przez prasowe agencje informacyjne, czy zbiór tekstów aktów prawnych.

Z kolekcji hipertekstowej tworzony jest nieorientowany graf ważony. Stanowi to drugi etap metody. Węzłami tego grafu są dokumenty hipertekstowe, a krawędziami wzajemne podobieństwo pomiędzy dokumentami łączonymi przez tę krawędź. Można tutaj zastosować dowolną, symetryczną funkcję podobieństwa², na przykład którąś z tych, opisanych w pkt. 3.1. W niniejszej pracy, dla potrzeb przeprowadzonego dalej eksperymentu, wybrano funkcję podobieństwa kompleksowego, przedstawioną w poprzednim rozdziale.

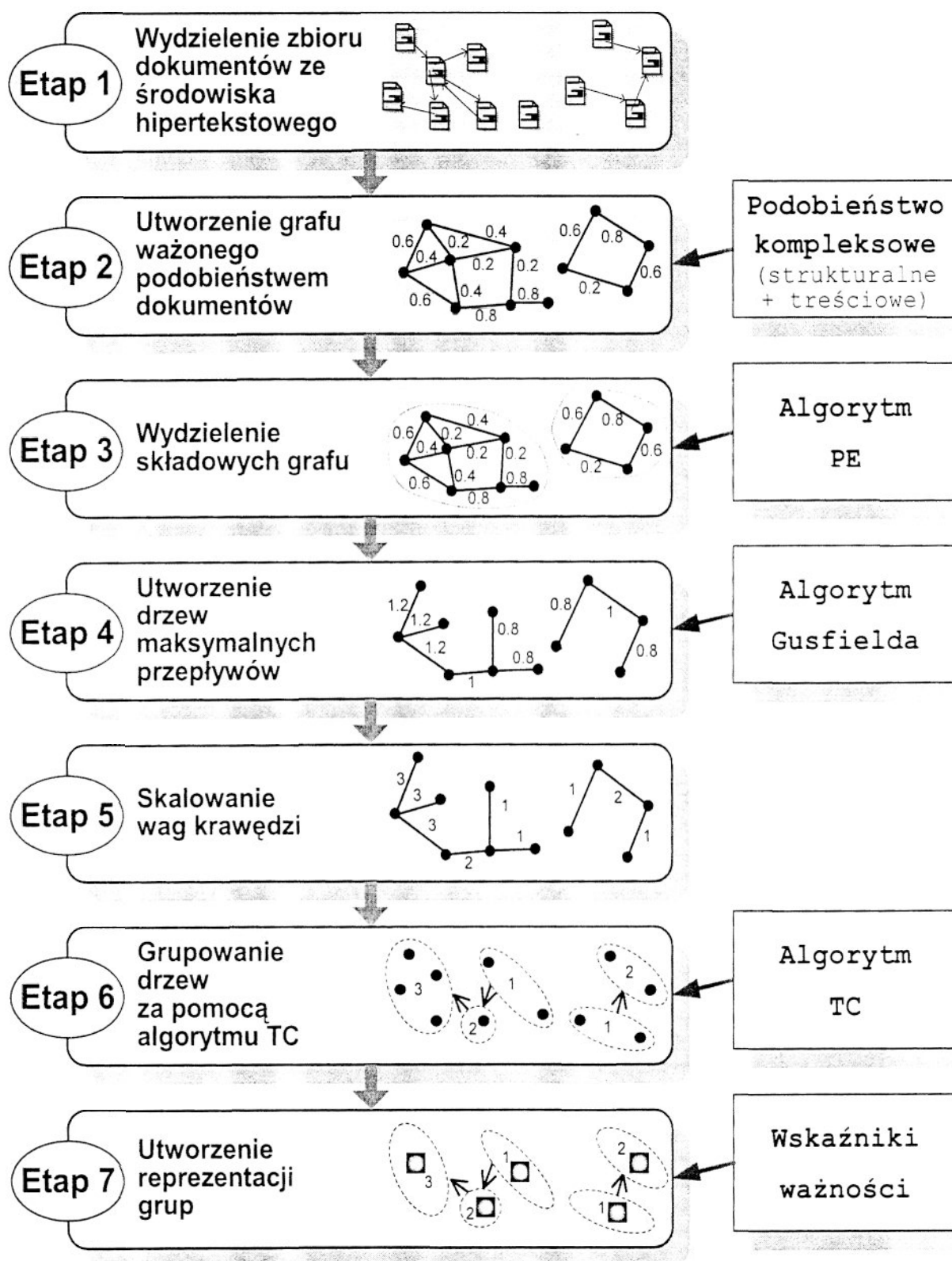
Potraktowanie kolekcji hipertekstowej jako podzbioru szerszego środowiska hipertekstowego umożliwia wykorzystanie (w funkcji podobieństwa kompleksowego) informacji zawartej w odsyłaczach prowadzących z dokumentów należących do kolekcji do dokumentów, które w kolekcji się nie znalazły, a należą do środowiska. Dzięki temu implementacja metody uwzględnia specyfikę otwartego i zmiennego w czasie środowiska, jakim jest system WWW.

Poprzez podobieństwo kompleksowe, w procesie grupowania wykorzystano zarówno informację zawartą w odsyłaczach, jak i tę, którą niesie w sobie treść dokumentów. Istnienie krawędzi między dwoma dokumentami (węzłami w grafie) oznacza, iż ich wzajemne podobieństwo przekro-

¹ Wstępna wersja metody grupowania na podstawie drzewa maksymalnych przepływów została opublikowana na konferencji MISSI'98 [Kaz98a], a końcowa w pracy [Kaz00c].

² Warunek symetryczności wynika z założenia, że graf jest nieorientowany.

czyli pewien próg τ . Jeżeli podobieństwo to jest mniejsze od progu τ , wtedy nie istnieje krawędź łącząca te dokumenty.



Rys. 5.1. Schemat grupowania dokumentów hipertekstowych na podstawie drzewa maksymalnych przepływów.

Następnie z grafu, który może być niespójny, z wykorzystaniem opracowanego algorytmu PE, wydzielane są składowe, czyli maksymalne podgrafy spójne (etap 3).

Z każdej takiej składowej z osobna tworzone jest drzewo maksymalnych przepływów z wykorzystaniem algorytmu Gusfielda (etap 4).

W powstałych w ten sposób drzewach przeskalowujemy się wagi krawędzi, przenosząc je z liczb rzeczywistych w liczby naturalne (etap 5). W procesie skalowania występują parametry, od doboru których zależy to, czy powstałe hierarchie będą bardziej lub mniej rozbudowane¹.

Następnie do każdego drzewa stosowany jest – opracowany dla potrzeb metody - algorytm TC (*tree clustering*) grupowania drzewa maksymalnych przepływów (etap 6). Z każdego drzewa powstaje jedna hierarchia grup węzłów (dokumentów hipertekstowych). Powstałe hierarchie są wolne od niejednoznaczności drzewa wygenerowanego przez algorytm Gusfielda, tzn. dla danej składowej grafu ważonego, powstaje zawsze jedna i ta sama hierarchia.

Każda hierarchia odzwierciedla (przeniesione poprzez wartości maksymalnych przepływów) powiązanie dokumentów w ramach grup, przy czym dokumenty należące do korzenia są najmniej związane ze sobą. Na przykład, jeżeli w pewnym zbiorze dokumentów Z_1 maksymalny przepływ dla dowolnej pary dokumentów jest równy lub większy od pewnej wartości c_1 i w Z_1 istnieje podzbiór Z_2 , w którym maksymalny przepływ jest większy lub równy c_2 (przy czym $c_2 > c_1$), wtedy ze zbioru Z_1 tworzymy dwa zbiory: Z_2 oraz Z_1' . Z_1' zawiera dokumenty pozostałe z Z_1 po wydzieleniu Z_2 . Traktując zbiór Z_2 , jako dziecko Z_1' otrzymujemy hierarchię grup dokumentów. Hierarchia będzie bardziej rozbudowana, jeżeli w zbiorze Z_1 istnieje więcej podobnych do Z_2 podzbiorów lub w Z_2 istnieją podzbiory o maksymalnych przepływie c_3 większym niż c_2 .

Taka hierarchia jest inna niż hierarchie uzyskiwane w metodach HACM, opisanych w pkt. 3.5, w których dokumenty są podzielone na grupy umieszczane w liściach. Tutaj dokumenty mogą należeć do elementów hierarchii wyżej położonych. Konsekwencją takiego podejścia jest możliwość istnienia grup pustych.

Dla każdej grupy dokumentów (węzłów) wyznacza się jej reprezentację, aby ułatwić ich prezentację użytkownikowi, na przykład przy wyszukiwaniu informacji.

5.2. GRAF WAŻONY

Nieważony, zorientowany graf jest podstawową formą reprezentacji hipertekstu, w której krawędzie odpowiadają odsyłaczom, a węzły – dokumentom tekstowym. W opisywanej metodzie grupowania wykorzystuje się jednak graf ważony, nieorientowany, w którym krawędzie odpowiadają nie konkretnym odsyłaczom, ale podobieństwu dokumentów łączonych przez krawędź.

¹ Chodzi o parametry, których dobór ma wpływ na to, że wejściowa kolekcja jest dzielona na większą lub mniejszą liczbę grup, z których powstają hierarchie o różnej liczbie poziomów.

Definicja 5.1.

Grafem ważonym dla kolekcji hipertekstowej $HT(D,L)$ nazywamy parę (V, E^G) , gdzie: V – zbiór węzłów, a E^G – zbiór krawędzi, taką, że:

1. $V=D$.
2. E^G – zbiór krawędzi składający się z nieuporządkowanych par węzłów postaci $\{v_i, v_j\}$, gdzie $v_i, v_j \in V$, $v_i \neq v_j$.
3. Z każdą krawędzią $\{v_i, v_j\} \in E^G$ związana jest jedna liczba rzeczywista, dodatnia, zwana wagą, oznaczana przez w_{ij}^G , przy czym:

$$w_{ij}^G = S(d_i, d_j),$$

gdzie: $S(d_i, d_j)$ – funkcja podobieństwa kompleksowego między dokumentem d_i oraz d_j .

Tak zdefiniowany graf oznaczany będzie przez $G(V, E^G)$,

Warunek 1 służy jedynie wprowadzeniu nowego oznaczenia dla zbioru węzłów V (zamiast D) i pojedynczych węzłów – v_i, v_j (zamiast d_i, d_j). Ma to na celu zwiększenie przejrzystości tej części pracy, która będzie związana bliżej z teorią grafów. Każdemu węzłowi v_i należącemu do zbioru V , odpowiada dokładnie jeden dokument d_i , należący do zbioru D .

Warunek 2 odpowiada temu, że graf jest nieorientowany i pozbawiony pętli.

Przypomnijmy (za def. 1.1), że jeżeli $\{v_i, v_j\} \in E^G$, to e_{ij}^G oznacza krawędź $\{v_i, v_j\}$.

Graf ważony dla kolekcji hipertekstowej $HT(D,L)$ nazywać się będzie dalej **grafem kolekcji hipertekstowej** lub krótko **grafem ważonym**.

Zbiór E^G grafu $G(V, E^G)$ składa się z pojedynczych krawędzi e_{ij}^G , którym nadawana jest jedna waga w_{ij}^G . Krawędź e_{ij}^G łączy węzły v_i oraz v_j , należące do V , co odpowiada połączeniu dokumentów d_i z d_j , należących do D . Założeniem dla zbudowania grafu $G(V, E^G)$ dla potrzeb grupowania dokumentów na podstawie drzewa maksymalnych przepływów jest to, że waga w_{ij}^G odzwierciedla podobieństwo tematyczne między węzłami (dokumentami), które łączy, to znaczy podobieństwo między v_i oraz v_j (między d_i oraz d_j). W niniejszej pracy, do ważenia krawędzi grafu, wykorzystano funkcję podobieństwa kompleksowego $S(d_i, d_j)$ opisaną w pkt. 4.7. Mamy więc:

$$w_{ij}^G = S(d_i, d_j), \text{ dla } S(d_i, d_j) > 0. \quad (5.1)$$

Podobieństwo $S(d_i, d_j)$ przedstawia wzór (4.15), w którym występuje zmodyfikowane podobieństwo treściowe $S^{cc}(d_i, d_j)$. Duża część par dokumentów $\{d_i, d_j\}$ dla systemu WWW ma przynajmniej jeden termin wspólny. Na stronach WWW bardzo często pojawiają się specyficzne dla tego systemu słowa, np. „home”, „page”, „www”, „here”. W konsekwencji, wartość zmodyfikowanego podobieństwa treściowego $S^{cc}(d_i, d_j)$ będzie (dla dużej części par $\{d_i, d_j\}$) większa od 0. W związku z tym nawet przy braku odsyłaczy między parą dokumentów $\{d_i, d_j\}$ ¹, wartość $S(d_i, d_j) > 0$. W wyniku zastosowania wzoru (5.1) otrzymalibyśmy więc „prawie pełny” graf ważony². Chcąc pominąć przy dalszym przetwarzaniu mało istotne krawędzie, które łączą dokumen-

¹ dla wersji D pierwotnego podobieństwa strukturalnego także przy braku odsyłaczy pośrednich

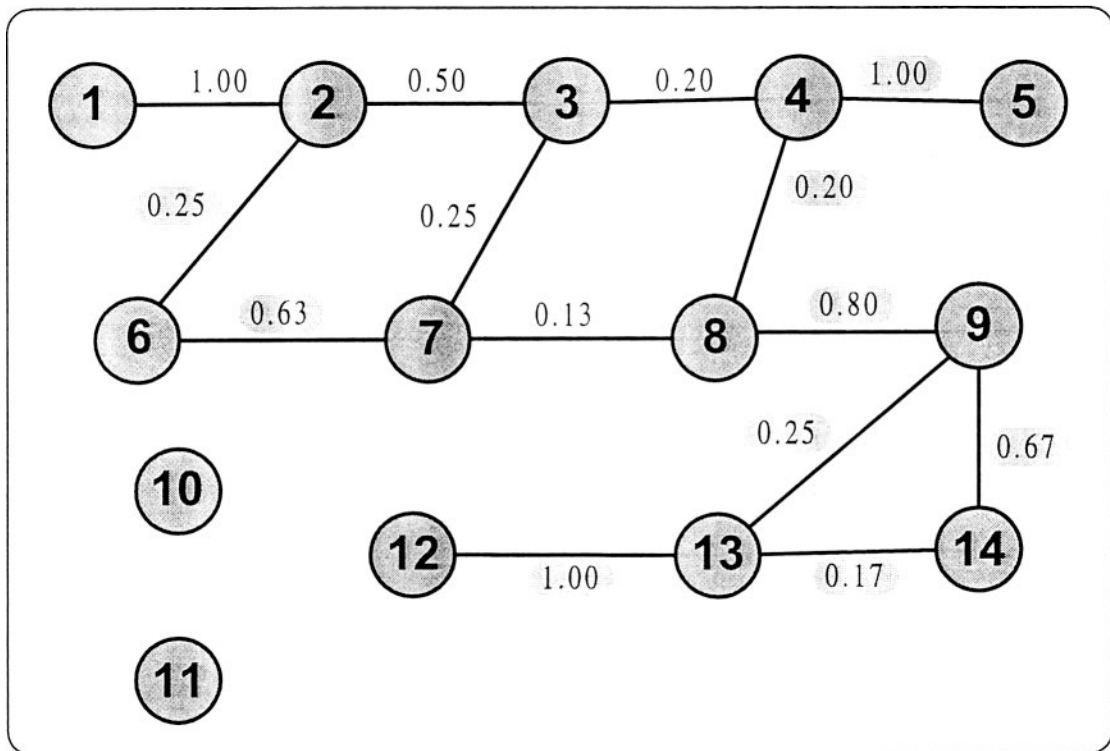
² Graf pełny to taki, w którym każdy węzeł jest połączony krawędzią z każdym innym węzłem.

ty, niezwiązane przez odsyłacze oraz dla których podobieństwo treściowe między nimi jest małe, wprowadzono **próg istotności podobieństwa** τ , przy czym $\tau \in [0,1)$. W grafie będzie istnieć krawędź między węzłem v_i oraz v_j tylko wtedy, gdy wartość funkcji podobieństwa kompleksowego między odpowiadającymi węzłom dokumentami d_i oraz d_j będzie większa od progu τ , czyli $S(d_i, d_j) > \tau$. W przeciwnym razie między węzłami v_i oraz v_j nie będzie krawędzi. Wagę krawędzi można więc opisać korzystając ze wzoru:

$$w_{ij}^G = S(d_i, d_j), \quad \text{dla } S(d_i, d_j) > \tau, \quad i \neq j, \quad d_i, d_j \in D \quad (5.2)$$

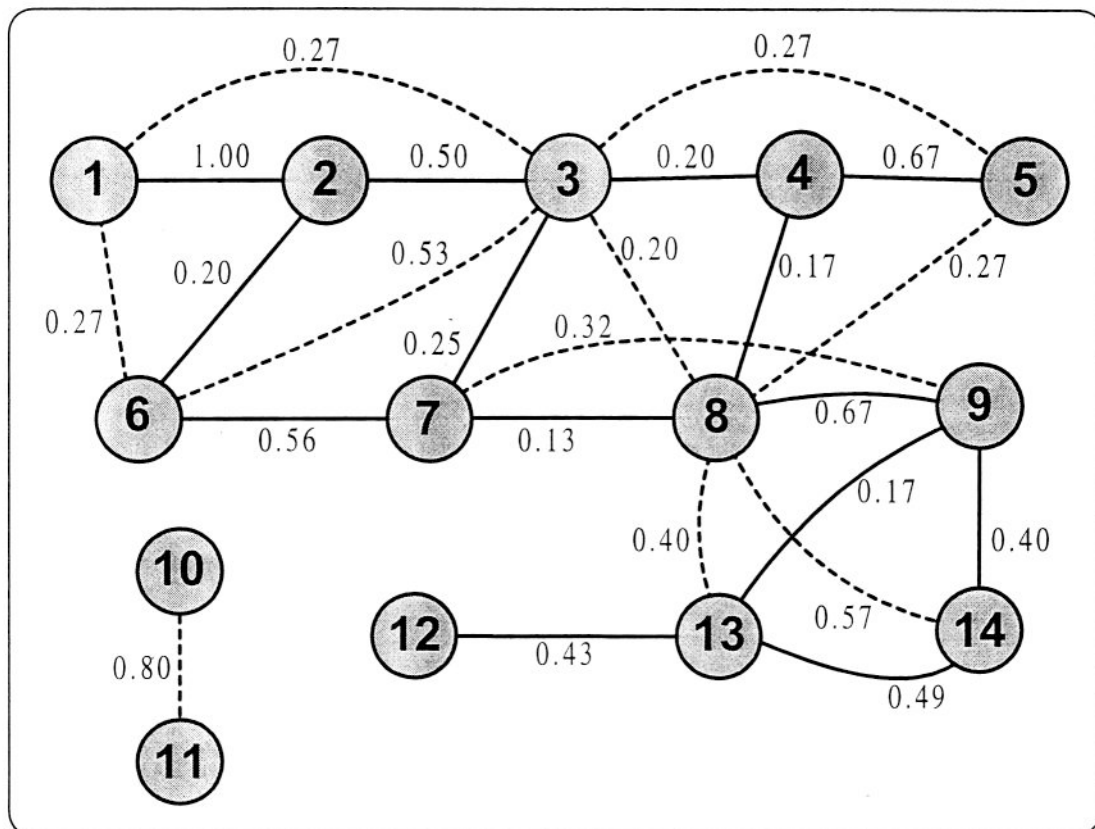
Zastosowanie progu τ ma na celu wyeliminowanie w grafie krawędzi, które łączyłyby dwa dokumenty mało do siebie podobne.

Wagi krawędzi w_{ij}^G należą do przedziału $(\tau, 1]$.



Rys. 5.2. Graf ważony utworzony dla kolekcji z rys. 4.1 oraz wersji B pierwotnego podobieństwa strukturalnego.

Dla kolekcji dokumentów, przedstawionej na rys. 4.1, utworzono grafy ważone wartością podobieństwa kompleksowego: rys. 5.2 i 5.3. Parametry zostały tak dobrane, aby uwydatnić znaczenie pierwotnego podobieństwa strukturalnego i różnice między wersją B oraz D dla tego podobieństwa. W przykładzie z rys. 5.2 wykorzystano wersję B, a z rys. 5.3 – wersję D. Założono, że współczynnik $\tau=0$ oraz nie uwzględniono zmodyfikowanego podobieństwa treściowego, tj. $S^{cc}(d_i, d_j)=0$, czyli $w_{ij}^G = S(d_i, d_j) = S^{ps}(d_i, d_j)$. Na rys. 5.3, linią przerywaną zaznaczono połączenia, które nie występują między węzłami na rys. 5.2. Jest to związane z tym, że wersja D funkcji pierwotnego podobieństwa strukturalnego między dwoma dokumentami uwzględnia (prócz odsyłaczy bezpośrednio łączących dokumenty) także odsyłacze pośrednie cytowania (połączenia do wspólnych potomków). Część wartości wag krawędzi, czyli wartości podobieństw kompleksowych, dla wersji B i D, pokrywa się. Te, które są różne, na rys. 5.2 oznaczono szarym tłem.



Rys. 5.3. Graf ważony utworzony dla kolekcji z rys. 4.1 oraz wersji D pierwotnego podobieństwa strukturalnego.

Założeniem metody grupowania wykorzystującej funkcję podobieństwa kompleksowego jest to, że krawędzie łączące węzły grafu powstają z przeniesienia odsyłaczy łączących dokumenty. Dokonuje się tego między innymi poprzez przekształcenie pierwotnego podobieństwa strukturalnego $S^{ps}(d_i, d_j)$ w zmodyfikowane podobieństwo strukturalne $S^{cs}(d_i, d_j)$ tak, aby zapewnić dla $S^{ps}(d_i, d_j) > 0$ przekroczenie przez $S^{cs}(d_i, d_j)$ progu τ . Jeżeli między dwoma dokumentami istnieje bezpośredni (lub pośredni¹) odsyłacz hipertekstowy (czyli $S^{cs}(d_i, d_j) > \tau$), wtedy istnieje krawędź łącząca węzły odpowiadające tym dokumentom w grafie ważonym.

Wzór (4.15) na funkcję podobieństwa kompleksowego ma tę własność, że dla zmodyfikowanego podobieństwa strukturalnego $S^{cs}(d_i, d_j) = 0$, czyli wtedy, gdy nie istnieją bezpośrednie lub pośrednie odsyłacze między dokumentem d_i oraz d_j , przybiera postać (4.16). Jeżeli wartość $S^{cs}(d_i, d_j)$ przekroczy próg τ , wtedy otrzymujemy: $S(d_i, d_j) > \tau$. Po zastosowaniu wzoru (5.2) uzyskamy krawędź między parą węzłów - dokumentów $\{d_i, d_j\}$, dla której, jak wspomniano, nie istnieją łączące dokumenty odsyłacze bezpośrednie lub pośrednie. Warunkiem jest tylko stosunkowo silne, zmodyfikowane podobieństwo treściowe, które zapewniłoby przekroczenie przez $S^{cc}(d_i, d_j)$ progu τ . Można to rozumieć w następujący sposób: jeżeli istnieje odsyłacz hipertekstowy między dwoma dokumentami, to krawędź w grafie odpowiada temu odsyłaczowi; jeżeli jednak brak jest odsyłacza, to ewentualna krawędź odpowiada dużemu podobieństwu tekstów obu dokumentów.

Po podstawieniu $S(d_i, d_j)$ ze wzoru (4.16) do (5.2) otrzymujemy:

$$w_{ij}^G = S^{cc}(d_i, d_j), \quad \text{dla } S^{cc}(d_i, d_j) > \tau, \quad S^{cs}(d_i, d_j) = 0, \quad i \neq j, \quad d_i, d_j \in D \quad (5.3)$$

Wzór (5.3) jest wzorem na prosty mechanizm automatycznego tworzenia krawędzi grafu, który jest bliski metodom automatycznego tworzenia hipertekstu, czyli generowania odsyłaczy na podstawie podobieństwa semantycznego, przedstawionym między innymi w [IPM97].

¹ dla wersji D pierwotnego podobieństwa strukturalnego

5.3. WYDZIELENIE SKŁADOWYCH GRAFU

W trzecim etapie metody grupowania z grafu $G(V, E^G)$ wydzielane są składowe, czyli maksymalne podgrafy spójne, które oznaczamy przez $G^S(V^S, E^S)$.

Z definicji składowej (1.5) wynika, że:

1. Zbiór węzłów składowej grafu ważonego $V^S \subseteq V$.
2. Zbiór krawędzi składowej E^S zawiera wszystkie krawędzie z E^G , których końcami są węzły należące do V^S .

Składowe grafu $G(V, E^G)$ oznaczmy przez:

$$G_i^S(V_i^S, E_i^S), \quad 1 \leq i \leq M,$$

gdzie: $G_i^S(V_i^S, E_i^S)$ – i -ta składową w grafie $G(V, E^G)$;

M – liczba wszystkich, różnych składowych w grafie $G(V, E^G)$.

Konsekwencją wydzielenia składowych z grafu ważonego jest powstanie w kolejnych etapach metody tyłu hierarchii, ile jest różnych składowych w grafie.

Kolejne etapy metody będą dotyczyły tylko jednej składowej, więc algorytmy zawarte w tych etapach należy zastosować osobno, dla każdej z tych składowych.

Na rys. 5.1 (etap 1) przedstawiona jest kolekcja dokumentów hipertekstowych, dla której został utworzony graf ważony (etap 2). Wydzielone zostały z niego dwie składowe (etap 3), z których powstały także dwie hierarchie (etap 6).

Do wydzielenia składowych opracowano algorytm PE (*partition extraction*), w którym przetwarzane są, po kolei, wszystkie krawędzie grafu $G(V, E^G)$:

Algorytm PE wydzielenia składowych grafu

Wejście: graf $G(V, E^G)$.

Wyjście: zbiór X składowych grafu $G_i^S(V_i^S, E_i^S)$, $i=1, 2, \dots, M$.

Krok 1. Zainicjuj zbiór X składowych grafu wartością $\emptyset$.

Krok 2. Weź krawędź e_{ij}^G ze zbioru E^G .

Krok 3. Sprawdź, czy węzły końcowe v_i oraz v_j krawędzi e_{ij}^G , należą do składowych ze zbioru X :

- a) jeżeli należą do tej samej składowej $G^S(V^S, E^S)$, dodaj krawędź e_{ij}^G do E^S ;
- b) jeżeli należą do różnych składowych, połącz te składowe w jedną składową i dodaj do niej krawędź e_{ij}^G ;
- c) jeżeli jeden z węzłów końcowych v_i lub v_j należy do składowej $G^S(V^S, E^S)$, a drugi nie należy do żadnej składowej z X , wtedy dodaj drugi węzeł do V^S a krawędź e_{ij}^G do E^S ;
- d) jeżeli żaden z węzłów v_i lub v_j nie należy do składowej w X , utwórz z $V^S = \{v_i, v_j\}$ oraz $E^S = \{e_{ij}^G\}$ nową składową $G^S(V^S, E^S)$ w X .

Krok 4. Usuń krawędź e_{ij}^G z E^G . Jeżeli $E^G \neq \emptyset$, powtórz kroki 2-4.

Warunkiem istnienia składowej jest to, że zbiór jej węzłów V^S jest niepusty. Zbiór krawędzi E^S może być jednak pusty. W takim przypadku składowa jest złożona tylko z jednego dokumentu (węzła) izolowanego v_i , tj. $G^S(\{v_i\}, \emptyset)$. Z takiej składowej w wyniku grupowania powstaje hierarchia składająca się z jednej, 1-elementowej grupy. Jest to jednak przypadek szczególny i uwzględnianie go jedynie niepotrzebnie rozbudowywałoby przedstawione dalej rozumowania. Ponieważ grupowanie ma sens tylko wtedy, gdy obiektów do grupowania jest co najmniej dwa, więc dla potrzeb dalszej części pracy przyjęto, że zbiór V^S składowej grafu ważonego składa się z co najmniej dwóch węzłów.

W skrajnym przypadku, z całej kolekcji hipertekstowej $HT(D, L)$ może powstać tyle składowych (złożonych z pojedynczych izolowanych dokumentów), ile jest dokumentów w zbiorze D , czyli N składowych i N hierarchii. Taką sytuację mamy tylko wtedy, gdy nie ma w zbiorze L ani bezpośrednich ani pośrednich odsyłaczy między dowolną parą dokumentów ze zbioru D , jak również podobieństwo treściowe, dla każdej pary dokumentów, jest nie większe od τ .

5.4. DRZEWO MAKSYMALNYCH PRZEPŁYWÓW

Z poszczególnych składowych grafu tworzone są drzewa maksymalnych przepływów (etap 4), przy czym, nie muszą być one jednocześnie drzewami minimalnych przekrojów. W tym celu można wykorzystać dowolny, poprawnie działający algorytm generowania drzewa maksymalnych

przepływów dla składowej niezorientowanego grafu ważonego¹. W literaturze znajduje się opis dwóch takich algorytmów:

- a) Algorytm EQ Gusfielda (zał. A),
- b) Algorytm Hu [Hu82].

Algorytm przedstawiony przez Hu, który tworzy drzewo maksymalnych przepływów, będące jednocześnie drzewem minimalnych przekrojów ma jednak pewne wady. Został zaprezentowany w takiej postaci, że trudno opracować program go implementujący. Poza tym, wymaga on tego, aby wykorzystywane w kolejnych krokach minimalne przekroje nie przecinały się z poprzednimi. W związku z tym, zdecydowano się na zastosowanie algorytmu Gusfielda.

W obu wyżej wymienionych algorytmach $(N-1)$ razy wywoływany jest algorytm zwracający minimalny przekrój i wartość maksymalnego przepływu dla pary węzłów. Istnieje wiele algorytmów znajdowania minimalnego przekroju i maksymalnego przepływu, w tym powszechnie znane: Forda-Fulkersona (opis dostępny m.in. w pracach [Kul86, Hu82, Cor98]), Edmondsa-Karpa, będący ulepszeniem poprzedniego [Cor98, Hu82], Dinica [Lip89, Hu82], Korzanowa [Hu82], Malhotry-Pramodh Kumara-Maheshwariego [Hu82].

W implementacji metody grupowania zastosowano algorytm opracowany w drugiej połowie lat osiemdziesiątych przez Goldberga i Tarjana [Gol86, Gol88] głównie dlatego, że program go realizujący jest ogólnie dostępny [Sko].

Wagi krawędzi należą do przedziału $(0,1]$, dlatego wartość maksymalnego przepływu (waga krawędzi drzewa maksymalnych przepływów) – będąc sumą wag krawędzi minimalnego przekroju (twierdzenie 1.2) – jest liczbą rzeczywistą większą od zera.

5.5. SKALOWANIE WAG KRAWĘDZI DRZEWA

Wartości maksymalnego przepływu, które są wagami krawędzi drzewa maksymalnych przepływów są liczbami rzeczywistymi. W szczególnym przypadku każda krawędź w drzewie może mieć inną wagę. Skalowanie ma na celu przesunięcie zakresu wartości wag krawędzi drzewa maksymalnego przepływu $T(V^S, E^T)$ do zbioru liczb naturalnych. Parametrem procesu skalowania jest liczba max^{lev} różnych wartości przeskalowanego maksymalnego przepływu. Zakres wartości maksymalnego przepływu dzielimy na przedziały o długości równej $range$, przy czym:

$$range = \frac{max^f - min^f}{max^{lev}}$$

gdzie: max^f – maksymalna wartość maksymalnego przepływu,

min^f – minimalna wartość maksymalnego przepływu.

Każdemu przedziałowi, który posiada swój kolejny numer, będący liczbą naturalną, odpowiada kolejny zakres wartości maksymalnych przepływów $f_{i,j}$. Numer przedziału będziemy nazywać **przeskalowaną wartością maksymalnego przepływu** i oznaczać: $f'_{i,j}$:

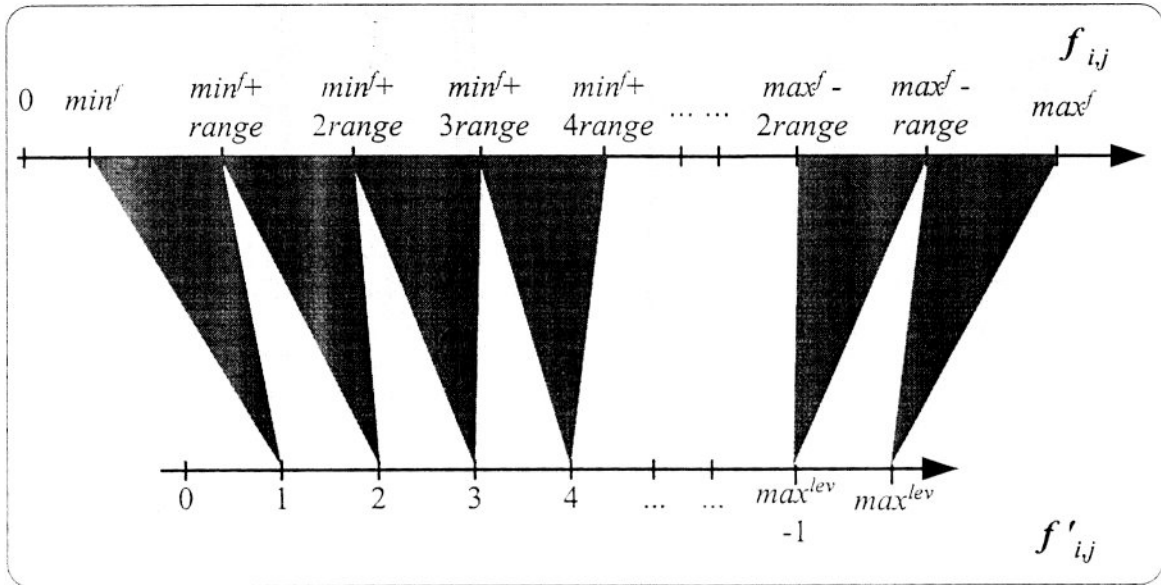
$$f'_{i,j} = \begin{cases} trunc\left(\frac{f_{i,j} - min^f}{range}\right) + 1, & \text{dla } f_{i,j} < max^f, \\ max^{lev}, & \text{dla } f_{i,j} = max^f \end{cases}, \quad (5.5)$$

¹ Na dalszym etapie grupowania do drzewa maksymalnych przepływów stosowany jest algorytm TC. Posiada on tę własność, że drzewo to może być dowolne, byleby spełniało warunek drzewa maksymalnych przepływów dla danej składowej grafu ważonego (def. 1.12). Zostanie to udowodnione w pkt. 5.8.2.

gdzie: $\text{trunc}()$ jest standardową funkcją zaokrąglenia do liczb całkowitych poprzez odrzucenie części ułamkowej.

Ostatni przedział (ostatni - maksymalny poziom) jest domknięty z obu stron. Wszystkie pozostałe tylko z lewej strony.

Proces skalowania zilustrowano na rys. 5.4.



Rys. 5.4. Ilustracja procesu skalowania wartości maksymalnego przepływu $f_{i,j}$.

Przeskalowana wartość maksymalnego przepływu jest zatem wartością maksymalnego przepływu przeniesioną z liczb rzeczywistych w zbiór liczb naturalnych z zakresu $[1, \max^{lev}]$. Zachodzi tutaj jak gdyby pierwszy, wstępny proces grupowania, tzn. wartości maksymalnego przepływu należące do jednego przedziału, odpowiadają temu samemu numerowi przedziału, tj. mają tę samą wartość przeskalowanego maksymalnego przepływu. Jest to koncepcja podobna do techniki plastrów (pkt 3.5.2), a opisaną w pracy [Maa96], która ma na celu zmniejszenie liczby grup, tworzących hierarchię.

Skalowanie można traktować jako proces normalizacji, tzn. nieograniczony z góry zbiór wartości maksymalnego przepływu jest ograniczany do przedziału $[1, \max^{lev}]$.

Wśród przeskalowanych wartości maksymalnego przepływu $f'_{i,j}$ nie muszą znajdować się wszystkie liczby naturalne z zakresu $[1, \max^{lev}]$. Wiąże się to z tym, że nie dla każdego przedziału musi istnieć wartość maksymalnego przepływu $f_{i,j}$ w składowej grafu ważonego.

Aby nie mnożyć oznaczeń, drzewo maksymalnych przepływów, w którym wagi uległy przeskalowaniu (podobnie jak drzewo z nieprzeskalowanymi wagami) będzie oznaczane przez $T(V^S, E^T)$.

5.5.1. PARAMETRY PROCESU SKALOWANIA

W procesie skalowania wykorzystywane są trzy zasadnicze parametry: $\max^{lev}$, $\min^f$ i $\max^f$.

Na ogół z kolekcji D można na ogół wydzielić więcej niż jedną składową. Z każdej takiej składowej powstaje jedno drzewo maksymalnych przepływów. W każdym drzewie mamy minimalne i maksymalne wartości wag krawędzi (wartości maksymalnego przepływu). Ponieważ wartość maksymalnego przepływu między dwoma węzłami składowej grafu musi być większa od najmniejszej wagi krawędzi występującej w grafie. Wartości wag krawędzi grafu ważonego są większe od τ (wzór 5.2), więc wagi krawędzi każdego drzewa maksymalnych przepływów także muszą być większe od τ .

W związku z powyższym, minimalną wartość maksymalnego przepływu $\min^f$ można wyznaczyć jako:

1. Współczynnik τ czyli dolną granicę wartości wag krawędzi grafu ważonego i drzewa maksymalnych przepływów. Jest to wartość wspólna dla wszystkich grafów ważonych, czyli wszystkich kolekcji dokumentów:

$$\min^f = \tau.$$

2. Minimalną wartość maksymalnego przepływu występującą we wszystkich drzewach maksymalnych przepływów, pochodzących z danego grafu ważonego. Wartość ta jest wspólna dla wszystkich drzew w ramach jednego grafu ważonego, czyli:

$$\min^f = \min\{f_{i,j}: v_i, v_j \in V_k^S, k=1,2,3,\dots,M\}.$$

3. Minimalną wartość maksymalnego przepływu występującą w danym, pojedynczym drzewie maksymalnych przepływów. Wartości te mogą być różne dla różnych drzew. Minimalną wartość maksymalnego przepływu w k -tym drzewie oznaczmy przez $\min_k^f$. Jej wartość można wyrazić przez:

$$\min_k^f = \min\{f_{i,j}: v_i, v_j \in V_k^S\}.$$

Wartość z punktu 2 jest nie mniejsza od wartości z punktu drugiego, a wartość z punktu trzeciego jest maksymalna.

Jeżeli jako parametr $\min^f$ wybrano by liczbę z punktu 3, wtedy każde drzewo maksymalnych przepływów (dla danego grafu) byłoby inaczej skalowane. W pozostałych przypadkach skalowanie dla każdego drzewa przebiegałoby na takich samych warunkach.

Analogicznie do $\min^f$ można rozważać dobór parametru $\max^f$. W przypadku $\max^f$ nie istnieje jednak górne ograniczenie wartości maksymalnego przepływu, wspólne dla wszystkich grafów ważonych (wszystkich kolekcji). W związku z tym jako wartość dla $\max^f$ można wybrać:

1. Maksymalną wartość maksymalnego przepływu występującą we wszystkich drzewach maksymalnych przepływów, pochodzących z danego grafu ważonego. Jest to wartość wspólna dla wszystkich drzew w ramach jednego grafu, czyli

$$\max^f = \max\{f_{i,j}: v_i, v_j \in V_k^S, k=1,2,3,\dots,M\}.$$

2. Maksymalną wartość maksymalnego przepływu występującą w danym, pojedynczym drzewie maksymalnych przepływów. Wartość ta może przyjąć różną wartość dla każdego drzewa. Maksymalną wartość maksymalnego przepływu, występującą w k -tym drzewie, oznaczmy przez $\max_k^f$. Można ją wyrazić przez:

$$\max_k^f = \max\{f_{i,j}: v_i, v_j \in V_k^S\}.$$

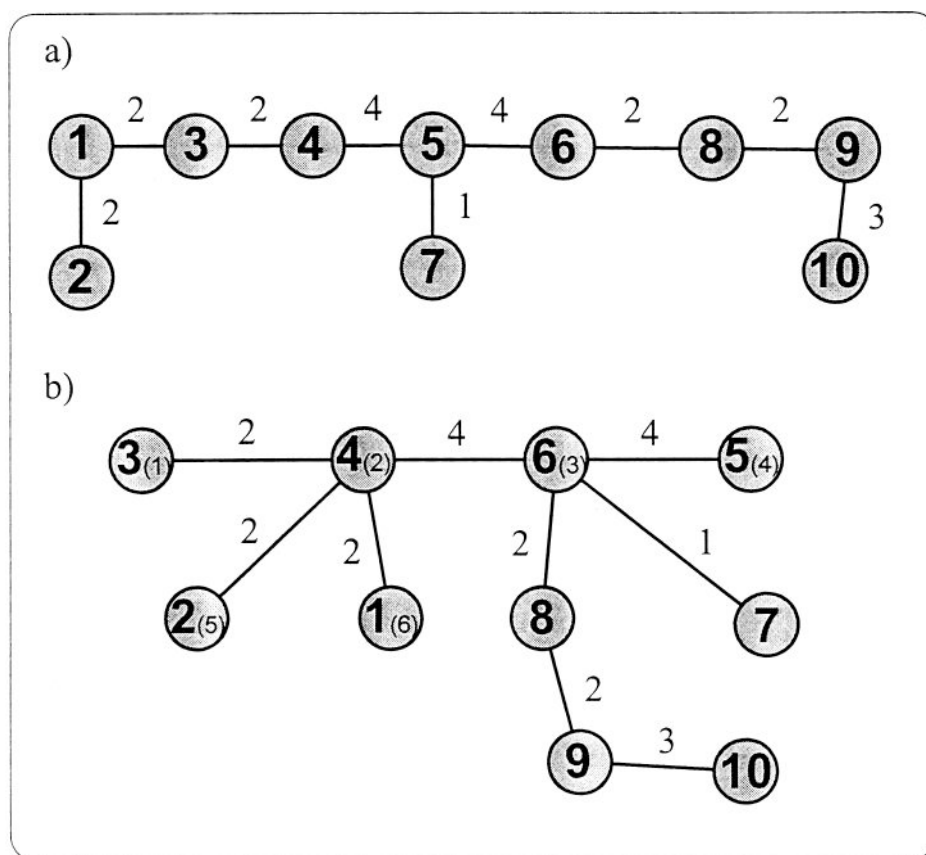
Dobór parametru $\max^{lev}$ zależy od najmniejszej odległości (różnicy wartości) między dwoma dowolnymi, różnymi wartościami maksymalnego przepływu występującymi w drzewie. Jeżeli wartość $range$ nie przekroczy połowy wartości minimalnej odległości, wtedy różnych, przeskalowanych wartości maksymalnego przepływu będzie tyle samo, co nieprzeskalowanych. W takim przypadku końcowe hierarchie grup węzłów powstałe z drzewa z przeskalowanego w ten sposób wagami i z drzewa niepoddanego procesowi przeskalowania, będą takie same tzn. będą zawierały te same węzły, w tych samych elementach hierarchii, na tych samych numerach poziomów.

Wartość $\max^{lev}$ ma zasadniczy wpływ na rozbudowanie hierarchii, tj. na liczbę grup i liczbę poziomów hierarchii. Dla małych wartości $\max^{lev}$ hierarchie będą posiadały małą liczbę poziomów. Dla $\max^{lev}=1$, nie uzyskamy żadnego podziału: wszystkie węzły znajdą się w jednej grupie. Duża wartość $\max^{lev}$ implikuje hierarchie mocno rozbudowane (grupy w nich występujące będą zawierały małą liczbę dokumentów). W przypadku zastosowania grupowania dla potrzeb prezentacji

wyników wyszukiwania, górnym ograniczeniem wartości max^{lev} są możliwości percepcyjne człowieka. Przy przeglądaniu hierarchii o dużej liczbie poziomów i grup użytkownik będzie miał poczucie zagubienia.

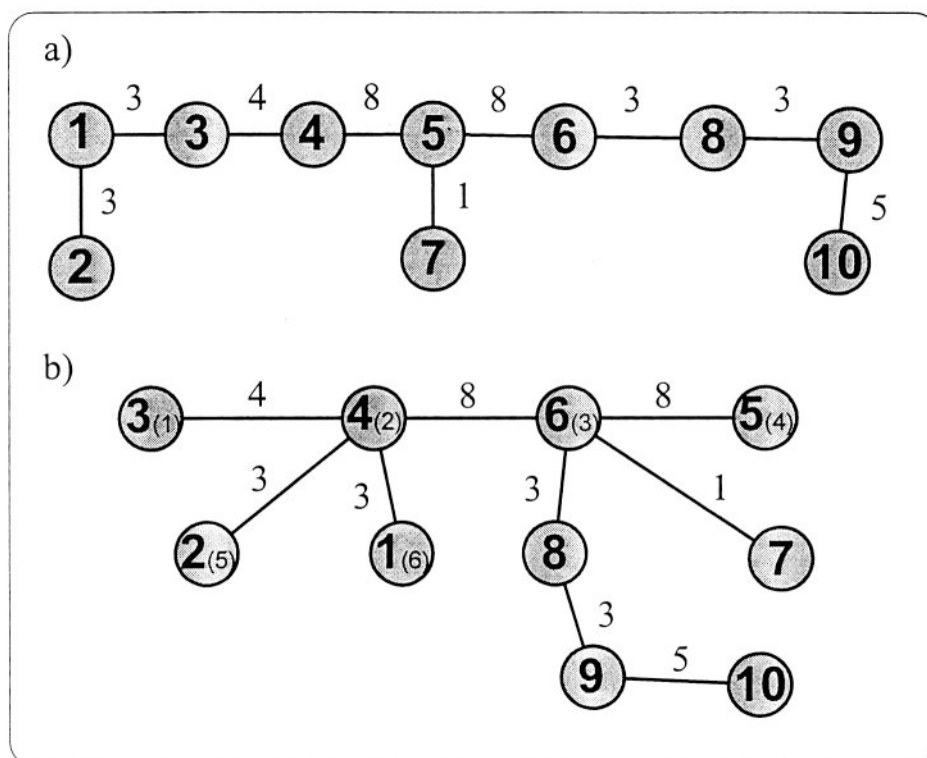
5.5.2. PRZYKŁADY SKALOWANIA

Proces skalowania można prześledzić na przykładach. Dla drzew maksymalnych przepływów z rys. 1.3 wartość $min^f=0.4$, a $max^f=2.0$. Jeżeli przyjmimy $max^{lev}=4$, wtedy szerokość przedziału $range=0.4$ i otrzymamy następujące cztery przedziały: $[0.4,0.8)$, $[0.8,1.2)$, $[1.2,1.6)$, $[1.6,2.0]$. Wagi krawędzi drzew maksymalnych przepływów z rys. 1.3 zostaną zmienione, tak jak na rys. 5.5. Otrzymamy wtedy cztery różne przeskalowane wartości wag krawędzi: 1, 2, 3, 4, mimo istnienia pięciu różnych wartości maksymalnego przepływu: 0.4, 0.8, 1.0, 1.2, 2.0. Wartości 0.8 oraz 1.0 zostały przeskalowane do tej samej liczby naturalnej 2.



Rys. 5.5. Drzewa z przeskalowanymi wartościami maksymalnych przepływów, odpowiadające drzewom z rys. 1.3 dla $max^{lev}=4$

Jeżeli natomiast przyjmimy $max^{lev}=8$, wtedy szerokość przedziału zmaleje i wyniesie: $range=0.2$. Otrzymamy osiem przedziałów: $[0.4,0.6)$, $[0.6,0.8)$, $[0.8,1.0)$, $[1.0,1.2)$, $[1.2,1.4)$, $[1.4,1.6)$, $[1.6,1.8)$, $[1.8,2.0]$. Wagi krawędzi drzew maksymalnych przepływów z rys. 1.3 zostaną zmienione jak na rys. 5.6. Wśród przeskalowanych wartości maksymalnego przepływu zaistniało pięć różnych liczb naturalnych: 1, 3, 4, 5, 8. Różnych wartości nieprzeskalowanych jest także pięć. W przedziałach o numerach 2, 6 oraz 7 nie znalazły się natomiast żadne wartości spośród tych wymienionych z rys. 1.3.



Rys. 5.6. Drzewa z przeskalowanymi wartościami maksymalnych przepływów, odpowiadające drzewom z rys. 1.3, dla $\max^{lev}=8$.

5.6. HIERARCHIA GRUP

Jako efekt końcowy metody grupowania dla każdego drzewa maksymalnych przepływów powstaje jedna hierarchia nie zachodzących na siebie grup węzłów – dokumentów.

Definicja hierarchii 5.2 jest rozszerzeniem definicji 1.14.

Definicja 5.2.

Hierarchia na zbiorze V^S nazywamy uporządkowanie $(H(V^S), \leq_H)$, spełniające warunki:

1. $H(V^S) \subseteq P(V^S) \times \mathbb{N}_+ \times \mathbb{N}$,
gdzie $P(V^S)$ – zbiór wszystkich podzbiorów zbioru V^S .
2. $(\forall (Z, c, e), (Z', c', e') \in H(V^S)) Z \neq Z' \Rightarrow Z \cap Z' = \emptyset$
gdzie: (Z, c, e) – element hierarchii (jeżeli $Z = \emptyset$, to nazywa się go: elementem pustym hierarchii, jeżeli $Z \neq \emptyset$, to jest to element niepusty hierarchii), c – związanie węzłów w zbiorze Z , e – numer elementu pustego.
3. $\bigcup \{ Z : (\exists c \in \mathbb{N}_+) (\exists e \in \mathbb{N}) (Z, c, e) \in H(V^S) \} = V^S$.
4. $(\forall (Z, c, e) \in H(V^S)) (Z \neq \emptyset \Leftrightarrow e = 0)$.
5. $(\forall (\emptyset, c, e), (\emptyset, c', e') \in H(V^S)) (\emptyset, c, e) \neq (\emptyset, c', e') \Rightarrow e \neq e'$.
6. $(\exists (Z_0, c_0, e_0)) (\forall (Z, c, e) \in H(V^S)) (Z, c, e) \leq_H (Z_0, c_0, e_0)$.
7. $(\forall (Z, c, e), (Z', c', e') \in H(V^S)) ((Z, c, e) <_H (Z', c', e') \Rightarrow c' < c)$.
8. $(\forall (Z, c, e) \in H(V^S)) \{ c \neq c_0 \Rightarrow (\exists! (Z', c', e') \in H(V^S)) [(Z, c, e) <_H (Z', c', e') \wedge \neg((\exists (Z'', c'', e'') \in H(V^S)) (Z, c, e) <_H (Z'', c'', e'') <_H (Z', c', e'))] \}$.

Zbiór $H(V^S)$ nazywamy **zbiorem elementów hierarchii**.

Z warunku 1 wynika, że zbiór elementów hierarchii składa się z trójek (Z, c, e) , które nazywamy elementami hierarchii. Ponieważ V^S to zbiór węzłów składowej grafu ważonego i zarazem węzłów drzewa maksymalnych przepływów, zatem zbiór Z to grupa węzłów drzewa maksymalnych przepływów. Natomiast c określa związanie węzłów między sobą¹. Odpowiada to wartości maksymalnego, przeskalowanego przepływu między dwoma dowolnymi węzłami $v_i, v_j \in Z$. Związanie c wyznacza stopień powiązania węzłów (dokumentów hipertekstowych), tzn. im większa wartość związania, tym bardziej podobne do siebie są dokumenty należące do zbioru Z .

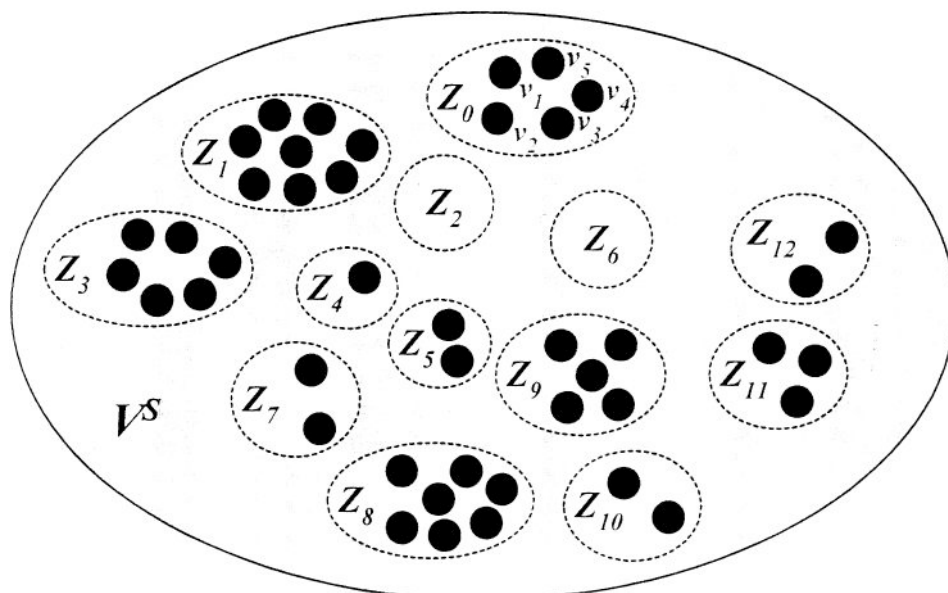
Warunek 2 definicji zapewnia to, że hierarchia składa się wyłącznie z elementów, w których zbiory Z są rozłączne. Zbiory puste oczywiście także są rozłączne.

Z warunku 3 wynika, że każdy z elementów zbioru V^S (węzeł grafu i drzewa) znajdzie się w co najmniej jednym elemencie hierarchii – w jakimś zbiorze Z .

Warunki 2 i 3 łącznie gwarantują to, że wszystkie elementy zbioru V^S są rozdzielone pomiędzy zbiory Z należące do hierarchii i każdy z nich należy tylko do dokładnie jednego zbioru Z – tylko do jednego elementu hierarchii. Podział zbioru V^S ilustruje rys. 5.7. Czarne, małe kółka oznaczają na nim elementy zbioru V^S tj. $v_1, v_2, v_3, \dots$. Każdy z elementów zbioru V^S należy do pewnego zbioru Z . Zbiory Z_2 oraz Z_6 są puste – nie posiadają żadnych elementów zbioru V^S .

Wartość e określa kolejny numer elementu pustego w ramach całej hierarchii. Jeżeli zbiór Z nie jest pusty, wtedy (na podstawie warunku 4) wartość e wynosi zero. Elementy niepuste hierarchii są rozróżniane poprzez zawartość zbioru Z .

¹ oznaczenie c pochodzi od ang. *connection*

Rys. 5.7. Podział zbioru V^S na rozłączne podzbiory Z_i .

Z warunku 5 wynika, że dwa różne elementy hierarchii posiadające pusty zbiór Z , różnią się wartością e (wartość związania c może być taka sama). Wartości e dla pustych elementów nie muszą być kolejnymi liczbami naturalnymi.

Warunek 6 określa istnienie tylko jednego, wyróżnionego elementu - korzenia hierarchii (elementu najwyższego hierarchii).

Warunek 7 określa, że to, co jest „niżej” w hierarchii ma większą wartość c – większy poziom związania węzłów. Jest to równoznaczne z tym, że to, co jest „niżej”, nie może mieć takiej samej wartości c . W związku z tym korzeń hierarchii musi mieć najmniejszą wartość związania węzłów – c_0 .

Warunek 8 zapewnia strukturę drzewiastą, w której każdy element hierarchii ma tylko jednego rodzica (bezpośrednio nad nim znajdującego się przodka).

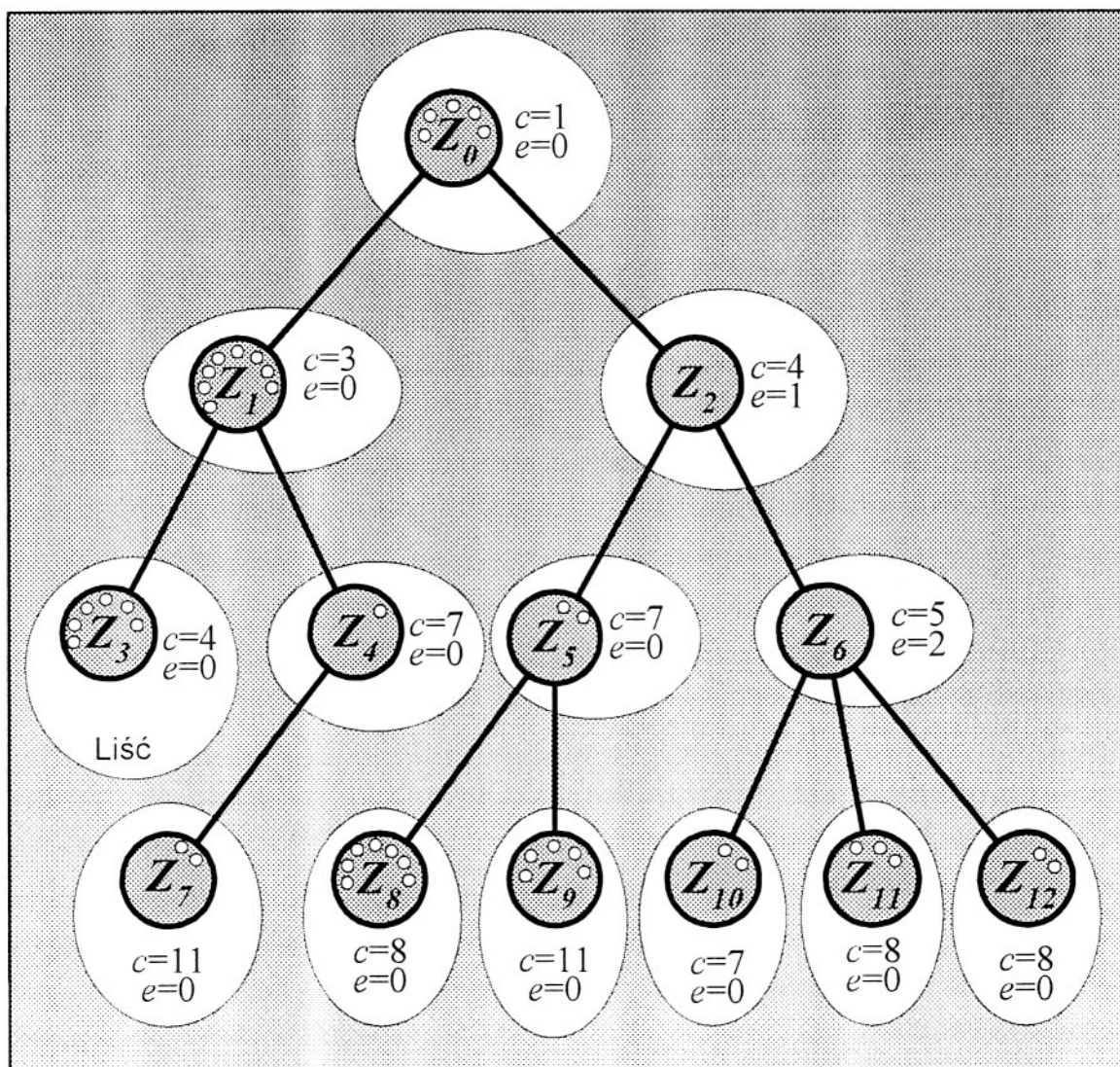
Na rys. 5.8 przedstawiono hierarchię zgodną z definicją 5.2. Elementy hierarchii (Z, c, e) , czyli zbiory Z razem ze związaniami c oraz numerami e , oznaczono białymi elipsami.

Hierarchia składa się z elementów (Z, c, e) zawierających w zbiorze Z dokumenty z kolekcji hipertekstowej $HT(D, L)$, gdyż $Z \subseteq V^S \subseteq V = D$. Liczba c określa podobieństwo dokumentów w grupie, które jest większe na niższych poziomach hierarchii. Korzeń hierarchii zawiera więc dokumenty najslabiej związane ze sobą, czyli najmniej podobne. Liście natomiast to grupy dokumentów najściślej powiązane ze sobą w ramach grupy. Zilustrowano to na rys. 5.9, zawierającym odpowiednik hierarchii z rys. 5.8.

Hierarchię można także interpretować następująco: dokumenty które należą do grup wyżej położonych zawierają w sobie treści ogólniejsze, a dokumenty z dolnych warstw hierarchii szczegółowiej opisują wybrane elementy szerszej dziedziny. Dokumenty najniżej umieszczone (liście) to takie, z których nie można wydzielić szczegółowszej tematyki. Przykładem tego typu hierarchii jest katalog Yahoo¹. Występują w nim dokumenty (odsyłacze do serwisów informacyjnych WWW), w prawie każdej kategorii (nie tylko w liściach). Dokumenty te opisują ogólnie daną

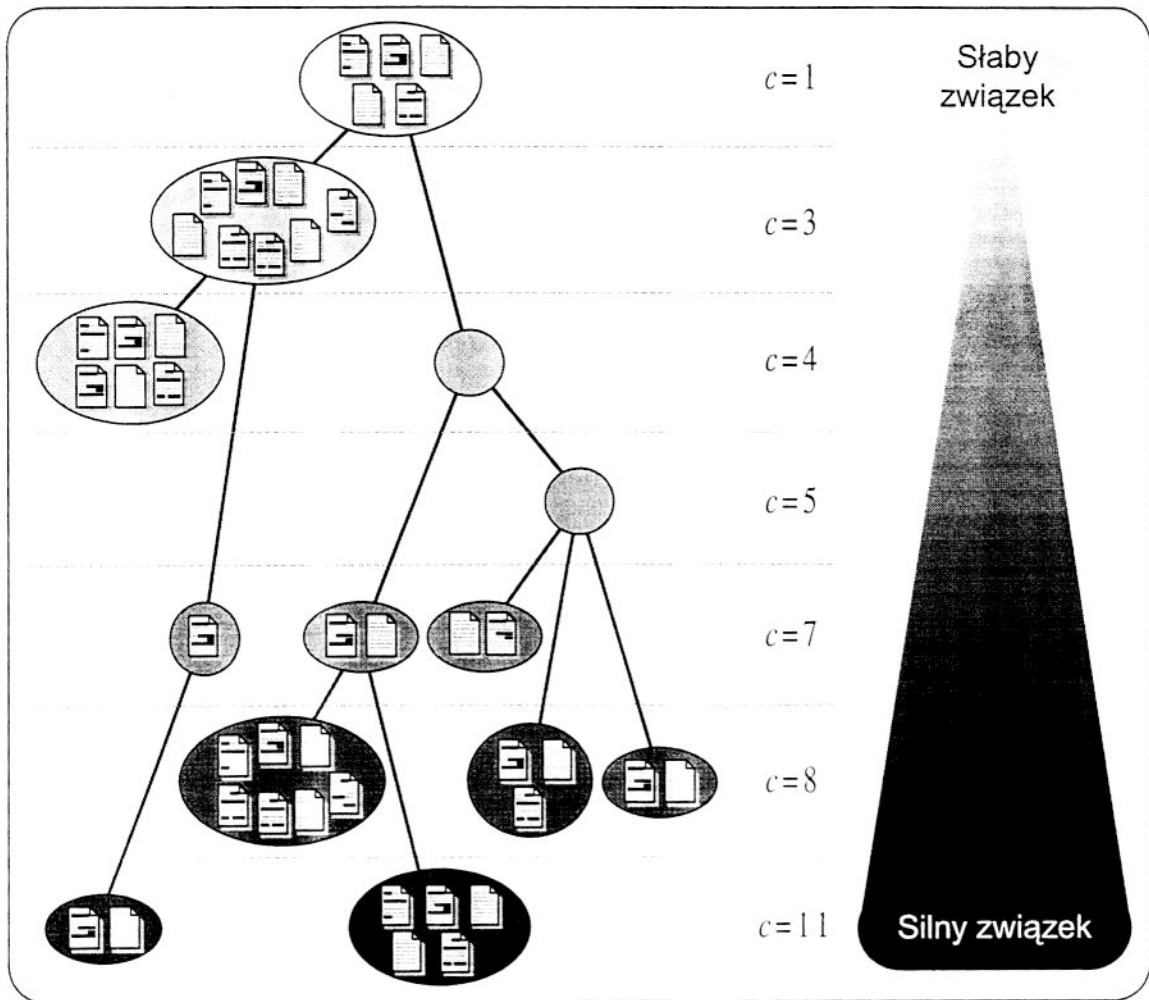
¹ Katalog ten jest dostępny pod adresem: <http://www.yahoo.com>. Składa się on obecnie z 14 głównych kategorii tematycznych. Korzeń hierarchii nie zawiera dokumentów, ale już w niektórych kategoriach pierwszego poziomu (głównych) umieszczono pojedyncze odsyłacze do serwisów WWW, które mieszczą się w ogólnej tematyce danej kategorii. Na niższych poziomach (węższe dziedziny) znajduje się zwykle lista dokumentów (odsyłaczy do serwisów z tej dziedziny) a także odesłania do kategorii szczegółowszych.

kategorię, a dokumenty z kategorii niższych wybrane dziedziny szczegółowe. Cechą zasadniczo różniącą katalog Yahoo z tutaj zdefiniowaną hierarchią jest istnienie połączeń poziomych. W katalogu do danego elementu hierarchii mogą prowadzić odesłania nie tylko od jego bezpośredniego rodzica, ale także od innych elementów, czyli w istocie jest to sieć a nie hierarchia.



Rys. 5.8. Hierarchia zgodna z definicją 5.2. Zbiory Z pokrywają się z tymi, z rys. 5.7.

Hierarchia z def. 5.2 jest inna, niż ta, która jest efektem działania hierarchicznych metod podziału przedstawionych w 3 rozdziale. W tamtych hierarchiach wszystkie elementy zbioru wejściowego są umieszczane w liściach hierarchii. Te elementy, które nie są liśćmi, traktuje się jako niezawierające w ogóle dokumentów lub składające się z sumy dokumentów, pochodzących z wszystkich liści – potomków. W elementach nie będących liśćmi hierarchii (z def. 5.2), która powstaje w metodzie grupowania na podstawie drzewa maksymalnych przepływów, umieszczane są te dokumenty, które „nie mieszczą się” w niższych elementach – ściślejszych kategoriach tematycznych. Przejście z hierarchii w rozumieniu def. 5.2 do hierarchii, w której elementy zawierają dokumenty z wszystkich liści poniżej danego elementu, możliwe jest poprzez grupy rozszerzone opisane w pkt. 5.10 (porównaj także rys. 3.1 z 5.14).



Rys. 5.9. Hierarchia grup dokumentów, odpowiadająca tej z rys. 5.8.

W dalszej części pracy zostaną przedstawione dowody, dla których przeprowadzenia niezbędne jest zdefiniowanie pewnych pojęć związanych z hierarchią.

Definicja 5.3.

Dwie hierarchie grup $(H_1(V^S), \leq_{H_1})$ oraz $(H_2(V^S), \leq_{H_2})$ na zbiorze V^S **sa równoważne** tzn.:

$$(H_1(V^S), \leq_{H_1}) \equiv (H_2(V^S), \leq_{H_2}),$$

wtedy i tylko wtedy, gdy istnieje funkcja $\varphi: H_1(V^S) \rightarrow H_2(V^S)$ taka, że:

1. φ jest wzajemnie jednoznaczna,
2. $(\forall (Z, c, e) \in H_1(V^S)) (\exists e' \in \mathbb{N}) \varphi(Z, c, e) = (Z, c, e')$,
3. $(\forall (Z, c, e), (Z', c', e') \in H_1(V^S)) (Z, c, e) \leq_{H_1} (Z', c', e') \Rightarrow \varphi(Z, c, e) \leq_{H_2} \varphi(Z', c', e')$.

Oznacza to, że dwie hierarchie są równoważne wtedy, gdy różnią się jedynie ponumerowaniem (e) pustych elementów.

Warunek pierwszy definicji jest równoznaczny z tym, że funkcja φ jest różnowartościowa i „na”.

Warunek drugi (w połączeniu z warunkiem 4 definicji 5.2) oznacza, że dla elementów hierarchii (Z, c, e) , które zawierają niepusty zbiór Z , funkcja φ jest identycznością, tzn. $e = e' = 0$.

Ponieważ w warunku trzecim jest kwantyfikator ogólny, więc obie hierarchie $(H_1(V^S), \leq_{H_1})$ oraz $(H_2(V^S), \leq_{H_2})$ wyglądają tak samo, czyli funkcja φ zachowuje strukturę hierarchii - izomorfizm struktur, tzn. odpowiadające sobie elementy znajdują się w tych samych miejscach w obu hierarchiach.

Określmy teraz pojęcie poziomu hierarchii.

Definicja 5.4.

Niech $(H(V^S), \leq_H)$ będzie hierarchią na zbiorze V^S .

Definiujemy indukcyjnie: $K_i \subseteq P(V^S) \times \mathbf{N}$ oraz $c_i \in \mathbf{N}_+ \cup \{\infty\}$, dla $i \in \mathbf{N}$. Niech (Z_0, c_0, e_0) będzie korzeniem hierarchii (def. 5.2, pkt. 6):

1. $K_0 = \{(Z_0, e_0)\}$.
2. Załóżmy, że mamy zdefiniowane K_i oraz c_i , wtedy:

$$c_{i+1} = \min \{c : c > c_i \wedge (\exists Z \in P(V^S)) (\exists e \in \mathbf{N}) (Z, c, e) \in H(V^S)\},$$

$$K_{i+1} = \{(Z, e) \in P(V^S) \times \mathbf{N} : (Z, c_{i+1}, e) \in H(V^S)\}.$$

Jeżeli dla pewnego $j \in \mathbf{N}$ nie istnieje związanie c , takie że $c > c_j$, to przyjmujemy $c_i = \infty$ oraz $K_i = \emptyset$, dla $i \geq j+1$.

Zbiór K_i nazwijmy **i -tym poziomem hierarchii $(H(V^S), \leq_H)$** .

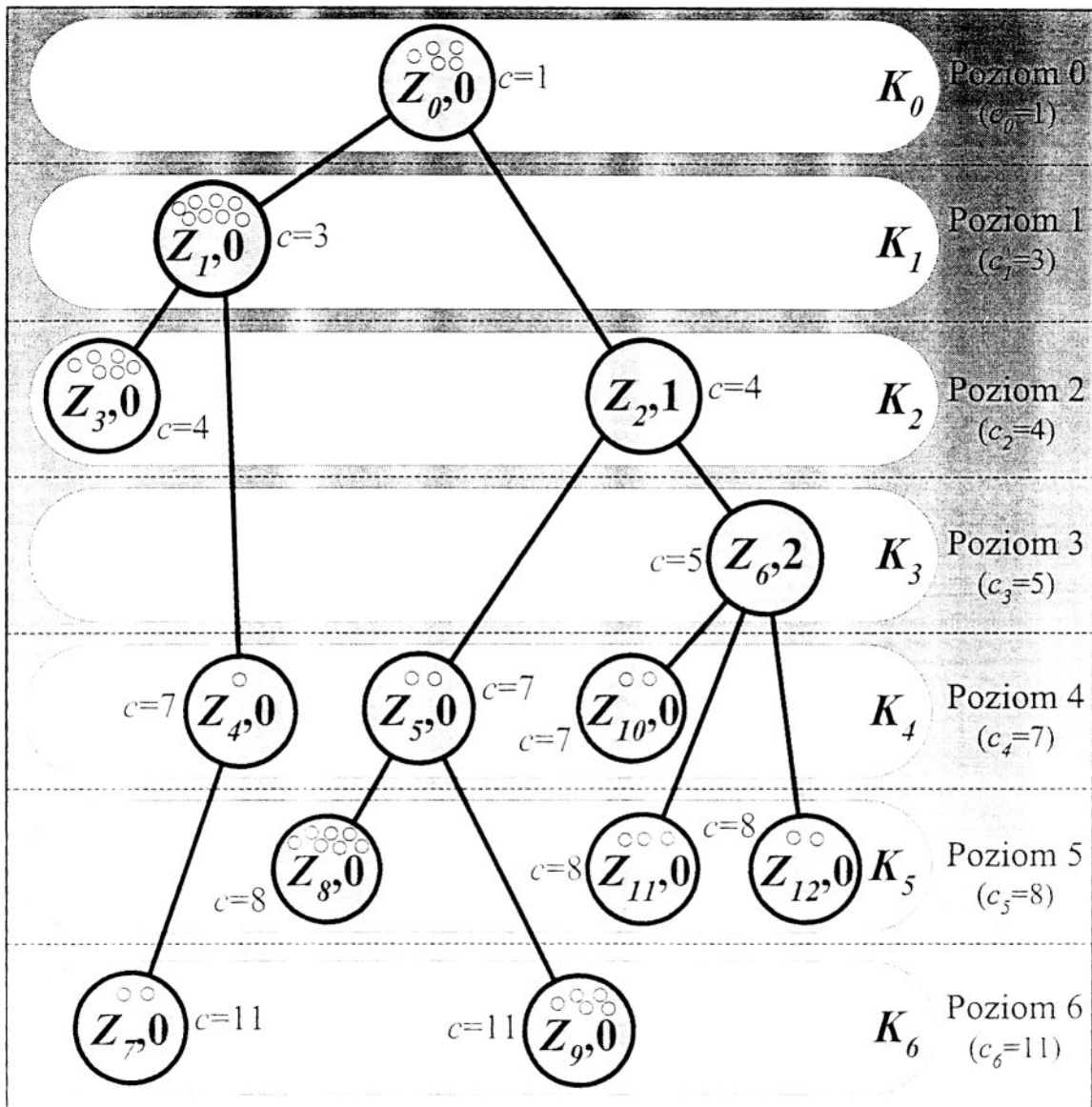
Wszystkie elementy hierarchii $(H(V^S), \leq_H)$ znajdujące się na tym samym i -tym poziomie (K_i) mają taką samą wartość związania c_i . Zbiór K_i składa się z par (Z, e) zawierających zbiory Z znajdujące się na i -tym poziomie.

Liczba wartości związań c_i jest skończona w danej hierarchii, czyli po ponumerowaniu wszystkich wartości związań (indeks i) kolejnym związaniom c_i nadajemy wartość ∞ , natomiast $K_i = \emptyset$. Znak ∞ jest tutaj tylko symbolem oznaczającym, że niższe poziomy hierarchii już nie istnieją.

Rysunek 5.10 zawiera hierarchię z rys. 5.8 podzieloną na poziomy, zgodnie z def. 5.4. Poziom zerowy K_0 składa się z jednego elementu $(Z_0, 0)$; $K_1 = \{(Z_1, 0)\}$; $K_2 = \{(Z_3, 0), (Z_1, 1)\}$; $K_3 = \{(Z_6, 2)\}$; $K_4 = \{(Z_4, 0), (Z_5, 0), (Z_{10}, 0)\}$; itd. Poziomy – zgodnie z definicją 5.4 – zaznaczono także na rys. 5.9.

Im niższy poziom, tym większa jest wartość związania c . Numerami poziomów są kolejne liczby naturalne (łącznie z zerem), tj. 0, 1, 2, 3, 4, ... Wartości związania węzłów c skojarzone z tymi poziomami to także rosnące liczby naturalne (bez zera), gdyż przeskalowane wartości maksymalnego przepływu są liczbami naturalnymi. Jednak wartości związania węzłów nie muszą być wcale kolejnymi liczbami naturalnymi. Dla hierarchii z rys. 5.10 nie istnieją na przykład związania o wartości 2, 6, 9 oraz 10.

Poziomy o większych numerach (określanych przez indeks) będziemy nazywać **niższymi**, a o mniejszych numerach - **wyższymi**. Jest to zgodne z graficznym przedstawianiem hierarchii, (rys. 5.8 i 5.10). Poziomem najwyższym jest poziom zerowy, zawierający korzeń hierarchii.



Rys. 5.10. Hierarchia z rys. 5.8 z zaznaczonymi poziomami.

Definicja 5.5.

Poziomem węzła $v \in V^S$ w hierarchii $H(V^S)$ nazywamy liczbę naturalną (włącznie z zerem) i , taką że $(\exists (Z,e) \in K_i) v \in Z$.

Poziom węzła v oznaczać będziemy przez $p(v)$.

Definicja 5.5 określa, że poziomem węzła jest numer poziomu hierarchii, zawierający element hierarchii, do którego ten węzeł należy.

Definicja 5.6.

Niech $v_1, v_2 \in V^S$, $Z_1, Z_2 \in P(V^S)$ tak, że $v_1 \in Z_1$ i $v_2 \in Z_2$ oraz $(Z_i, c_{p(v_i)}, 0) \in H(V^S)$, wtedy **najniższym wspólnym poziomem dwóch węzłów v_1, v_2** nazywamy największą liczbę naturalną (w tym także może być zero) i , taką że:

$$(\exists (Z, e) \in K_i) [(Z_1, c_{p(v_1)}, 0) \leq_H (Z, c_i, e) \wedge (Z_2, c_{p(v_2)}, 0) \leq_H (Z, c_i, e)].$$

Najniższy, wspólny poziom węzłów v_1, v_2 będziemy oznaczać przez: $p(v_1, v_2)$.

Najniższym poziomem wspólnym dwóch węzłów jest numer najniższego poziomu hierarchii, na którym leży wspólny przodek zarówno elementu, do którego należy węzeł pierwszy jak i elementu, do którego należy węzeł drugi.

W hierarchii z rys. 5.10, dla dwóch węzłów, np. $v_1 \in Z_8$ oraz $v_2 \in Z_{10}$ ich najniższym wspólnym poziomem $p(v_1, v_2)$ jest poziom 2, gdyż na nim leży wspólny przodek zarówno elementu $(Z_8, 8, 0)$ jak i $(Z_{10}, 5, 0)$, czyli element $(Z_2, 4, 1)$. Oczywiście istnieje także inny, wspólny przodek obu elementów tzn. $(Z_0, 1, 0)$, ale leży on na wyższym poziomie (zerowym).

Zawsze istnieje wspólny przodek dla dwóch elementów: korzeń hierarchii, chociaż nie zawsze jest on przodkiem najniżej położonym.

Definicja 5.7.

Zbiorem liści będących potomkami elementu (Z, c, e) należącego do hierarchii $(H(V^S), \leq_H)$ nazywamy zbiór $F(Z, c, e)$ taki, że:

$$F(Z, c, e) = \{ (Z', c', e') \in H(V^S) : (Z', c', e') \leq_H (Z, c, e) \wedge \neg (\exists (Z'', c'', e'') \in H(V^S)) (Z'', c'', e'') <_H (Z', c', e') \}.$$

W hierarchii z rys. 5.8 dla elementu $(Z_4, 7, 0)$, zbiór liści wynosi: $F(Z_4, 7, 0) = \{(Z_7, 11, 0)\}$. natomiast, dla elementu pustego $(Z_2, 4, 1)$ zbiór $F(Z_2, 4, 1) = \{(Z_8, 8, 0), (Z_9, 11, 0), (Z_{10}, 7, 0), (Z_{11}, 8, 0), (Z_{12}, 8, 0)\}$. Dla elementu, który jest już liściem, jego zbiór F składa się tylko z niego samego. Dla korzenia hierarchii, zbiór $F(Z_0, c_0, e_0)$ zawiera wszystkie liście występujące w hierarchii.

5.7. ALGORYTM TC GRUPOWANIA DRZEWA MAKSYMALNYCH PRZEPŁYWÓW

Wygenerowanie hierarchii grup dokumentów (zgodnych z definicją 5.2) jest możliwe za pomocą opracowanego algorytmu TC (*tree clustering*) grupowania drzewa maksymalnych przepływów.

Algorytm ten dla danego drzewa maksymalnych przepływów $T(V^S, E^T)$ generuje dokładnie jedną hierarchię $(H(T), \leq_{H(T)})$ ¹. Ponieważ jedno takie drzewo powstaje z jednej składowej grafu ważonego $G^S(V^S, E^S)$, więc zastosowanie algorytmu TC powoduje powstanie jednej hierarchii dla jednej składowej. Oznacza to, że jeżeli ze zbioru D dokumentów, będącego elementem kolekcji hipertekstowej $HT(D, L)$, w drugim i trzecim etapie metody powstanie kilka składowych, to po zasto-

¹ Symbol $(H(T), \leq_{H(T)})$ oznacza hierarchię na zbiorze V^S , powstałą z drzewa $T(V^S, E^T)$.

sowaniu algorytmu TC do każdej z nich z osobna, wygenerowanych zostanie tyle hierarchii, ile jest składowych.

Algorytm TC grupowania drzewa maksymalnych przepływów

Wejście: drzewo maksymalnych przepływów $T(V^S, E^T)$.

Wyjście: hierarchia elementów (Z, c, e) związana z drzewem - $(H(T), \leq_{H(T)})$.

Krok 1. Zainicjuj pusty element - korzeń hierarchii $(Z, c, e) := (\emptyset, 0, 0)$. Zainicjuj składową drzewa $T^C(V^C, E^C) := T(V^S, E^T)$. Zainicjuj zmienną *parent* pomocniczą wartością *root*. Zainicjuj wartością zero zmienną e_{next} (następny numer elementu pustego).

Krok 2. W składowej $T^C(V^C, E^C)$ znajdź zbiór E_{min} krawędzi o minimalnej wadze c_{min} (waga c_{min} jest minimalną przeskalowaną wartością maksymalnego przepływu występującą w drzewie $T^C(V^C, E^C)$).

Usuń zbiór E_{min} ze zbioru krawędzi składowej, czyli:

$$E^C := E^C \setminus E_{min}.$$

Krok 3. Znajdź zbiór węzłów izolowanych w drzewie $T^C(V^C, E^C)$ i podstaw ten zbiór pod zmienną Z . Jeżeli $Z = \emptyset$, to zwiększ o jeden wartość zmiennej określającej następny numer elementu pustego e_{next} czyli:

$$e_{next} := e_{next} + 1,$$

oraz ustaw wartość aktualnego numeru elementu pustego: $e_c := e_{next}$.

Jeżeli $Z \neq \emptyset$, wtedy nie zmieniaj wartości zmiennej e_{next} oraz wartość aktualnego numeru elementu pustego ustaw na 0, tzn. $e_c := 0$.

Ze zbioru Z , wartości wagi c_{min} oraz aktualnego numeru elementu pustego e_c utwórz element hierarchii (Z, c_{min}, e_c) . Rodzicem dla tego elementu będzie wartość zmiennej *parent*.

Usuń zbiór węzłów izolowanych Z ze składowej $T^C(V^C, E^C)$ czyli:

$$V^C := V^C \setminus Z.$$

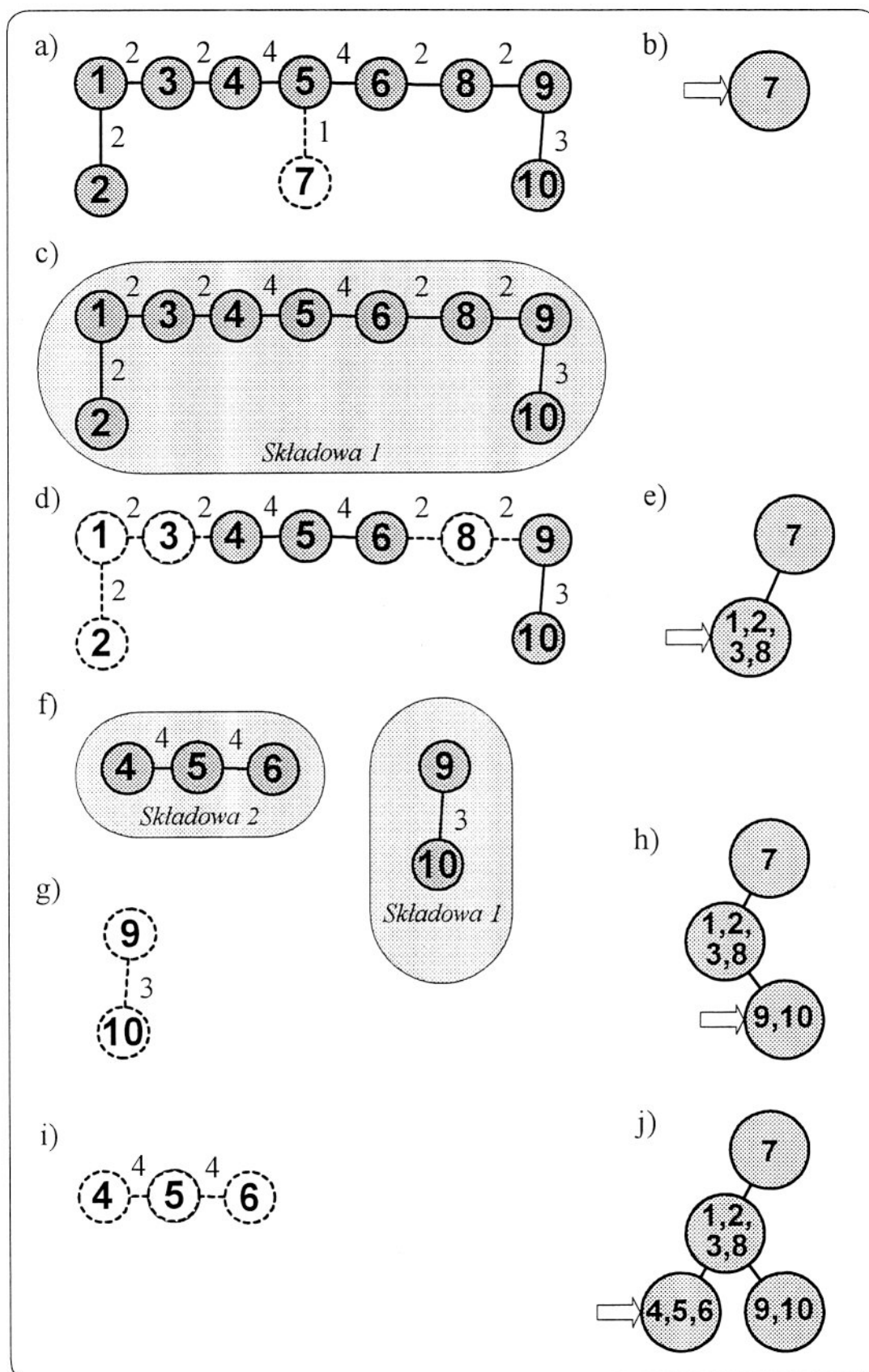
Krok 4. Wydzielony element (Z, c_{min}, e_c) wyprowadź na wyjście, tzn. dodaj do hierarchii $(H(T), \leq_{H(T)})$ poprzez dodanie elementu (Z, c_{min}, e_c) do zbioru elementów hierarchii $H(T)$ oraz – jeżeli $parent \neq root$ – rozszerzenie $\leq_{H(T)}$ w następujący sposób:

1. $(Z, c_{min}, e_c) \leq_{H(T)} (Z, c_{min}, e_c)$; zapewnienie zwrotności relacji $\leq_{H(T)}$.
2. $(Z, c_{min}, e_c) <_{H(T)} parent$; relacja z bezpośrednim przodkiem – rodzicem.
3. $(Z, c_{min}, e_c) <_{H(T)} przodek$, dla wszystkich elementów $przodek \in H(T)$ znajdujących się aktualnie w relacji: $parent <_{H(T)} przodek$; dodanie relacji z pozostałymi przodkami, oprócz rodzica.

Krok 5. Wyodrębnij wszystkie składowe S_i w grafie $T^C(V^C, E^C)$. Dla każdej wyodrębnionej składowej S_i osobno wykonaj kroki 2, 3, 4, 5, podstawiając $T^C(V^C, E^C) := S_i$ oraz $parent := (Z, c_{min}, e_c)$. Jeżeli nie istnieją składowe w grafie $T^C(V^C, E^C)$, zakończ algorytm.

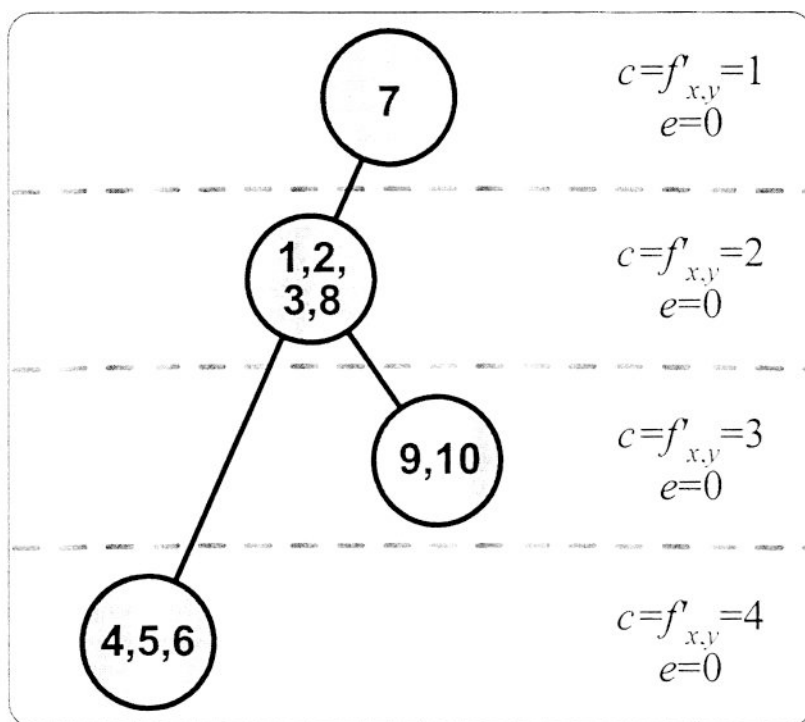
Działanie algorytmu TC jest zstępujące: od góry hierarchii do dołu, tzn. najpierw na wyjście (krok 4) przekazywany jest korzeń hierarchii, następnie jego dzieci a potem kolejne pokolenia.

Numeracja pustych elementów hierarchii (zmienna e_{next}) zaczyna się od 1, gdyż najpierw następuje inkrementacja a potem wykorzystanie wartości tej zmiennej.



Rys. 5.11. Przebieg działania algorytmu TC dla drzewa rys. 5.5a.

Prześledźmy działanie algorytmu TC dla drzewa z rys. 5.5a. Kolejność wydzielania grup i tworzenia hierarchii pokazano na rys. 5.11. W kroku 2 wydzielana jest jedna krawędź o wadze 1 (rys. a), narysowana linią przerywaną. W kroku 3 z węzła 7 (linia przerywana), który stał się izolowany, tworzony jest korzeń hierarchii: $(\{7\}, 1, 0)$, który w 4 kroku wyprowadzany jest na wyjście (rys. b). Pozostałe drzewo $T^C(V^C, E^C)$ – krok 4 – składa się z jednej składowej (rys. c), dla której wywoływane są kroki 2-4. W ponownie wykonywanym kroku 2 krawędzie mają minimalną wagę $c_{min}=2$ (rys. d). W kroku 3 z węzłów izolowanych: 1, 2, 3, 8 tworzony jest drugi element hierarchii (dziecko korzenia): $(\{1, 2, 3, 8\}, 2, 0)$, (rys. e). W pozostałym po kroku 3 grafie wydzielane są dwie składowe (rys. f). Następuje tutaj „rozgałęzienie” hierarchii. Kolejność przetwarzania składowych zależy wyłącznie od sposobu implementacji algorytmu i nie ma wpływu na postać hierarchii. Z jednej z tych wydzielonych składowych, po usunięciu jedynej krawędzi o wadze 3 (rys. g), tworzony jest kolejny element hierarchii: $(\{9, 10\}, 3, 0)$ – dziecko drugiego elementu (rys. h). Z drugiej składowej, po usunięciu wszystkich krawędzi o wadze 4 (rys. i), powstaje element hierarchii $(\{4, 5, 6\}, 4, 0)$, (rys. j). Utworzona przez algorytm TC hierarchia, z podziałem na poziomy o tych samych wartościach związania c , została przedstawiona na rys. 5.12.

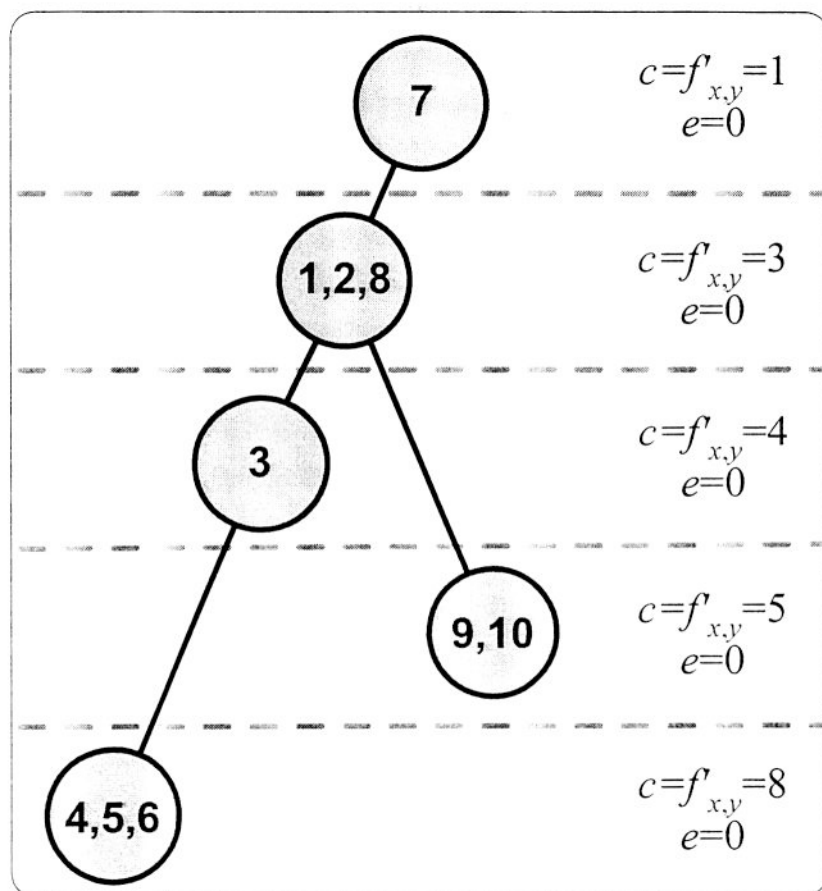


Rys. 5.12. Hierarchia, utworzona za pomocą algorytmu TC, dla drzew z rys. 5.5.

Drzewo, dla którego prezentowano przebieg działania algorytmu (rys. 5.5a), powstało dla składowej grafu ważonego z rys. 1.1, dla $max^{lev}=4$. Dla tej samej składowej, na rys. 5.5b przedstawiono inne drzewo maksymalnych przepływów z przeskalowanymi wartościami (także dla $max^{lev}=4$). Działanie algorytmu TC, dla drzewa z rys. 5.5b, przebiegałoby dokładnie tak samo, jak to pokazano wyżej. Oba drzewa różnią się postaciami, chociaż pochodzą z tej samej składowej. Dla każdego z tych drzew, po zastosowaniu algorytmu TC, wygenerowana została taka sama hierarchia.

Niezależność hierarchii tworzonej przez algorytm TC od postaci drzewa maksymalnych przepływów, jako własność algorytmu TC, zostanie udowodniona w dalszej części pracy (twierdzenie 5.13).

W przypadku, gdy do drzew maksymalnych przepływów z rys. 1.3 zastosujemy inną wartość parametru $max^{lev}=8$ podczas procesu skalowania, wtedy otrzymamy drzewa przedstawione na rys. 5.6. Dla każdego z tych drzew algorytm TC utworzy taką samą hierarchię przedstawioną na rys. 5.13.



Rys. 5.13. Hierarchia utworzona za pomocą algorytmu TC dla drzew z rys. 5.6.

Ponieważ w kroku 5, przy wyodrębnianiu składowych, następuje „rozgałęzienie”, więc algorytm TC najłatwiej zaimplementować w postaci programu rekurencyjnego.

Definicja 5.8.

Hierarchia grup związana z drzewem $T(V^S, E^T)$ nazywamy hierarchią $(H(T), \leq_{H(T)})$ na zbiorze V^S powstałą z drzewa $T(V^S, E^T)$ poprzez zastosowanie algorytmu TC grupowania drzewa maksymalnych przepływów.

Definicja 5.8 służy wprowadzeniu, niezbędnego dalej, oznaczenia $(H(T), \leq_{H(T)})$, dla określenia hierarchii, powstałej z drzewa maksymalnych przepływów $T(V^S, E^T)$.

5.8. WŁASNOŚCI ALGORYTMU TC

W pierwszej kolejności zostanie wykazane, że produktem algorytmu TC jest hierarchia grup zgodna z definicją 5.2 (twierdzenie 5.1). Następnie zostanie udowodnione to, że postać hierarchii grup utworzonej przez algorytm TC nie zależy od postaci drzewa maksymalnych przepływów (twierdzenie 5.13).

5.8.1. ZGODNOŚĆ Z DEFINICJĄ HIERARCHII

Twierdzenie 5.1.

Hierarchia utworzona z drzewa maksymalnych przepływów $T(V^S, E^T)$, za pomocą algorytmu TC, jest zgodna z definicją 5.2, hierarchii na zbiorze V^S .

Dowód:

Z opisu działania algorytmu TC, w części związanej z wyprowadzaniem elementów hierarchii na wyjście, wszystkie cechy relacji porządkującej (wzory: 1.1, 1.2, 1.3) są spełnione gdyż:

1. Zwrotność (1.1) wynika wprost z punktu 1 wyprowadzania na wyjście algorytmu (krok 4).
2. Słaba antysymetria (1.2). Gdyby nie była spełniona, to by oznaczało, że istnieją dwa elementy w hierarchii, z których każdy jest przodkiem (nie necessarily rodzicem) drugiego, co na mocy punktu 2 oraz 3 wyprowadzania na wyjście (krok 4 algorytmu TC) jest niemożliwe, gdyż przodkiem jest zawsze ten element hierarchii, który pojawił się już wcześniej na wyjściu algorytmu i został w kroku 5 podstawiony pod zmienną *parent*.
3. Przechodność (1.3) wynika z punktu 2 i 3 wyprowadzania na wyjście algorytmu TC.

Dalej zostanie wykazana zgodność produktu algorytmu TC z poszczególnymi warunkami wymienionymi w def. 5.2:

1. Spełnienie warunku 1 def. 5.2 jest oczywiste (krok 3 i 4 algorytmu TC).
2. Warunek 2 jest spełniony w kroku 3, gdyż te węzły, które są wyprowadzane na wyjście algorytmu w postaci zbioru Z , są usuwane ze zbioru V^C .
3. Warunek 3 jest prawdziwy, ponieważ algorytm jest wykonywany dopóty, dopóki istnieją składowe (aby istniała składowa muszą istnieć węzły), czyli musi wyczerpać wszystkie węzły (krok 5). Usuwane są tylko te węzły, które wyprowadzane są w postaci zbioru Z na wyjście algorytmu.
4. Warunek 4 jest spełniony w kroku 3, ponieważ zmiennej e_c nadawana jest wartość zero.
5. W kroku 3 nie może powtórzyć się numeracja elementu pustego hierarchii, gdyż dla każdego takiego wyprowadzanego elementu zmienna e_{next} jest inkrementowana, a więc spełniony jest warunek 5.
6. Spełnienie warunku 6 następuje w pierwszym przebiegu algorytmu, gdy wydzielany jest element (Z_0, c_0, e_0) , który staje się pierwszym rodzicem – korzeniem hierarchii (podstawienie pod zmienną *parent* w kroku 5). Wszystkie elementy, wyprowadzane na wyjście w kroku 4, w kolejnych przebiegach, mają przodka w postaci elementu wydzielonego w pierwszym przebiegu, czyli (Z_0, c_0, e_0) .
7. W kroku 2 algorytmu wydzielane są krawędzie minimalne (E_{min}). Jeżeli pewien element (Z', c', e') jest rodzicem, a tym bardziej przodkiem elementu (Z, c, e) , to (Z', c', e') musiał być wyprowadzony na wyjście wcześniej (musiał być wcześniej w kroku 5 podstawiony pod zmienną *parent*), czyli wartość jego związania c' była mniejsza, gdyż mniejsza była minimalna waga w_{ij} ustalona w kroku 2, podczas wydzielania (Z', c', e') . Warunek 7 jest zatem spełniony.

8. W kroku 4 na wyjście wyprowadzany jest element z jednym rodzicem i rodzic ten jest na wyższym poziomie niż element dodawany (punkt 2 wyprowadzania). Rodzic ów także musiał mieć tylko jednego swojego rodzica, ale na jeszcze wyższym poziomie itd. W związku z tym nie możliwe jest dodanie w pkt. 3 wyprowadzenia relacji z innym przodkiem, który byłby niżej położony niż rodzic dodany w punkcie 2 wyprowadzenia na wyjście. Zatem warunek 8 jest również spełniony.

□

5.8.2. NIEZALEŻNOŚĆ HIERARCHII OD DRZEWA MAKSYMALNYCH PRZEPŁYWÓW

Dalej przedstawiony zostanie dowód niezależności hierarchii grup powstałej po zastosowaniu algorytmu TC od postaci drzewa maksymalnych przepływów. Dla danej składowej grafu ważonego może istnieć wiele różnych drzew maksymalnego przepływu, jednak za pomocą algorytmu TC można z nich utworzyć tylko jedną hierarchię grup.

Dla przeprowadzenia dowodu twierdzenia o niezależności hierarchii od postaci drzewa maksymalnych przepływów niezbędne jest zdefiniowanie dodatkowych pojęć oraz udowodnienie pewnych własności hierarchii.

Lemat 5.2.

Jeżeli $T_1(V_1^S, E_1^T)$ oraz $T_2(V_2^S, E_2^T)$ – dwa dowolne drzewa maksymalnych przepływów dla tej samej składowej grafu ważonego $G^S(V^S, E^S)$, to niech $(H(T_1), \leq_{H(T_1)})$ oraz $(H(T_2), \leq_{H(T_2)})$ będą hierarchiami grup, związanymi z drzewem odpowiednio $T_1(V_1^S, E_1^T)$ i $T_2(V_2^S, E_2^T)$, wtedy:

$$\{c_i^{T_1} : i \in \mathbf{N} \wedge c_i^{T_1} \neq \infty\} = \{c_i^{T_2} : i \in \mathbf{N} \wedge c_i^{T_2} \neq \infty\},$$

przy czym indeks górny T_1 oraz T_2 jest równoznaczny temu, że dane oznaczenie dotyczy hierarchii dla odpowiednio drzewa maksymalnych przepływów T_1 lub T_2 .

Indeks górny T_1 oraz T_2 jest równoznaczny temu, że dane oznaczenie dotyczy hierarchii powstałej z odpowiednio drzewa maksymalnych przepływów $T_1(V_1^S, E_1^T)$ oraz $T_2(V_2^S, E_2^T)$ i dotyczyć to będzie całej dalszej części pracy.

Dowód:

Z algorytmu TC (kroki 2 i 3) wynika, że w ciągu $\{c_i\}_{i \in \mathbf{N}}$ występują dokładnie takie same liczby naturalne, dodatnie, które są wagami krawędzi w odpowiadającym, przeskalowanemu drzewie maksymalnych przepływów.

Założmy, nie wprost, że w ciągu $\{c_i^{T_1}\}_{i \in \mathbf{N}}$ występuje wartość związania nie występująca w ciągu $\{c_i^{T_2}\}_{i \in \mathbf{N}}$. Oznaczmy ją $c_{i_0}^{T_1}$. Z twierdzenia 1.3 wynika, że w drzewie T_1 istnieje krawędź o wadze $c_{i_0}^{T_1}$, łącząca węzły v_1 oraz v_2 . Wartość maksymalnego, przeskalowanego przepływu między tymi węzłami wynosi zatem $c_{i_0}^{T_1}$ w drzewie T_1 i tyle samo w drzewie T_2 , gdyż oba drzewa pochodzą od tej samej składowej grafu ważonego. Stąd wynika jednak, że w drzewie T_2 istnieje krawędź o wadze $c_{i_0}^{T_1}$, po usunięciu której węzły v_1 oraz v_2 będą w różnych składowych, co jest wbrew założeniu. Otrzymana sprzeczność kończy dowód.

□

Zbiór wartości związań węzłów jest taki sam we wszystkich drzewach maksymalnych przepływów, powstałych z tej samej składowej grafu ważonego.

Prawdziwy jest nawet dalej idący wniosek:

Wniosek 5.3:

$$(\forall i \in \mathbf{N}) \quad c_i^{T_1} = c_i^{T_2}.$$

Dowód:

Prawdziwość wniosku wynika bezpośrednio z lematu 5.2 oraz faktu, że ciągi $\{c_i^{T_1}\}_{i \in \mathbf{N}}$, $\{c_i^{T_2}\}_{i \in \mathbf{N}}$ są rosnące¹, tzn. $c_0^{T_1} < c_1^{T_1} < c_2^{T_1} < \dots$

□

Wniosek 5.3 pozwala przestać rozróżniać pomiędzy drzewami T_1 oraz T_2 i zamiast $c_i^{T_1}$ i $c_i^{T_2}$ pisać po prostu c_i .

Lemat 5.4.

Niech $(H(T_1), \leq_{H(T_1)})$ oraz $(H(T_2), \leq_{H(T_2)})$ będą hierarchiami jak w lemacie 5.2, wtedy:

$$(\forall v \in V^S) \quad p^{T_1}(v) = p^{T_2}(v).$$

Dowód:

Założmy, nie wprost, że istnieje taki węzeł $v_j \in V^S$, że $p^{T_1}(v) = i_1 < p^{T_2}(v) = i_2$. Wtedy z twierdzenia 1.3 w drzewie T_1 węzeł v_j jest izolowany po usunięciu pewnej krawędzi o wadze c_{i_1} (krok 3 algorytmu TC), natomiast w drzewie T_2 – nie. W T_2 (więc także w składowej grafu ważonego $G^S(V^S, E^S)^2$) istnieje taki węzeł v_k , że $f_{j,k} > c_{i_1}$, czyli sprzeczność.

□

Z lematu 5.4 wynika, że każdy węzeł znajduje się na odpowiednim, tym samym poziomie hierarchii (def. 5.5); bez względu na to, z jakiego drzewa maksymalnych przepływów ta hierarchia powstała. Zatem, zamiast $p^{T_1}(v)$ i $p^{T_2}(v)$ można pisać $p(v)$.

Wniosek 5.5:

$$(\forall i \in \mathbf{N}) \quad \bigcup L_i^{T_1} = \bigcup L_i^{T_2}.$$

Dowód:

¹ chodzi o to, że są „ściśle rosnące”, tzn. nie możliwa jest równość $c_i^{T_1} = c_{i+1}^{T_1}$ dla żadnego $i \in \mathbf{N}$

² wartość maksymalnego przepływu w składowej należy oczywiście przeskalować

Prawdziwość wniosku wynika bezpośrednio z lematu 5.4, ponieważ przynależność węzła do poziomu hierarchii jest jednoznacznie wyznaczona przez poziom tego węzła w hierarchii.

□

Udowodniono więc, że poszczególne poziomy w obu drzewach T_1 oraz T_2 składają się z tych samych węzłów ze zbioru V^S .

Lemat 5.6.

Dla dwóch węzłów v_1, v_2 , należących do jednej hierarchii, zachodzą równości:

$$p(v_1, v_2) \leq p(v_1) \text{ oraz } p(v_1, v_2) \leq p(v_2).$$

Dowód:

Prawdziwość lematu 5.6 wynika wprost z definicji 5.6.

□

Najniższy, wspólny poziom dwóch węzłów (def. 5.6) nie może być niższy niż poziom każdego z tych węzłów z osobna.

Lemat 5.7.

Dla dwóch węzłów v_1, v_2 należących do jednej hierarchii zachodzi równość:

$$f'_{1,2} = c_{p(v_1, v_2)}.$$

gdzie: $c_{p(v_1, v_2)}$ - wartość związania skojarzona z najniższym, wspólnym poziomem $p(v_1, v_2)$ węzłów v_1 oraz v_2 .

Dowód:

Z wniosku 1.4 wynika, że $f'_{1,2}$ jest równe minimalnej wadze w_{ij}^T krawędzi, po usunięciu której v_1 oraz v_2 znajdują się w różnych składowych drzewa maksymalnych przepływów.

Jeżeli dwa węzły v_1 oraz v_2 , należące do dwóch, różnych elementów hierarchii, odpowiednio $(Z_1, c_{p(v_1)}, 0)$ i $(Z_2, c_{p(v_2)}, 0)$ i elementy te mają wspólnego przodka (Z, c, e) , to w drzewie maksymalnych przepływów istnieje ścieżka między v_1 oraz v_2 , w której wagi wszystkich krawędzi są większe lub równe c . Wynika to z tego, że w algorytmie TC, tuż przed wydzieleniem (Z, c, e) , węzły v_1 oraz v_2 musiały być w jednej składowej. W składowej tej wagi wszystkich krawędzi były większe lub równe c , gdyż c było minimalne (krok 2 algorytmu).

Ponadto, jeżeli (Z, c, e) jest najniższym przodkiem elementu $(Z_1, c_{p(v_1)}, 0)$ i $(Z_2, c_{p(v_2)}, 0)$, (tzn. że nie mają one wspólnego przodka na niższym poziomie), to w ścieżce w drzewie maksymalnych przepływów, między v_1 oraz v_2 , istnieje krawędź o przeskalowanej wadze dokładnie równej c . Jest tak dlatego, gdyż z algorytmu wynika, że po usunięciu wszystkich krawędzi o wadze c , przestała istnieć ścieżka pomiędzy v_1 i v_2 .

Stąd i z definicji 5.6 wynika, że $f'_{1,2} = w_{ij}^T = c = c_{p(v_1, v_2)}$.

□

Wartość związania (skojarzona z najniższym, wspólnym poziomem dwóch węzłów) jest równa przeskalowanej wartości maksymalnego przepływu między tymi węzłami w dowolnym drzewie maksymalnych przepływów, pod warunkiem, że pochodzi ono od tej samej składowej grafu ważonego.

Wniosek 5.8:

Niech $(H(T_1), \leq_{H(T_1)})$ i $(H(T_2), \leq_{H(T_2)})$ będą hierarchiami. Wtedy:

$$(\forall v_1, v_2 \in V^S) \quad p^{T_1}(v_1, v_2) = p^{T_2}(v_1, v_2).$$

Dowód:

Mamy $c_{p^{T_1}(v_1, v_2)} = f'_{1,2} = c_{p^{T_2}(v_1, v_2)}$, czyli $p^{T_1}(v_1, v_2) = p^{T_2}(v_1, v_2)$.

□

Na podstawie wniosku 5.8 można stwierdzić, że najniższy, wspólny poziom dwóch, dowolnych węzłów w hierarchii utworzonej za pomocą algorytmu TC, nie zależy od drzewa maksymalnych przepływów przekazanego na wejście tego algorytmu (dla tej samej składowej grafu ważonego). W związku z tym można przestać rozróżniać między drzewami T_1 oraz T_2 i zamiast $p^{T_1}(v_1, v_2)$ i $p^{T_2}(v_1, v_2)$ wystarczy pisać $p(v_1, v_2)$.

Można poczynić w tym miejscu uwagę wyjaśniającą szerszy sens związania c . Jak wcześniej wspomniano, c określa wartość przeskalowanego maksymalnego przepływu między dwoma dowolnymi węzłami należącymi do zbioru Z elementu (Z, c, e) hierarchii. Ale to nie wszystko. Związanie c wyznacza także wartość przeskalowanego maksymalnego przepływu między dwoma węzłami należącymi do potomków elementu (Z, c, e) , przy czym muszą to być potomkowie z różnych odgałęzień. Przykładowo: dla pustego elementu $(Z_2, 4, 1)$ z rys. 5.8, wartość $c=4$ oznacza, że dla $v_1 \in Z_8$ i $v_2 \in Z_{10}$ wartość $f'_{1,2}=4$. Dla samego elementu $(Z_2, 4, 1)$ sens związania c , oznaczający wartość maksymalnego, przeskalowanego przepływu między węzłami należącymi do elementu, nie ma nawet zastosowania, gdyż Z_2 nie posiada żadnych węzłów.

Lemat 5.9.

Niech $(H(T_1), \leq_{H(T_1)})$ oraz $(H(T_2), \leq_{H(T_2)})$ będą hierarchiami j.w., wtedy:

$$(\forall i \in \mathbf{N}) \{ (Z, 0): (Z, c_i, 0) \in H(T_1) \} = \{ (Z, 0): (Z, c_i, 0) \in H(T_2) \}.$$

Dowód:

Żałujemy, nie wprost, że $\{ (Z, 0): (z, c_i, 0) \in H(T_1) \} \neq \{ (Z, 0): (z, c_i, 0) \in H(T_2) \}$. Na mocy wniosku 5.5, bez straty ogólności, oznacza to, że:

$$(\exists v_1, v_2 \in V^S) [p(v_1) = p(v_2) = i \wedge (\exists (Z, 0) \in K_i^{T_1})(v_1, v_2 \in Z) \wedge \neg(\exists (Z, 0) \in K_i^{T_2})(v_1, v_2 \in Z)].$$

Wtedy z warunku $(\exists (Z, 0) \in K_i^{T_1})(v_1, v_2 \in Z)$ oraz definicji 5.6 wnioskujemy, że $p(v_1, v_2) = i$. Natomiast z warunków $p(v_1) = p(v_2) = i$ oraz $\neg(\exists (Z, 0) \in K_i^{T_2})(v_1, v_2 \in Z)$ wynika, że w $H(T_2)$ węzły v_1 oraz v_2 należą do różnych grup leżących na i -tym poziomie, co oznacza, że $p(v_1, v_2) < i$. Sprzeczność.

□

Z lematu 5.9 wynika, że jeżeli dwie hierarchie powstały z tej samej składowej grafu ważonego, to nawet jeżeli drzewa maksymalnych przepływów, związane z tymi hierarchiami są różne, wtedy i tak obie hierarchie zawierają ten sam zbiór niepustych elementów. Skład elementów niepustych nie zależy więc od drzewa maksymalnych przepływów.

Twierdzenie 5.10.

Dla hierarchii $(H(V^S), \leq_H)$ wygenerowanej przez algorytm TC mamy, że jeżeli element (Z', c', e') należy do zbioru liści, będących potomkami pewnego elementu $(Z, c, e) \in H(V^S)$ – def. 5.7, to:

$$Z' \neq \emptyset, \text{ a nawet } \text{card}(Z') \geq 2.$$

Dowód:

Zbiór Z' powstaje po usunięciu ze składowej $T^C(V^C, E^C)$ ostatniej (lub ostatnich) krawędzi (krok 3 algorytmu TC). Muszą one łączyć przynajmniej dwa węzły.

□

Każdy element hierarchii wygenerowanej przez algorytm TC, będący liściem, posiada co najmniej dwa węzły.

Lemat 5.11.

Dla hierarchii $(H(V^S), \leq_H)$ wygenerowanej przez algorytm TC mamy, że jeżeli $(\emptyset, c, e) \in H(V^S)$ to $\text{card}(F(\emptyset, c, e)) \geq 2$.

Dowód:

Jeżeli na wyjściu algorytmu TC jest wyprowadzana trójka $(\emptyset, c, e)$ oznacza to, że po usunięciu wszystkich krawędzi o wadze c wejściowa składowa drzewa rozpadła się¹ na dwie lub więcej „podskładowych”, z których żadna nie jest węzłem izolowanym (gdyż $Z = \emptyset$). Do każdej z tych „podskładowych” z osobna ponownie stosujemy algorytm TC i w efekcie (na mocy twierdzenia 5.10) od każdej z nich pochodzi co najmniej jedna trójka $(Z, c', 0)$, należąca do $F(\emptyset, c, e)$, w której $Z \neq \emptyset$.

□

Jeżeli w hierarchii istnieje element pusty, wtedy ten element posiada wśród swoich potomków co najmniej dwa liście, które na mocy twierdzenia 5.10 mają po minimum dwa węzły.

¹ z twierdzenia 1.1 pkt. 5 wynika, że usunięcie dowolnej, jednej krawędzi powoduje rozpad drzewa na 2 składowe

Lemat 5.12.

Niech $(H(T_1), \leq_{H(T_1)})$ i $(H(T_2), \leq_{H(T_2)})$ będą hierarchiami pochodzącymi od drzewa odpowiednio T_1 oraz T_2 . Jeżeli w zbiorze elementów hierarchii $H(T_1)$ istnieje pusty element $(\emptyset, c_i, e)$, to istnieje także $e' \in \mathbf{N}$ takie, że:

$$(\emptyset, c_i, e') \in H(T_2) \text{ oraz}$$

$$F^{T_1}(\emptyset, c_i, e) = F^{T_2}(\emptyset, c_i, e').$$

Dowód:

1. Ustalmy $(Z, c, 0) \in F^{T_1}(\emptyset, c_i, e)$. Z dowodu lematu 5.11 wynika, że istnieje takie $(\hat{Z}, \hat{c}, 0) \in F^{T_1}(\emptyset, c_i, e)$, że jeżeli $v_1 \in Z$ i $v_2 \in \hat{Z}$, to $p(v_1, v_2) = i$. Wtedy z lematu 5.9 wiemy, że $(Z, c, 0), (\hat{Z}, \hat{c}, 0) \in H(T_2)$. Ponieważ $p(v_1, v_2) = i$, więc $\exists (Z'', c_i, e') \in H(T_2)$, takie że $(Z, c, 0) \leq_{H(T_2)} (Z'', c_i, e')$ oraz $(\hat{Z}, \hat{c}, 0) \leq_{H(T_2)} (Z'', c_i, e')$.

Udowodnimy zostanie teraz, że każdy liść z $F^{T_1}(\emptyset, c_i, e)$ jest potomkiem elementu (Z'', c_i, e') w hierarchii $(H(T_2), \leq_{H(T_2)})$. Weźmy dowolne $(Z', c', 0) \in F^{T_1}(\emptyset, c_i, e)$. Niech $v_3 \in Z'$. Jeżeli $p(v_1, v_3) = i$, to oznacza, że w hierarchii $(H(T_2), \leq_{H(T_2)})$ elementy $(Z, c, 0)$ i $(Z', c', 0)$ mają wspólnego przodka na poziomie i . Ponieważ $(Z, c, 0)$ może mieć tylko jednego przodka na poziomie i , to musi nim być (Z'', c_i, e') . Jeżeli $p(v_1, v_3) > i$, to elementy $(Z, c, 0)$ i $(Z', c', 0)$ mają wspólnego przodka poniżej poziomu i . Z definicji hierarchii (5.2, pkt 8) wynika, że ów wspólny przodek musi być potomkiem elementu (Z'', c_i, e') . W obu przypadkach element (Z'', c_i, e') jest przodkiem elementu $(Z', c', 0)$, co kończy rozumowanie.

2. Gdyby $Z'' \neq \emptyset$, to weźmy $v_4 \in Z''$. Wtedy w hierarchii $(H(T_2), \leq_{H(T_2)})$ mamy $p(v_1, v_4) = p(v_4) = i$ (bo v_4 należy do przodka elementu, w którym jest v_1). Natomiast w hierarchii $(H(T_1), \leq_{H(T_1)})$ mamy, że $(Z, c, 0) \leq_{H(T_1)} (\emptyset, c_i, e)$ oraz $(\emptyset, e)$ i (Z'', e') leżą na tym samym i -tym poziomie, czyli $p(v_1, v_4) < i$, sprzeczność. Zatem $Z'' = \emptyset$.
3. Wykażemy teraz, że $(Z, c, 0) \in F^{T_2}(\emptyset, c_i, e')$, czyli jest liściem także w hierarchii $(H(T_2), \leq_{H(T_2)})$.

Założmy, że istnieje taki element $(Z^*, c^*, e^*) \in H(T_2)$, że $(Z^*, c^*, e^*) <_{H(T_2)} (Z, c, 0)$. Bez straty ogólności można przyjąć, że $Z^* \neq \emptyset$.

Niech $v_5 \in Z^*$. Wtedy w hierarchii $(H(T_2), \leq_{H(T_2)})$ mamy, że $p(v_1, v_5) = p(v_1)$, natomiast w hierarchii $(H(T_1), \leq_{H(T_1)})$ mamy, że $p(v_1, v_5) < p(v_1)$. To ostatnie wynika z tego, że $(Z, c, 0)$ jest liściem w hierarchii $(H(T_1), \leq_{H(T_1)})$, czyli wspólny przodek $(Z, c, 0)$ oraz $(Z^*, c^*, 0)$ musi leżeć wyżej niż $(Z, c, 0)$.

4. Pokazano zatem, że $F^{T_1}(\emptyset, c_i, e) \subseteq F^{T_2}(\emptyset, c_i, e')$, dla pewnego $e' \in \mathbf{N}$. Ponieważ sytuacja jest symetryczna, więc te same argumenty wskazują, że $F^{T_2}(\emptyset, c_i, e') \subseteq F^{T_1}(\emptyset, c_i, e'')$, dla pewnego $e'' \in \mathbf{N}_+$. Oznacza to, że $F^{T_1}(\emptyset, c_i, e) \subseteq F^{T_1}(\emptyset, c_i, e'')$. Łatwo zauważyć, że oba elementy $(\emptyset, c_i, e)$ i $(\emptyset, c_i, e'')$ leżą

na tym samym poziomie, więc muszą być one w istocie tym samym elementem (liście jednego zawierają się w liściach drugiego), czyli $e=e'$ ", co kończy dowód.

□

Jeżeli w dwóch hierarchiach powstałych z dwóch różnych drzew maksymalnych przepływów, ale dla tej samej składowej grafu ważonego $G^S(V^S, E^S)$, istnieją elementy puste, to dla każdego takiego elementu pustego z jednej hierarchii istnieje odpowiadający mu pusty element w drugiej hierarchii, posiadający ten sam zbiór potomków – liści.

Twierdzenie 5.13.

Jeżeli $T_1(V_1^S, E_1^T)$ oraz $T_2(V_2^S, E_2^T)$ – dwa dowolne drzewa maksymalnych przepływów dla tej samej składowej grafu ważonego $G^S(V^S, E^S)$, to niech $(H(T_1), \leq_{H(T_1)})$ oraz $(H(T_2), \leq_{H(T_2)})$ będą hierarchiami grup, związanymi z drzewem odpowiednio $T_1(V_1^S, E_1^T)$ i $T_2(V_2^S, E_2^T)$, wtedy:

$$(H(T_1), \leq_{H(T_1)}) \equiv (H(T_2), \leq_{H(T_2)}).$$

Poniżej (w dowodzie) zostanie wykazane, że dwie, dowolne hierarchie powstałe z tej samej składowej grafu ważonego (z wykorzystaniem algorytmu TC), dla dwóch różnych drzew maksymalnych przepływów, różnią się jedynie numeracją elementów pustych – e . Hierarchie takie są równoważne, gdyż numeracja e jest pomocnicza i nie wnosi nic do struktury hierarchii (relacje porządku są w obu hierarchiach takie same), ani do wartości związków c węzłów. Nie wpływa również na zawartość elementów niepustych (obie hierarchie zachowują ten sam podział zbioru V^S na zbiory Z).

Dowód:

Dla przeprowadzenia dowodu potrzebne jest zdefiniowanie odwzorowania $\varphi: H(T_1) \rightarrow H(T_2)$ spełniającego warunki definicji 5.3.

Z lematu 5.9 wiemy, że trójka $(Z, c, 0)$ jest w zbiorze $H(T_1)$ hierarchii wtedy i tylko wtedy, gdy jest także w zbiorze $H(T_2)$ drugiej hierarchii. Definiujemy zatem odwzorowanie dla niepustych elementów: $\varphi(Z, c, 0) = (Z, c, 0)$, Jest to identyczność.

Pozostaje zdefiniować odwzorowanie φ dla trójek $(\emptyset, c, e)$, czyli dla elementów pustych. Można tutaj skorzystać z lematu 5.12 i zdefiniować $\varphi(\emptyset, c, e) = (\emptyset, c, e')$, gdzie $F^{T_1}(\emptyset, c, e) = F^{T_2}(\emptyset, c, e')$, tzn. elementowi pustemu w jednej hierarchii odpowiada element pusty w drugiej hierarchii, posiadający ten sam zbiór potomków – liści.

Tak zdefiniowane odwzorowanie jest różnowartościowe¹ i "na" oraz spełnia warunek 2 z definicji 5.3. Pozostaje nam pokazać, że spełniany jest także warunek 3 tej definicji.

¹ Możliwe są trzy przypadki:

- Dwa, różne elementy niepuste przechodzą na te same dwa, różne elementy niepuste – identyczność.
- Elementy: pusty i niepusty (które ze względu na zbiór Z , są różne) przechodzą na element pusty i niepusty (które oczywiście także są różne).
- Dwa, różne elementy puste (o różnych zbiorach liści) przechodzą na dwa, różne elementy puste, bo o odpowiednio tych samych, więc różnych zbiorach liści.

Założmy, że tak nie jest. Niech $(Z_1, c_{i_1}, e_1) <_{H(T_1)} (Z_2, c_{i_2}, e_2)$ oraz $\neg[\varphi(Z_1, c_{i_1}, e_1) <_{H(T_2)} (Z_2, c_{i_2}, e_2)]$. Ustalmy dowolne $v_1, v_2 \in V^S$ takie, że: jeżeli $Z_1 \neq \emptyset$, to $v_1 \in Z_1$ oraz jeżeli $Z_1 = \emptyset$, to $v_1 \in Z'_1$, przy czym $(Z'_1, c', 0) \in F^{T_1}(Z_1, c_{i_1}, e_1)$. Analogicznie ustalmy: $v_2 \in Z_2$, jeżeli $Z_2 \neq \emptyset$ lub jeżeli $Z'_2 = \emptyset$, to $v_2 \in Z'_2$, gdzie $(Z'_2, c'', 0) \in F^{T_1}(Z_2, c_{i_2}, e_2)$. Wtedy w pierwszej hierarchii $(H(T_1), \leq_{H(T_1)})$ mamy, że $p(v_1, v_2) = i_2$. Z definicji odwzorowania φ wiemy, że dla $k \in \{1, 2\}$, jeżeli $e_k = 0$, to $\varphi(Z_k, c_{i_k}, 0) = (Z_k, c_{i_k}, 0)$. Jeżeli, natomiast, $e_k \neq 0$, to $Z_k = \emptyset$ i $\varphi(\emptyset, c_{i_k}, e_k) = (\emptyset, c_{i_k}, e'_k)$, gdzie $F^{T_1}(\emptyset, c_{i_k}, e_k) = F^{T_2}(\emptyset, c_{i_k}, e'_k)$. Oznacza to, że w drugiej hierarchii $(H(T_2), \leq_{H(T_2)})$ węzeł v_k jest elementem $\varphi(Z_k, c_{i_k}, e_k)$ lub elementem liścia, którego $\varphi(Z_k, c_{i_k}, e_k)$ jest przodkiem. Ponieważ $\neg[\varphi(Z_1, c_{i_1}, e_1) \leq_{H(T_2)} \varphi(Z_2, c_{i_2}, e_2)]$, więc nie istnieje taki element $(Z^*, c^*, e^*) \in H(T_2)$, że $(Z^*, c^*, e^*) \leq_{H(T_2)} \varphi(Z_2, c_{i_2}, e_2)$, który byłby wspólnym przodkiem dla v_1 oraz v_2 , czyli $p(v_1, v_2) < p(v_2)$. Sprzeczność.

□

Twierdzenie 5.13, korzystając z poprzednich lematów, wniosków i definicji, mówi o niezależności hierarchii od postaci drzewa maksymalnych przepływów.

5.8.3. ROZKŁAD DOKUMENTÓW W HIERARCHII

Każdy element hierarchii, a ściślej jego zbiór Z , można traktować jako grupę dokumentów hipertekstowych¹. W związku z tym, jako efekt działania algorytmu TC grupowania drzewa maksymalnych przepływów otrzymuje się hierarchię grup dokumentów. Posiadają one takie własności, jak elementy hierarchii. W szczególności grupy położone na niższym poziomie (o większym numerze poziomu) mają większą wartość związania, a tym samym większą wartość maksymalnego przepływu między dowolnymi dwoma węzłami w ramach grupy. W konsekwencji w porównaniu do grup wyżej położonych (zobacz także rys. 5.9) grupy niżej położone są ściślej ze sobą związane tematycznie.

Przy założeniu, że składowa grafu ważonego, z którego powstała hierarchia posiada co najmniej dwa dokumenty (węzły), rozkład dokumentów w hierarchii grup (zbiorach Z) jest następujący:

- a) Hierarchia posiada tylko jedną grupę dokumentów (jeden element hierarchii), która jest najwyższa - tylko jeden korzeń.

Wynika to z warunku 6 definicji 5.2, gdyż, jeżeli chcemy by w hierarchii $(H(V^S), \leq_H)$ istniały dwa elementy najwyższe (Z_0, c_0, e_0) i (Z'_0, c'_0, e'_0) , wtedy na mocy warunku 6 definicji mamy, że: $(Z_0, c_0, e_0) \leq_H (Z'_0, c'_0, e'_0)$ oraz $(Z'_0, c'_0, e'_0) \leq_H (Z_0, c_0, e_0)$. Ze wzoru (1.2) wynika natomiast, że $(Z_0, c_0, e_0) = (Z'_0, c'_0, e'_0)$.

- b) W liściach (elementach hierarchii, które nie posiadają potomków) mogą występować wyłącznie grupy (zbiory Z), które zawierają co najmniej 2 dokumenty.

Wynika to wprost z twierdzenia 5.10.

- c) Grupy niebędące liśćmi mogą posiadać dowolną liczbę dokumentów lub nie zawierać w ogóle dokumentów.

Wynika to również z twierdzenia 5.10.

- d) Grupy puste (elementy puste hierarchii) mają co najmniej dwoje dzieci.

¹ Każdemu węzłowi $v_i \in V^S$ odpowiada dokładnie jeden dokument tekstowy $d_i \in D$.

Twierdzenie 5.14.

Każdy, pusty element $(\emptyset, c, e)$, należący do hierarchii $(H(V^S), \leq_H)$, posiada co najmniej dwoje dzieci w tej hierarchii.

Dowód:

Słuszność twierdzenia wynika z kroku 2 i 3 algorytmu TC, w którym usuwana jest przynajmniej jedna krawędź drzewa $T^C(V^C, E^C)$. Z twierdzenia 1.1, pkt 4 wiemy, że usunięcie jednej krawędzi powoduje rozpad drzewa na dwie składowe. Żadna z nich nie jest pojedynczym węzłem (dokumentem) izolowanym, gdyż powstała grupa węzłów Z jest pusta. W związku z tym po usunięciu jednej krawędzi powstają składowe, z których przy następnym wykonywaniu kroków 2, 3, 4, 5 algorytmu TC generowane są dwa elementy będące dziećmi danego elementu.

□

- e) Grupy niepuste mogą posiadać dowolną liczbę dzieci lub nie posiadać ich w ogóle (są wtedy liśćmi hierarchii).

Wniosek ten wynika z rozumowania podobnego jak to, które zostało przedstawione w punkcie poprzednim.

- f) Poszczególne grupy dokumentów są rozłączne.

Wynika to z tego, że hierarchia grup wygenerowana przez algorytm TC z drzewa maksymalnych przepływów, na mocy twierdzenia 5.1, spełnia warunek 2 definicji 5.2.

- g) Każdy dokument (ze zbioru V^S) znajduje się w jakiejś grupie (elemente hierarchii).

Wynika to z twierdzenia 5.1, a szczególności z tego, że hierarchia będąca produktem algorytmu TC spełnia warunek 3 definicji 5.2.

- h) Dla danej składowej grafu ważonego $G^S(V^S, E^S)$ czyli drzewa maksymalnych przepływów $T(V^S, E^T)$ może powstać hierarchia $(H(V^S), \leq_H)$ złożona z maksymalnie tylu grup, ile wynosi liczba krawędzi drzewa, co jest równe liczbie dokumentów (węzłów) w zbiorze V^S minus jeden (dla $\text{card}(V^S) \geq 2$). W związku z tym, prawdziwe jest:

$$\text{card}(H(V^S)) \leq \text{card}(E^T) = \text{card}(V^S) - 1$$

i równość może być osiągnięta.

Dla $V^S=1$ (czyli $E^T=\emptyset$) powstaje hierarchia złożona z jednej, zawierającej 1 dokument grupy.

Maksymalna liczba elementów hierarchii równa liczbie dokumentów minus 1 (równość w powyższej nierówności), wynika z tego, że w kroku 2 i 3 algorytmu TC usuwana jest co najmniej jedna krawędź z drzewa maksymalnych przepływów i powstaje przy tym dokładnie jedna grupa (element hierarchii). Oznacza to, że utworzeniu grupy musi towarzyszyć usunięcie przynajmniej jednej krawędzi. Natomiast z twierdzenia 1.1 pkt 3 wiemy, że w drzewie liczba krawędzi jest zawsze o jeden mniejsza od liczby węzłów.

W przypadku, gdy hierarchia powstała ze składowej złożonej z jednego dokumentu (węzła) izolowanego v_i , tj. $V^S=\{v_i\}$ i $E^S=\emptyset$, wtedy składa się ona wyłącznie z jednej grupy zawierającej jeden dokument. W najgorszym przypadku z całej kolekcji hipertekstowej $HT(D, L)$ może powstać tyle składowych (złożonych z pojedynczych izolowanych dokumentów), ile jest dokumentów w zbiorze D , czyli N składowych i N hierarchii.

5.9. ZŁOŻONOŚĆ OBLICZENIOWA METODY GRUPOWANIA

Metoda grupowania na podstawie drzewa maksymalnych przepływów składa się z sekwencyjnie wykonywanych etapów (pkt 5.1). Etap pierwszy to wydzielenie kolekcji hipertekstowej i w związku z tym jego złożoność obliczeniowa (czasowa) nie będzie rozpatrywana.

W etapie 2 tworzony jest graf ważony podobieństwem dokumentów. Potrzebne jest do tego obliczenie podobieństwa dla każdej pary dokumentów $\{d_i, d_j\}$, czyli dla $\binom{N}{2} = \frac{N(N-1)}{2}$ par. Złożoność czasowa tej operacji ze względu na obliczanie podobieństwa kompleksowego jest rzędu $O(N^2)$.

Złożoność czasowa wydzielenia składowych (etap 3) za pomocą algorytmu PE jest zależna od maksymalnej liczby krawędzi w grafie ważonym, czyli $\binom{N}{2} = \frac{N(N-1)}{2}$. W związku z tym złożoność obliczeniowa 3 etapu jest rzędu $O(N^2)$.

Złożoność tworzenia drzew maksymalnych przepływów (etap 4) zależy od złożoności wykorzystanych algorytmów. Algorytm Gusfielda ma złożoność $O(N)$, a wywoływany w nim, przy każdym przebiegu, algorytm znajdowania minimalnego przekroju i maksymalnego przepływu Goldberga i Tarjana ma złożoność $O(N^2 \log(N))$. W związku z tym złożoność 4 etapu wynosi $O(N^3 \log(N))$.

Złożoność skalowanie wag (5 etap) jest liniowo związana z liczbą krawędzi drzew, czyli jest rzędu $O(N)$, ponieważ krawędzi we wszystkich drzewach jest maksymalnie $N-1$ (twierdzenie 1.1 pkt 4).

Złożoność obliczeniową algorytmu TC determinuje liczba powtórzeń kroków 2, 3, 4, która w najgorszym przypadku wynosi tyle, ile liczba krawędzi drzewa maksymalnych przepływów, gdyż w jednym wykonaniu kroków 2, 3, 4 usuwana jest co najmniej jedna krawędź drzewa maksymalnych przepływów. Krawędzi tych, dla pojedynczego drzewa, jest o 1 mniej niż węzłów (twierdzenie 1.1 pkt 4). Dla wszystkich hierarchii łącznie jest to nie więcej niż $N-1$. W związku z tym złożoność czasowa algorytmu TC jest rzędu $O(N)$.

Ponieważ kolejne etapy metody są wykonywane sekwencyjnie, więc złożoność całej metody wynosi tyle, ile największa złożoność pojedynczego etapu. Największą złożoność ma etap 4, więc złożoność obliczeniowa całej metody jest rzędu $O(N^3 \log(N))$.

5.10. REPREZENTACJA GRUP DOKUMENTÓW

W poprzednich rozdziałach zaprezentowano metodę grupowania dokumentów hipertekstowych, której rezultatem jest hierarchia elementów. Zasadniczą częścią tych elementów jest zbiór Z , zawierający dokumenty tekstowe ze zbioru wejściowego D . Zbiór Z można traktować jako grupę dokumentów i dla tej grupy należy określić reprezentację. Jest ona potrzebna zwłaszcza wtedy, gdy taką grupę / grupy trzeba zaprezentować użytkownikowi.

Sposób wyznaczania reprezentacji grup nie jest częścią immanentną metody grupowania dokumentów hipertekstowych na podstawie drzewa maksymalnych przepływów. Wyznaczanie reprezentanta jest osobnym i szerszym problemem naukowym i nie będzie on tutaj szeroko rozwijany. Badania przeprowadzone w ramach niniejszej pracy miały na celu jedynie znalezienie i przystosowanie istniejących sposobów określania reprezentacji tak, aby były odpowiednie dla opisywanej metody grupowania.

Automatycznie generowana reprezentacja grupy dokumentów powinna posiadać zwięzłą postać, tzn. krótszą niż proste połączenie – konkatenacja treści wszystkich dokumentów z grupy. Oczywi-

ście można także grupy numerować kolejnymi liczbami naturalnymi 1, 2, 3, ..., itd. i te liczby traktować jako reprezentantów, jednakże nie będzie to zrozumiałe dla użytkownika. Reprezentacja powinna być więc związana z treścią dokumentów.

Odpowiednia reprezentacja grupy powinna ułatwiać zastosowanie grupowania przy wyszukiwaniu informacji, tzn. umożliwiać automatyczną lub wspomaganą automatycznie redefinicję pytania wyszukiwawczego w oparciu o reprezentanta.

Zdefiniujmy nowe pojęcie: grupy poszerzonej, która będzie wykorzystywana przy tworzeniu reprezentacji.

Definicja 5.9.

Grupa poszerzona dla danego elementu (Z,c,e) , należącego do hierarchii $(H(V^S), \leq_H)$, nazywamy zbiór dokumentów, oznaczany przez $EC(Z,c,e)$, taki że:

$$EC(Z,c,e) = \bigcup \{Z': (\exists (Z',c',e') \in H(V^S)) (Z',c',e') \leq_H (Z,c,e)\}$$

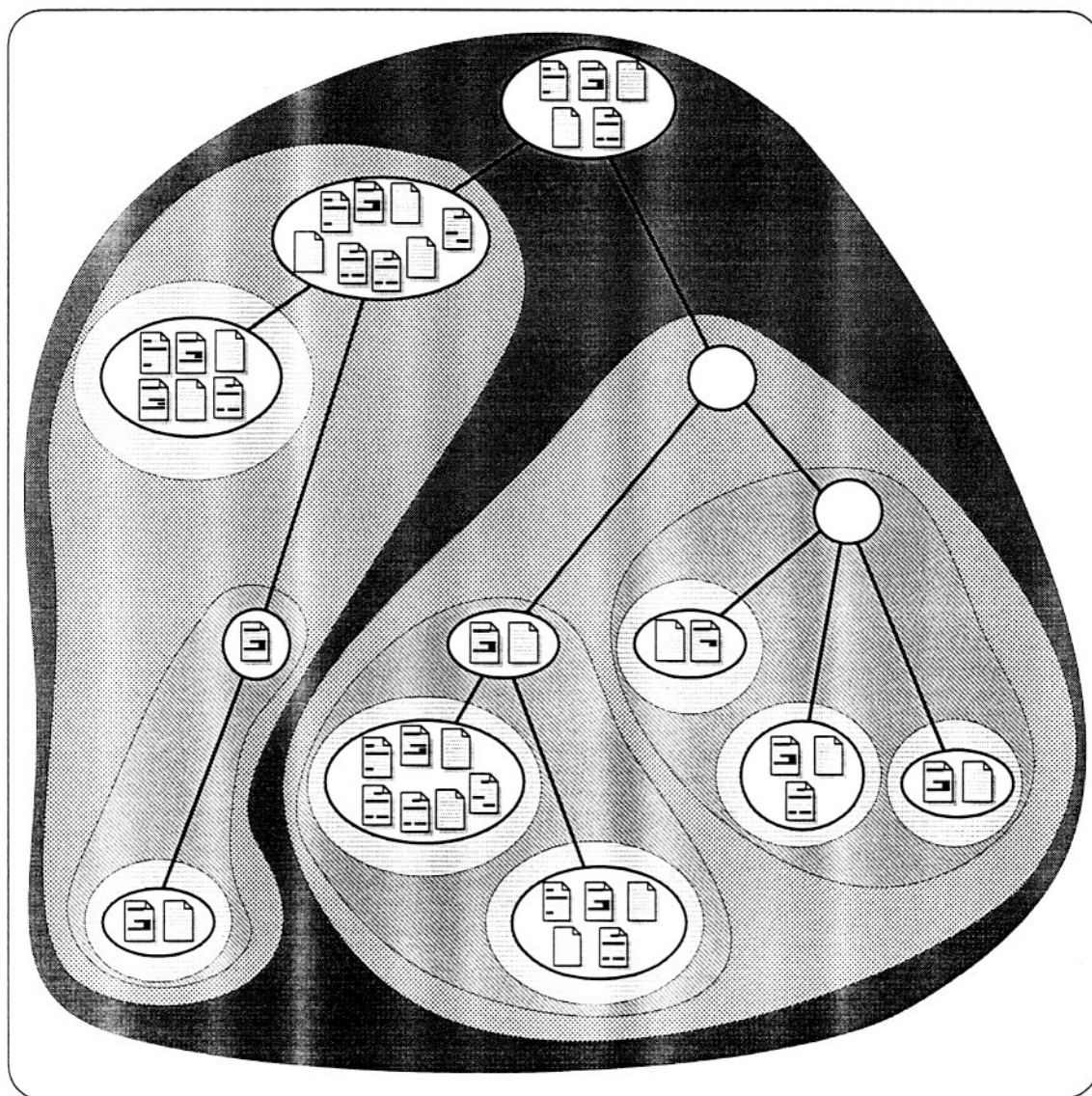
Oznacza to, że grupa poszerzona jest zbiorem dokumentów, będących częścią elementów hierarchii, dla których (Z,c,e) jest przodkiem. Wydzielamy zatem „podhierarchię” z hierarchii $(H(V^S), \leq_H)$, której korzeniem jest element (Z,c,e) .

Dokumenty ze zbioru Z (z elementu (Z,c,e)) także należą do $EC(Z,c,e)$ ¹, gdyż na mocy wzoru (1.1), mamy $(Z,c,e) \leq_H (Z,c,e)$.

W przypadku, gdy element (Z,c,e) jest liściem hierarchii, to $EC(Z,c,e) = Z$, gdyż tylko on sam jest ze sobą w relacji $\leq_H$.

Rysunek 5.14 zawiera hierarchię z rys. 5.9, w której, dla wszystkich jej elementów, zaznaczono grupy poszerzone.

¹ Oznaczenie EC pochodzi od ang. *extended cluster*



Rys. 5.14. Hierarchia z rys. 5.9 z zaznaczonymi grupami poszerzonymi.

Twierdzenie 5.15.

Dla każdego elementu (Z, c, e) , należącego do hierarchii $(H(V^S), \leq_H)$, mamy:

$$EC(Z, c, e) \neq \emptyset.$$

Dowód:

Możliwe są dwie sytuacje:

1. Jeżeli $Z = \emptyset$, wtedy na mocy lematu 5.11 mamy, że $\text{card}(F(Z, c, e)) \geq 2$. Z tego i z twierdzenia 5.10 wynika, że istnieje $(Z', c', e') \in F(Z, c, e)$, takie, że $\text{card}(Z') \geq 2$, czyli $EC(Z, c, e) \neq \emptyset$.
2. Jeżeli $Z \neq \emptyset$, to ponieważ $Z \subseteq EC(Z, c, e)$, więc $EC(Z, c, e) \neq \emptyset$.

□

Przypomnijmy, że z grafu ważonego $G(V, E^G)$ – w 3 etapie metody – wydzielanych jest M rozłącznych składowych:

$$G_i^S(V_i^S, E_i^S), \text{ dla } i = 1, 2, 3, \dots, M.$$

Dla każdej takiej składowej $G_i^S(V_i^S, E_i^S)$, czyli dla odpowiadającego jej drzewa, powstaje osobna, hierarchia grup.

Na podstawie definicji składowej (1.5) mamy:

$$\bigcup_{i=1}^M V_i^S = V = D.$$

Definicja 5.10.

Dopełnieniem grupy poszerzonej $EC(Z, c, e)$, które będzie się oznaczać przez $EC^C(Z, c, e)$, nazywamy taki zbiór dokumentów, że:

$$EC^C(Z, c, e) = D \setminus EC(Z, c, e).$$

Zbiór $EC^C(Z, c, e)$ jest dopełnieniem (ang. *complement*) zbioru $EC(Z, c, e)$ do całej kolekcji dokumentów D . W szczególności znajdują się w nim dokumenty z innych hierarchii niż ta, do której należy element (Z, c, e) .

Twierdzenie 5.16.

Dla każdego elementu (Z, c, e) należącego do dowolnej hierarchii $(H(V^S), \leq_H)$ powstałej z kolekcji hipertekstowej $HT(D, L)$, mamy:

$$M=1 \wedge (Z, c, e) = (Z_0, c_0, e_0) \in H(V^S) \Leftrightarrow EC^C(Z, c, e) = \emptyset,$$

gdzie (Z_0, c_0, e_0) – korzeń hierarchii.

Dowód:

1. Jeżeli $M=1$, czyli $V^S=D$ oraz (Z, c, e) jest korzeniem, to na mocy def. 5.9 oraz 5.2, pkt 6 mamy, że $EC(Z, c, e)=V^S=D$, czyli $D \setminus EC(Z, c, e)=EC^C(Z, c, e)=\emptyset$. Kończy to dowód w jedną stronę.
2. Wynikanie w drugą stronę jest równoważne dwóm wynikaniom:

$$EC^C(Z, c, e) = \emptyset \Rightarrow M=1$$

i jednocześnie

$$EC^C(Z, c, e) = \emptyset \Rightarrow (Z, c, e) = (Z_0, c_0, e_0).$$

W celu udowodnienia pierwszego z nich założmy, nie wprost, że $M>1$. Wtedy istnieje więcej hierarchii dla kolekcji hipertekstowej $HT(D, L)$ niż tylko $(H(V^S), \leq_H)$. Są one niepuste, tzn. zawierają co najmniej 1 dokument¹. Dokumenty z tych innych hierarchii oczywiście należą do zbioru $EC^C(Z, c, e)$, czyli $EC^C(Z, c, e) \neq \emptyset$. Sprzeczność.

¹ Zbiór węzłów każdej składowej musi zawierać co najmniej jeden węzeł.

Założmy teraz, nie wprost, że $(Z,c,e) \neq (Z_0,c_0,e_0)$, czyli $(Z,c,e) <_H (Z_0,c_0,e_0)$, wtedy możliwe są dwie sytuacje:

- a) $Z_0 \neq \emptyset$, więc $Z_0 \subset Z$, czyli $Z_0 \subset EC^C(Z,c,e) \neq \emptyset$, zatem $EC^C(Z,c,e) \neq \emptyset$.
- b) Niech $Z_0 = \emptyset$. Z twierdzenia 5.14, wiemy, że element (Z_0,c_0,e_0) posiada co najmniej dwoje dzieci. Weźmy z tych dzieci takie (Z',c',e') , że $(Z,c,e) \leq_H (Z',c',e')$. Niech (Z'',c'',e'') będzie, natomiast, dowolnym innym dzieckiem korzenia hierarchii. Ponieważ elementy (Z',c',e') i (Z'',c'',e'') są różnymi dziećmi tego samego elementu (Z_0,c_0,e_0) , więc $EC(Z',c',e') \cap EC(Z'',c'',e'') = \emptyset$. Z twierdzenia 5.15 mamy, że $EC(Z'',c'',e'') \neq \emptyset$. Ponieważ $EC(Z'',c'',e'') \subset EC^C(Z,c,e)$, więc $EC^C(Z,c,e) \neq \emptyset$. Sprzeczność

W obu sytuacjach otrzymujemy sprzeczność; kończy to dowód.

□

Rozpatrzmy hierarchię z rys. 5.13. Założmy, że jest to jedyna hierarchia powstała z kolekcji hipertekstowej $HT(D,L)$. Weźmy z tej hierarchii element $(\{1,2,8\},3,0)$. Dla tego elementu, zbiór $EC(\{1,2,8\},3,0) = \{1,2,3,4,5,6,8,9,10\}$ ¹, a $EC^C(\{1,2,8\},3,0) = \{7\}$. Dla elementu $(\{3\},4,0)$ natomiast zbiór $EC(\{3\},4,0) = \{3,4,5,6\}$, a $EC^C(\{3\},4,0) = \{1,2,7,8,9,10\}$.

5.10.1. WSKAŹNIKI WAŻNOŚCI

Ponieważ reprezentacja grupy, jak wspomniano wyżej, powinna być związana z treścią dokumentów, więc reprezentantów można szukać wśród terminów występujących w dokumentach. Jeżeli pewien termin dobrze opisuje (jest dobrym deskryptorem) grupę dokumentów, to nie powinien on dobrze opisywać pozostałych dokumentów. Założenie to jest szczególnie przydatne wtedy, gdy chcemy wykorzystać owe dobre deskryptory do redefinicji pytania wyszukiwawczego. Wynika to z tego, że w odpowiedzi na takie zmodyfikowane pytanie powinno się uzyskać przede wszystkim dokumenty z reprezentowanej grupy i nie otrzymać pozostałych.

Hierarchia, będąca produktem metody grupowania na podstawie drzewa maksymalnych przepływów, ma te cechy, że dla danego elementu (Z,c,e) , dokumenty pochodzące z $EC(Z,c,e)$ są wszystkie ze sobą związane na poziomie co najmniej c . Element (Z,c,e) wyznacza zatem związek dla wszystkich dokumentów z grupy poszerzonej.

Jako reprezentację grupy dokumentów Z (pochodzącej z elementu (Z,c,e) hierarchii) proponuje się więc terminy (słowa kluczowe), które byłyby dobrymi deskryptorami nie tylko samego zbioru Z , ale całej jego grupy poszerzonej. Można więc stwierdzić, że reprezentacja grupy dokumentów Z jest równocześnie reprezentacją całego drzewa, dla którego element (Z,c,e) jest korzeniem. Jednocześnie słowa kluczowe – reprezentanci nie powinni być dobrymi deskryptorami reszty kolekcji dokumentów czyli nie powinni reprezentować dopełnienia grupy poszerzonej $EC^C(Z,c,e)$.

Jest to koncepcja bliska metodzie **wskaźników ważności** (ang. *cue validity*), która pochodzi z psychologii kognitywnej², sprzed prawie 30 lat. Smith i Medin (1981) sformułowali wzór opisujący wskaźnik ważności. Wzór ten został następnie zaadaptowany przez Goldberga, dla potrzeb metody CMD (*Category Discrimination Method*) [Gol96], a później wykorzystany przez Chang i Hsu do grupowania stron WWW w systemie *WebYacht* [Chan99].

Wskaźnik ważności można opisać jako:

¹ Jest to suma grup: $\{1,2,8\}$, $\{3\}$, $\{4,5,6\}$, $\{9,10\}$.

² Pojęcie *cue validity* zostało po raz pierwszy zdefiniowane przez psychologów Roscha i Mervisa w 1975r..

$$cv_{t,Z} = \frac{P(t | EC(Z, c, e))}{P(t | EC(Z, c, e)) + P(t | EC^C(Z, c, e))}, \quad (5.5)$$

gdzie:

- $cv_{t,Z}$ – wskaźnik ważności terminu t dla grupy dokumentów Z , czyli dla grupy rozszerzonej $EC(Z, c, e)$ ¹;
- $P(t | EC(Z, c, e))$ – prawdopodobieństwo wystąpienia terminu t w grupie dokumentów $EC(Z, c, e)$ ²;
- $P(t | EC^C(Z, c, e))$ – prawdopodobieństwo wystąpienia terminu t w grupie dokumentów $EC^C(Z, c, e)$ ³.

Wartości wskaźników ważności należą do przedziału $[0, 1]$.

Prawdopodobieństwo wystąpienia terminu t w grupie dokumentów ($EC(Z, c, e)$ lub $EC^C(Z, c, e)$), można określić jako zmodyfikowaną częstość występowania (zmodyfikowaną liczbę wystąpień) tego terminu w danej grupie dokumentów, ($tf'_{EC(Z, c, e), t}$ lub $tf'_{EC^C(Z, c, e), t}$), podzieloną przez zmodyfikowaną częstość występowania tego terminu we wszystkich dokumentach z kolekcji D czyli $tf'_{D, t}$. Otrzymujemy zatem:

$$P(t | EC(Z, c, e)) = \frac{tf'_{EC(Z, c, e), t}}{tf'_{D, t}},$$

oraz:

$$P(t | EC^C(Z, c, e)) = \frac{tf'_{EC^C(Z, c, e), t}}{tf'_{D, t}}.$$

Podstawiając to do wzoru (5.5) uzyskujemy:

$$cv_{t,Z} = \frac{tf'_{EC(Z, c, e), t}}{tf'_{EC(Z, c, e), t} + tf'_{EC^C(Z, c, e), t}}. \quad (5.6)$$

Jeżeli $tf'_{EC^C(Z, c, e), t}$ jest równy zero, czyli dany termin nie występuje w dokumentach z dopełnienia grupy poszerzonej, wtedy dla każdego terminu t , należącego do dokumentów grupy poszerzonej, $cv_{t,Z} = 1$.

5.10.2. WYZNACZANIE REPREZENTACJI GRUPY DOKUMENTÓW

Korzystając ze wskaźników ważności wyznaczmy reprezentację grupy jako zbiór słów kluczowych (terminów). Wcześniej – korzystając ze wzoru (5.6) – określmy wskaźnik ważności dla dopełnienia grupy poszerzonej, który oznaczmy przez $cv_{t,Z}^C$:

$$cv_{t,Z}^C = \frac{tf'_{EC^C(Z, c, e), t}}{tf'_{EC(Z, c, e), t} + tf'_{EC^C(Z, c, e), t}}$$

¹ W pierwotnych definicjach wskaźnik ważności oznaczał ważność cechy (ang. *feature*) w danym pojęciu (ang. *concept*). W naszym przypadku cechą będzie termin, a pojęciem – grupa dokumentów.

² Pierwotnie oznaczało to prawdopodobieństwo wystąpienia danej cechy w pojęciu, dla którego wyznaczamy wskaźnik ważności.

³ W pierwotnej wersji oznaczało to prawdopodobieństwo wystąpienia danej cechy w pojęciu przeciwnym, kontrastowym do tego, dla którego wyznaczamy wskaźnik ważności.

Wskaźnik ważności $cv_{t,Z}^C$ określa, czy termin t jest „dobrym deskryptorem” dla dokumentów z dopełnienia grupy poszerzonej.

Definicja 5.11.

Reprezentacja elementu (Z,c,e) , należącego do hierarchii $(H(T), \leq_{H(T)})$, przy czym, $(Z,c,e) \neq (Z_0,c_0,e_0)$ lub liczba M hierarchii powstałych z kolekcji hipertekstowej $HT(D,L)$ jest większa od 1, nazywamy zbiór takich terminów t , dla których:

$$cv_{t,Z} > \delta_1 \quad \text{ i } \quad cv_{t,Z}^C < \delta_2,$$

gdzie:

t – termin występujący w dokumentach ze zbioru D ,

δ_1 – próg ważności grupy poszerzonej,

δ_2 – próg nieważności dopełnienia grupy poszerzonej.

Jeżeli $(Z,c,e) = (Z_0,c_0,e_0)$ oraz $M=1$, wtedy reprezentacja elementu (Z,c,e) jest zbiorem pustym.

Reprezentację elementu (Z,c,e) oznaczmy przez $R(Z,c,e)$.

Reprezentację tworzą słowa kluczowe, które wyróżniają się w dokumentach z grupy poszerzonej a nie wyróżniają się w dopełnieniu tej grupy.

Parametry (progi) δ_1 i δ_2 należą do przedziału $[0,1]$ a sens mają jedynie takie ich wartości, dla których spełnione jest: $\delta_1 \geq \delta_2$

Zalety zastosowania wskaźników ważności:

- Automatyczne usuwanie słów, które często występują w danej grupie oraz często w pozostałych grupach (zaleta a czasami wada), więc w konsekwencji nie potrzebna jest zwykle tzw. stop lista.
- Wydzielenie słów – kandydatów na słowa kluczowe odróżniające daną grupę od innych.
- Duża efektywność – stosunkowo mało potrzebnych obliczeń (porównań), gdyż wykorzystywane są tylko dwie wartości prawdopodobieństwa: dla danej grupy i dla reszty, a nie w stosunku do każdego dokumentu.
- Dobór progów δ_1 oraz δ_2 o odpowiednio dużej wartości umożliwia uzyskanie mniejszego lub większego zbioru terminów w reprezentacji. Większa wartość δ_1 powoduje zmniejszenie liczby terminów spełniających nierówność $cv_{t,Z} > \delta_1$.

Wady stosowania wskaźników ważności w postaci wzoru (5.6) wynikają z niedogodności tej metody dla małych zbiorów dokumentów. W sytuacjach, gdy dokumenty z dopełnienia grupy poszerzonej nie zawierają terminu t , wtedy $tf'_{EC^C(Z,c,e),t} = 0$. Konsekwencją tego jest:

- Brak rozróżnienia dużego $tf'_{EC(Z,c,e),t}$ od małego $tf'_{EC^C(Z,c,e),t}$ (w obu przypadkach $cv_{t,Z}$ ma największą wartość: 1).
- Uwydatnianie. Jeżeli przykładowo, dla pewnego terminu t , wartość $tf'_{EC(Z,c,e),t} = 8$, a $tf'_{EC^C(Z,c,e),t} = 2$, to $cv_{t,Z} = 0.8$. Jeżeli jednak $tf'_{EC^C(Z,c,e),t} = 0$ i $tf'_{EC(Z,c,e),t} = 1$, wtedy $cv_{t,Z} = 1$.

Terminy, sporadycznie występujące w danej grupie poszerzonej, mogą mieć większy wskaźnik ważności niż te, częściej występujące.

Wady te są nieuniknione dla małych zbiorów. W związku z tym Chang i Hsu przedstawili dwie propozycje [Chan98, Chan99]:

- a) Dodanie stałego parametru ε o małej wartości, w mianowniku. Umożliwi to rozróżnienie mniejszych wartości $tf'_{EC(Z,c,e),t}$ od większych, dla $tf'_{EC^C(Z,c,e),t}=0$.
- b) Normalizacja poprzez odwrotność częstości występowania terminu w całej kolekcji wyrażoną przez: $\log\left(\frac{1}{tf'_{D,t}}\right)$. Cechą tak wyrażonego wskaźnika ważności jest przeniesienie rzadkich słów uwydatnionych przez odwrotność częstości występowania we wszystkich dokumentach do stosunkowo znaczących.

Uzyskujemy zatem:

$$cv_{t,Z} = \frac{tf'_{EC(Z,c,e),t}}{tf'_{EC(Z,c,e),t} + tf'_{EC^C(Z,c,e),t} + \varepsilon} \times \log\left(\frac{1}{tf'_{D,t}}\right).$$

6. EKSPERYMENTALNA WERYFIKACJA METODY GRUPOWANIA

6.1. PROGRAM EKSPERYMENTÓW

W celu zbadania efektów zastosowania metody grupowania dokumentów na podstawie drzewa maksymalnych przepływów do rzeczywistych dokumentów przeprowadzono eksperymenty dla stron systemu WWW. Jako źródło kolekcji hipertekstowej wykorzystano listy odpowiedzi zwracanych przez wyszukiwarki internetowe.

Rozważano także przeprowadzenie eksperymentów dla tematycznych katalogów stron (na przykład Yahoo). Katalogi te są przede wszystkim zbiorami odesłań do stron domowych wybranych serwisów informacyjnych (strony domowe zwykle nie zawierają informacji tematycznej a jedynie odesłania „w głąb” serwisu). Jednocześnie katalogi komercyjne pozostają pod wpływem powiązań reklamowych, czego efektem jest częstsza obecność niektórych serwisów w różnych kategoriach.

W związku z tym, eksperymenty przeprowadzono na listach odpowiedzi wyszukiwarek internetowych. W tym celu opracowano system CIS (*Clustering Investigation System*), który umożliwia:

- a) Pobranie od użytkownika pytania w postaci listy słów kluczowych i przekazanie go do metawyszukiwarki MetaCrawler.
- b) Sprowadzenie stron WWW znajdujących się na liście odpowiedzi metawyszukiwarki. Strony te tworzą kolekcję hipertekstową, dla której jest przeprowadzane grupowanie.
- c) Zbudowanie hierarchii grup dokumentów z kolekcji sprowadzonych stron, z wykorzystaniem metody grupowania na podstawie drzewa maksymalnych przepływów.
- d) Prezentację hierarchii grup użytkownikowi, z możliwością wyświetlenia treści dokumentów.

Korzystając z systemu CIS utworzono 80 kolekcji (będących zbiorami odpowiedzi na 80 różnych pytań), zawierających łącznie ponad 6500 dokumentów. Najbardziej czasochłonne i kosztowne eksperymenty polegały na ocenie przez ekspertów wygenerowanych przez system CIS hierarchii. Każda kolekcja związana była z innym pytaniem sformułowanym przez eksperta. Jako eksperci zostali zaangażowani pracownicy naukowcy Politechniki Wrocławskiej. Wszyscy eksperci zadawali po dwa pytania. Z przyczyn organizacyjnych (ograniczone środki) przeprowadzono ocenę jedynie 25 kolekcji, których listę zamieszczono w zał. B.

Po wygenerowaniu przez system CIS hierarchii grup, ekspert oceniał prawidłowość grupowania proponując przeniesienia poszczególnych dokumentów. Wyróżniono następujące rodzaje przeniesień:

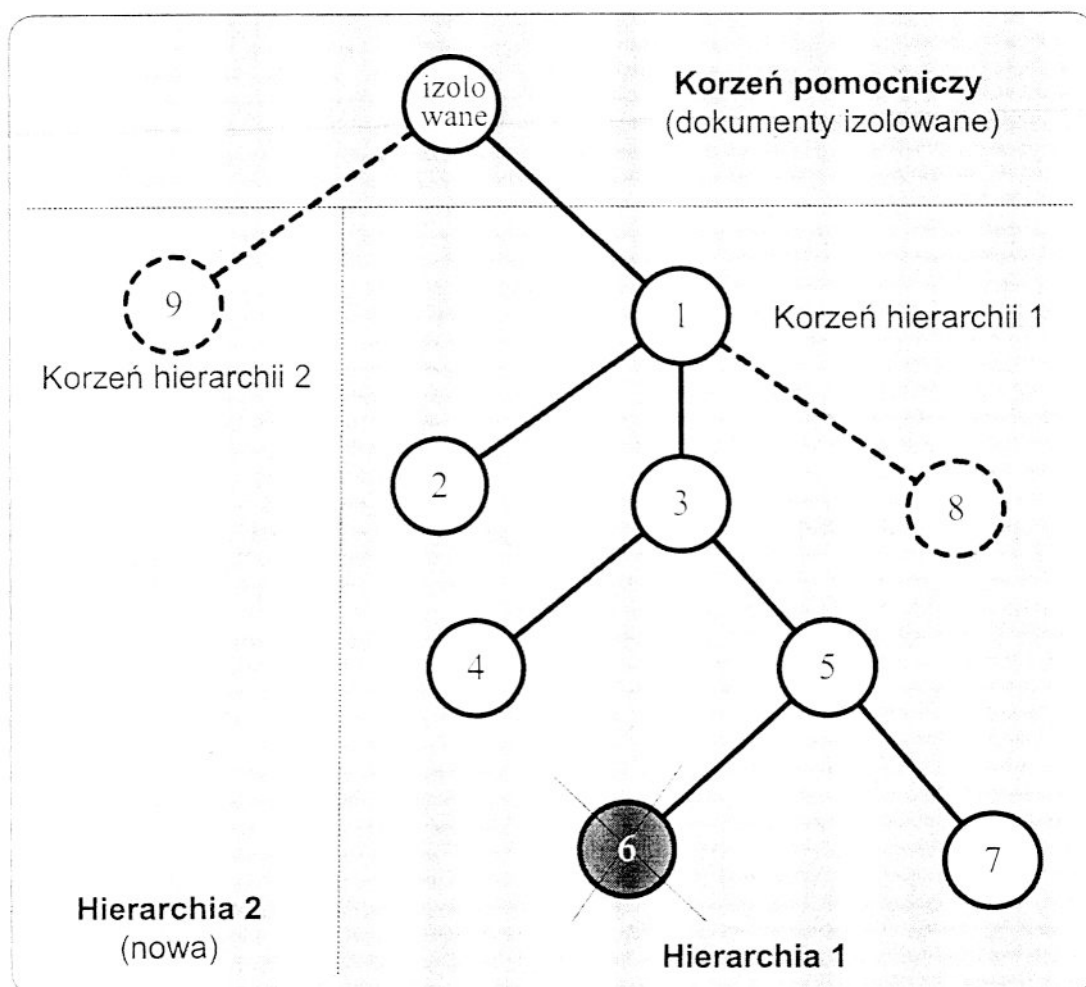
- a) do grupy szczegółowszej (do grupy – potomka);
- b) do grupy ogólniejszej (do grupy – przodka);
- c) do innej gałęzi (także do innej hierarchii);
- d) do grupy dokumentów izolowanych;
- e) z dokumentów izolowanych.

W wyniku przeniesień powstawały nowe grupy a nawet nowe hierarchie. Nowa grupa (hierarchia) była tworzona wtedy, gdy ekspert przeniósł do niej co najmniej 2 dokumenty. Istniejące grupy (hierarchie) ulegały likwidacji, poprzez przeniesienie wszystkich dokumentów do innych grup.

Ewentualne przemieszczenie całej grupy polegało na przeniesieniu do nowej grupy wszystkich jej dokumentów (usuwana była jedna grupa i tworzona inna). Analogicznie było w przypadku łączenia całych grup.

W trakcie oceny wypełniany był specjalnie opracowany formularz, oddzielnie dla każdej kolekcji (zał. C).

Zmiany wprowadzane przez ekspertów wyrażane były **przesunięciem dokumentu**. Przesunięcie dokumentu to długość najkrótszej ścieżki, jaką należy pokonać w hierarchii (rozumianej jako drzewo), aby dany dokument przenieść z jego aktualnej grupy, do grupy docelowej. W celu określenia przesunięcia dokonano między hierarchiami oraz do i z grupy dokumentów izolowanych, dla potrzeb obliczeniowych, generowano hierarchię rozszerzoną, w której z grupy dokumentów izolowanych tworzono korzeń pomocniczy. Do korzenia tego połączone są (jako dzieci) korzenie hierarchii wygenerowanych przez system CIS (lub utworzonych przez eksperta) – rys. 6.1.



Rys. 6.1. Rozszerzona hierarchia dla kolekcji 9.

Przesunięcie dokumentów przenoszonych z grupy izolowanych do pierwszej (lub odwrotnie) wynosi 1. Przesunięcie pojedynczego dokumentu z grupy 2 do 3, wynosi 2. Dla dokumentów przenoszonych z grupy 6 do 9, przesunięcie równa się 5.

Linia przerywaną oznaczono nowe grupy, utworzone przez eksperta: 8 i 9; przy czym grupa 9 tworzy nową hierarchię. Wszystkie dokumenty z grupy 6 zostały przez eksperta przeniesione do innych grup, a 6 uległa likwidacji.

Dla oceny wygenerowanych hierarchii opracowano następujące wskaźniki, które w sposób liczbowy wyznaczają poprawność grupowania:

- $\%DNP$ – procent dokumentów nieprzeniesionych;

$$\%DNP = \frac{N - DP}{N},$$

gdzie: DP – liczba dokumentów, które zostały przeniesione,
 N – liczba dokumentów w kolekcji.

- b) $\acute{S}PDP$ – średnie przesunięcie dla dokumentów przeniesionych:

$$\acute{S}PDP = \frac{WP}{DP},$$

gdzie: WP – suma wartości wszystkich przesunięć.

- c) $\acute{S}PDK$ – średnie przesunięcie dokumentów kolekcji (dla wszystkich dokumentów należących do kolekcji). Określa ono o ile poziomów, średnio należy przesunąć dokument, aby uzyskać hierarchię proponowaną przez eksperta:

$$\acute{S}PDK = \frac{WP}{N}.$$

- d) $\%NG$ – procent grup nowych:

$$\%NG = \frac{NG}{U'},$$

gdzie: NG – liczba nowych grup utworzonych przez eksperta,
 U' – liczba wszystkich grup po ocenie eksperta (bez dokumentów izolowanych).

- e) $\%ZG$ – procent grup zlikwidowanych:

$$\%ZG = \frac{ZG}{U},$$

gdzie: ZG – liczba grup zlikwidowanych przez eksperta,
 U – liczba wszystkich grup wygenerowanych przez system CIS (przed oceną eksperta, bez dokumentów izolowanych).

Dla hierarchii z rys. 6.1 wskaźnik $\%NG = \frac{2}{8} = 25\%$, natomiast $\%ZG = \frac{1}{7} = 14.3\%$.

Im wartości $\%DNP$ oraz $\%NG$ są bliższe 100%, a $\%ZG$ bliższe 0%, tym mniejszych zmian – w opinii eksperta – wymagają wygenerowane przez system CIS hierarchie.

Wskaźnik $\acute{S}PDP$ wyraża „jak bardzo należało przenosić dokumenty, aby uzyskać prawidłowe hierarchie”. Nie umożliwia on jednak porównania wyników dla kilku kolekcji, gdyż nie uwzględnia liczności całej kolekcji. Wskaźnik $\acute{S}PDK$ łączy oba ww. parametry. Określa on „o ile średnio należy przenieść każdy dokument w kolekcji, aby uzyskać właściwe hierarchie”.

Przeprowadzono kolejno dwie serie eksperymentów:

- Zaprezentowana w 4 i 5 rozdziale metoda grupowania zawiera kilka parametrów: max^{lev} , τ , μ , α , β oraz γ . Pierwsza seria eksperymentów miała na celu wyznaczenie wartości tych parametrów i została przeprowadzona z wykorzystaniem wybranych kolekcji (kpt. 6.3). Badano wpływ zmian wartości każdego parametru (przy stałych wartościach pozostałych) na wskaźniki ocen oraz postać hierarchii.
- Celem drugiej serii eksperymentów była odpowiedź na pytanie: czy utworzone przez system CIS hierarchie są prawidłowe? Seria ta została przeprowadzona z ustalonymi w pierwszej serii wartościami parametrów metody, dla 25 kolekcji dokumentów zawierających łącznie 1839 stron WWW (pkt 6.4).

W ramach eksperymentów wyznaczano:

- a) Parametry kolekcji, tj.:
- liczbę dokumentów w kolekcji (N);
 - odsetek dokumentów zawartych w liście odpowiedzi z metawyszukiwarki, których treści nie udawało się sprowadzić;
 - strukturę hipertekstową kolekcji wyrażoną przez łączną liczbę odsyłaczy oraz średnią liczbę odsyłaczy przypadającą na jeden dokument;
 - wielkość tekstów dokumentów wyrażoną przez średnią liczbę terminów przypadających na jeden dokument.
- b) Parametry grafu ważonego utworzonego z wykorzystaniem funkcji podobieństwa kompleksowego, tj.:
- liczbę dokumentów izolowanych (także odsetek tych dokumentów w stosunku do wielkości kolekcji N);
 - liczbę krawędzi w grafie;
 - średni stopień wężła (podwojona liczba krawędzi podzielona przez liczbę węzłów, łącznie z węzłami izolowanymi) oraz średni stopień wężła bez dokumentów izolowanych; średni stopień wężła określa „na ile graf jest gęsty”;
 - liczbę krawędzi łączących dokumenty, dla których zmodyfikowane podobieństwo strukturalne jest większe od 0 (które są połączone bezpośrednimi lub pośrednimi odsyłaczami) a także liczbę dokumentów (jako część kolekcji), będących końcowymi dla chociaż jednej takiej krawędzi; wartości te określają „na ile niezerowe podobieństwo strukturalne ma wpływ na postać grafu”.
- c) Postać hierarchii, tj.:
- liczbę hierarchii;
 - liczbę wszystkich grup;
 - liczbę liści (łącznie ze wszystkich hierarchii);
 - średnią liczbę dzieci („ile średnio dzieci posiadają elementy hierarchii nie będące liśćmi”), która określa na ile hierarchie są „rozbudowane wszcz”; dla średniej liczby dzieci równej 1 mamy, że żaden element hierarchii nie ma więcej niż 1 dziecko; wartość tę można obliczyć wyrażeniem: $\frac{\text{liczba grup} - \text{liczba hierarchii}}{\text{liczba grup} - \text{liczba liści}}$;
 - średnią liczbę dokumentów w grupie.
- d) Jakość wygenerowanych hierarchii, poprzez wskaźniki: $\%DNP$, $\%NG$, $\%ZG$, $\%PDP$, $\%PDK$.

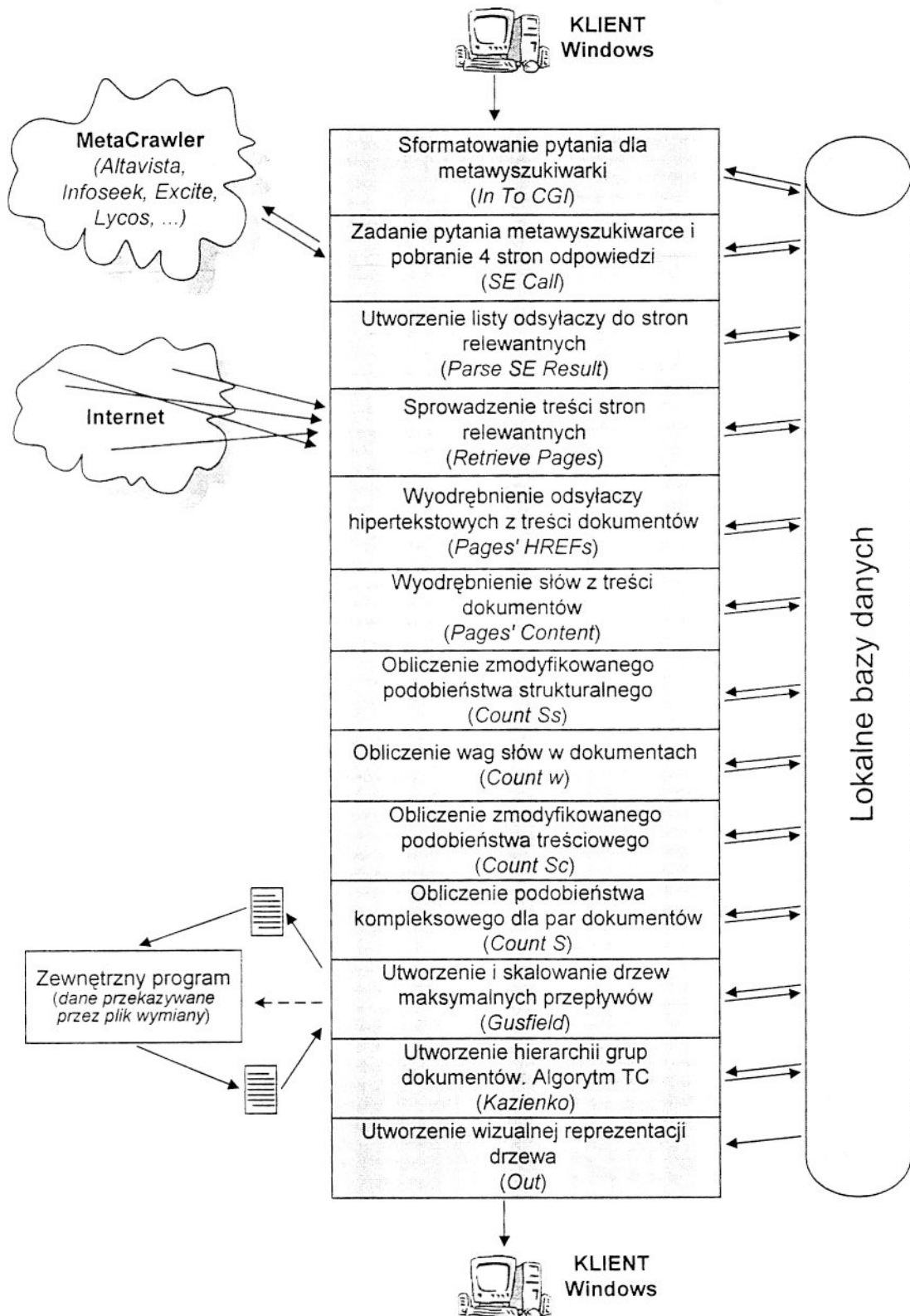
Liczby hierarchii, grup oraz liści podawane w opisach eksperymentów, nie uwzględniają dokumentów izolowanych. Natomiast przy ustalaniu średniego stopnia wężła w grafie, węzły izolowane były brane pod uwagę.

6.2.SYSTEM CIS

Metodę grupowania dokumentów hipertekstowych zaimplementowano w systemie CIS. Został on wykonany we współpracy ze studentami Wydziału Informatyki i Zarządzania Politechniki Wrocławskiej [Gdu99].

Struktura systemu CIS odzwierciedla kolejne etapy metody grupowania. Został on jednak rozszerzony o część związaną z uzyskaniem kolekcji dokumentów oraz prezentację hierarchii w postaci

drzewa grup (rys. 6.2). Kolekcją dokumentów jest zbiór stron WWW, uzyskany na podstawie listy odpowiedzi metawyszukiwarki MetaCrawler (<http://www.metacrawler.com>), na pytanie przekazane przez użytkownika do systemu. Pytanie składa się listy słów kluczowych połączonych logicznym operatorem „i”. Ponieważ MetaCrawler posiada ograniczenia na liczbę odpowiedzi umieszczonych na 1 stronie (max. 30) oraz na liczbę odpowiedzi pobieranych z poszczególnych wyszukiwarek (max. 30), więc w systemie CIS sprowadzanych jest do 4 stron odpowiedzi po max. 30 dokumentów każda. W efekcie kolekcja może zawierać do 120 dokumentów.



Rys. 6.2. Architektura systemu CIS.

Na każdym etapie działania systemu CIS dane wejściowe i wyjściowe są zapamiętywane w bazach danych. Działanie programu można przerwać i wznowić po każdym zakończonym etapie. Ułatwia to śledzenie i analizę efektów grupowania.

Z przyczyn organizacyjnych w systemie CIS zaimplementowano tylko wersję D funkcji pierwotnego podobieństwa strukturalnego (pkt 4.2).

Na rys. 6.3 przedstawiono ekran z systemu CIS w trakcie sprowadzania przez system stron WWW z sieci Internet. Adresy stron były zawarte w liście zwróconej przez metawyszukiwarkę, w odpowiedzi na pytanie: „gsm send”. Na rysunku umieszczono także fragment zawartości zbioru bazy danych, zawierającego informacje o sprowadzonych dokumentach (adres, treść HTML, wydzielony tytuł, opis i słowa kluczowe, liczbę odsyłaczy).

Sprowadzanie stron odbywa się wielowątkowo: 10 stron jednocześnie. Dla każdej strony system 5-krotnie próbuje nawiązać połączenie HTTP. Gdy żadne z połączeń nie dojdzie do skutku, strona jest pomijana¹. Nawiązanie połączenia HTTP nie gwarantuje uzyskania właściwej treści strony, gdyż serwer WWW może w odpowiedzi zwrócić informację o błędzie. Również w trakcie sprowadzania może zostać przerwana transmisja (zbyt wolne łącza, uszkodzenia elementów sieci Internet).

Strony sprowadzane są w postaci tekstowej – bez elementów multimedialnych.

The screenshot shows the Clustering Investigation System (C) 2000 PP [beta release] interface. The window title is "Clustering Investigation System (C) 2000 PP [beta release]". The menu on the left includes: File, Tools, Help, 1. In To CGI, 2. SE Call, 3. Parse SE Result, 4. Retrieve Pages, 5. Pages' HREFs, 6. Pages' Content, 7. Count Ss, 8. Count w, 9. Count Sc, 10. Count S, 11. Gustfield, 12. Kazienko, 13. Out. The main area displays a list of URLs under the heading "DocOK.DB". The list includes URLs such as "http://www.boss.ch/boss/gsm.htm...already in DB", "http://www.omnitel.it", "http://www.musicus.it", "http://www.dcs.ed.a", "http://www.gin.nl/lin", "http://www.digifone", "http://www.cher.com", "http://www.omnitel", "http://personalinet.fi", "http://www.broadba", "http://www.cher.com", "http://www.monico.u", "http://www.binteme", "http://home.swipnet", "http://www.infonet", "http://www.ais900.c", "http://www.fonlan.c", "http://www.dcs.ed.a", "http://www.fonlan.c", "http://infweb.nhh.no", "http://www.rad.net.i", "http://www.sms-it.cc", "http://stud.hials.no/~", "http://www.connect", "http://www.datacom", "http://www.prosimu", "http://www.webdat", "http://www.cellular.c", "http://w2.siemens.de", "http://cellular.co.za/a", "http://cua.netholic.co", "http://planetfone.com", "http://www.cellular.c", "http://home.sol.no/~", "http://www.ammizm", "http://www.abo.fi/~", "http://www.fmi.mh.s", "http://cellular.co.za/a", "http://fourworld.comp", "http://w2.siemens.de", "http://www.aworld.r", "http://std.cpc.ku.ac.tl", "http://www.binteme", "http://www.aworld.r", "http://www.aworld.r", "http://www.aworld.r", "http://www.aworld.r". The right pane shows a detailed view of a document, including fields for HTML, Text, Title, Description, and Keywords. The HTML field contains meta-information about the document, such as "Content-Type", "Author", and "Description". The Title field contains "saada tasuta sms-teade gsm-telefonile / send sms message to any gsm ...". The Description field contains "send sms message to any gsm". The Keywords field is empty. The bottom status bar indicates "Retrieving document # 110".

ID doc	Href	Nli	SumW2
868	http://www.wireless.ee/sms.html	1	
869	http://members.tripod.lycos.nl/engelande/	29	
870	http://www.mtn.co.za/regulars/sms/	2	
871	http://www.esms.com/	19	
872	http://www.ais900.com/vas/messaging.html	12	
875	http://www.lowwww.com/html/gsmtr.html	9	
876	http://www.lowwww.com/html/taxtr.html	9	
877	http://ds.dial.pipex.com/benw/	20	
878	http://www.wcs.cern.ch/public/services/sms/sms.html	1	
879	http://fones.co.uk/options_gsm_data.htm	32	
880	http://www.colargol.idb.hist.no/~terjere/gsmtekst.htm	5	

Number of records: 51

HTML: <HTML> <HEAD> <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1"> <META HTTP-EQUIV="Author" CONTENT="Juri Zaitsev, yz@wireless.ee"> <META NAME="description" CONTENT="Send SMS message to any

Title: ... saada tasuta sms-teade gsm-telefonile / send sms message to any gsm ...

Description: send sms message to any gsm

Keywords:

Rys. 6.3. Ekran systemu CIS, podczas wykonywania zadania sprowadzenia stron relewantnych z nałożonym ekranem zawartości bazy danych zawierającej sprowadzone dokumenty.

¹ Użytkownik może ponownie wywołać etap sprowadzania stron (*Retrieve Pages*). System będzie wtedy próbował uzyskać treści wszystkich stron, których nie udało się dotychczas sprowadzić (dla danego pytania).

Dla każdego pytania, oddzielnie można ustawiać odpowiednie parametry metody (rys. 6.4), zarówno podczas tworzenia nowego pytania, jak i w dowolnym momencie w trakcie działania systemu.

Efektem końcowym działania systemu CIS jest hierarchia grup dokumentów (rys. 6.5). Dla każdego dokumentu, użytkownikowi udostępniana jest tekstowa treść strony zapamiętana w systemie. Możliwe jest także pobranie i prezentacja pełnej treści strony (wraz z elementami multimedialnymi), bezpośrednio z sieci WWW¹.

Options [X]

HTTP retrieval tenacity

Launch up to retrieval threads simultaneously

Abort threads that show no activity for more than seconds

Retry denied requests up to times

Retry incomplete files up to times

Algorithm Options

Tau (cut point) is Alpha is

Sigma (So importance) is Beta is

Number of levels in tree is Gamma is

☐ Erase the same pages

Mi (links pointing out of collection)

Min and max edge capacity is

☐ min and max capacity for each coherent graph

☐ tau and max capacity for each coherent graph

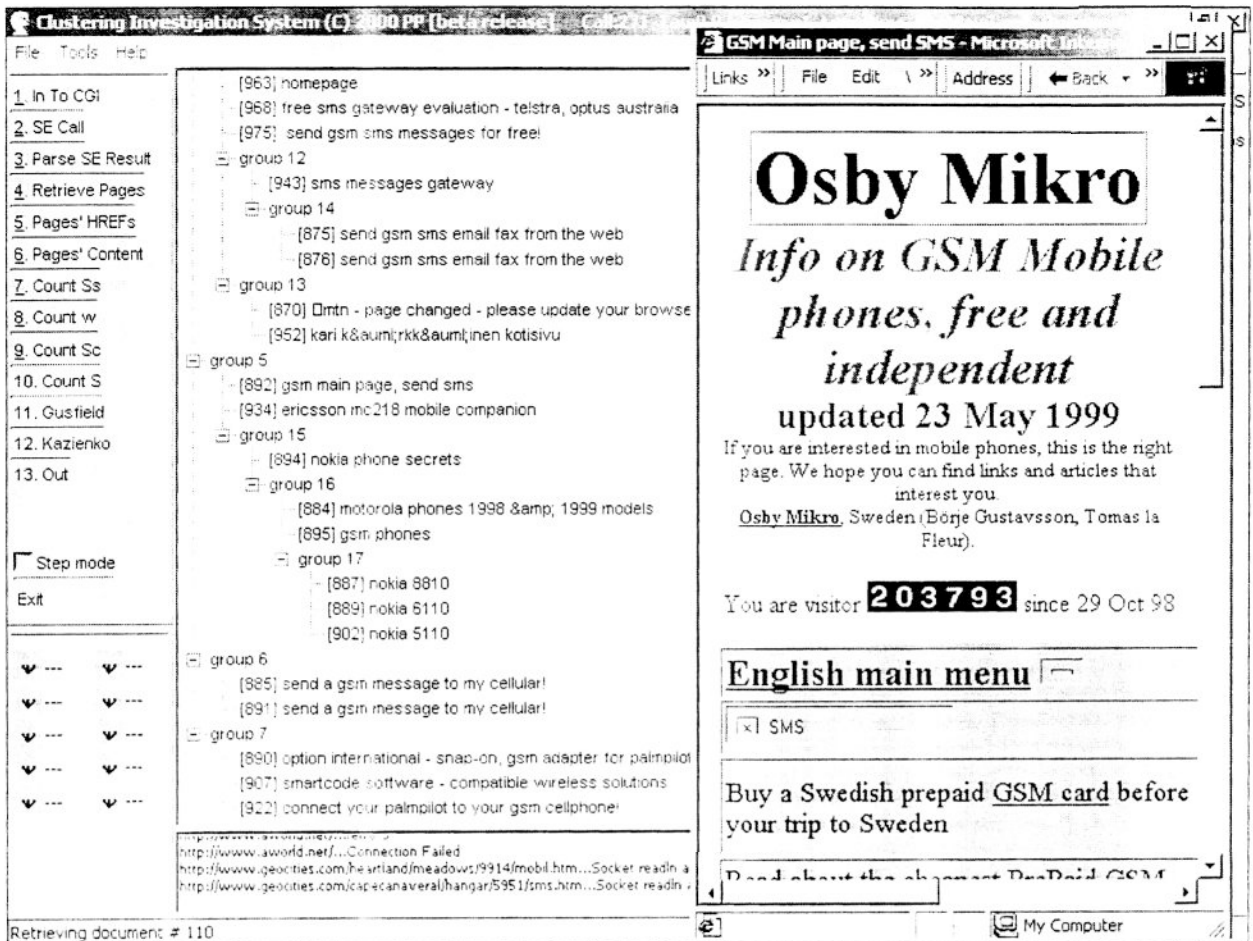
☐ min and max capacity for all graphs

☒ tau and max capacity for all graphs

OK Cancel

Rys. 6.4. Okno zmiany parametrów metody grupowania oraz parametrów systemu CIS.

¹ Jest to niezbędne w przypadku stron zawierających zbyt dużą liczbę elementów graficznych, dla których ocena dokonana jedynie na podstawie treści zawartej w ciele opisu HTML, może nie być możliwa.



Rys. 6.5. Ekran z hierarchią grup wygenerowaną przez system CIS dla pytania „gsm send”, z nałożonym ekranem prezentacji treści dokumentu nr 892.

6.3. EKSPERYMENTY ZWIĄZANE Z WYBOREM WARTOŚCI PARAMETRÓW

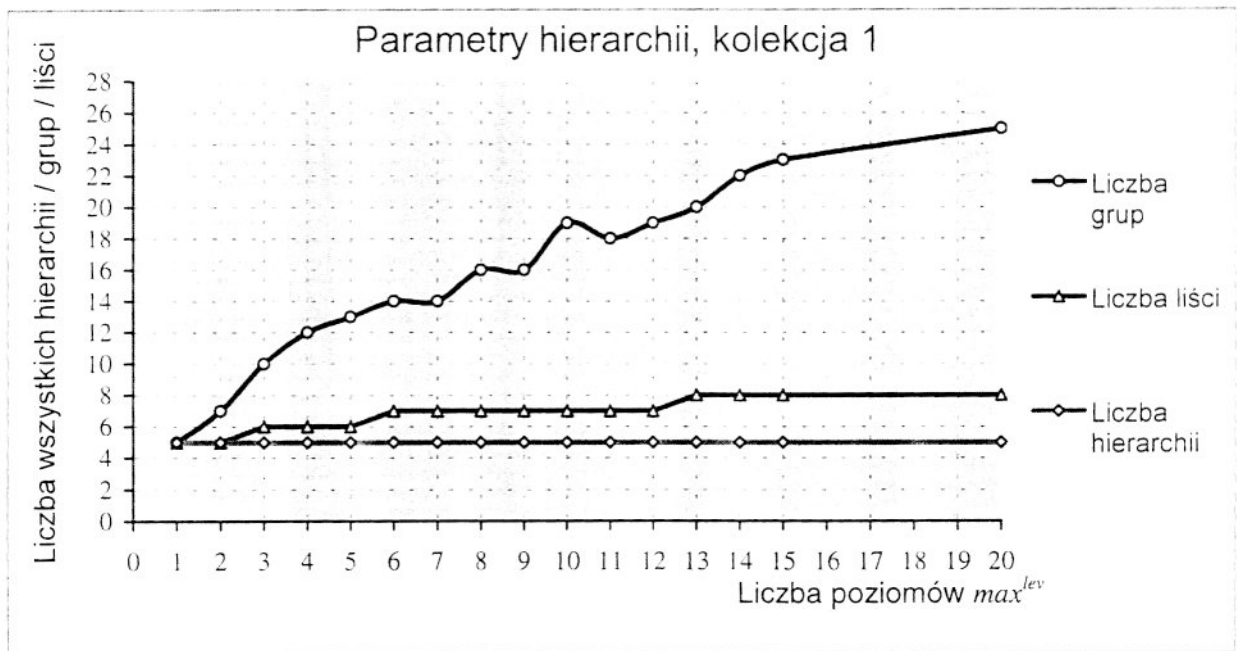
6.3.1. MAKSYMALNA LICZBA POZIOMÓW max^{lev}

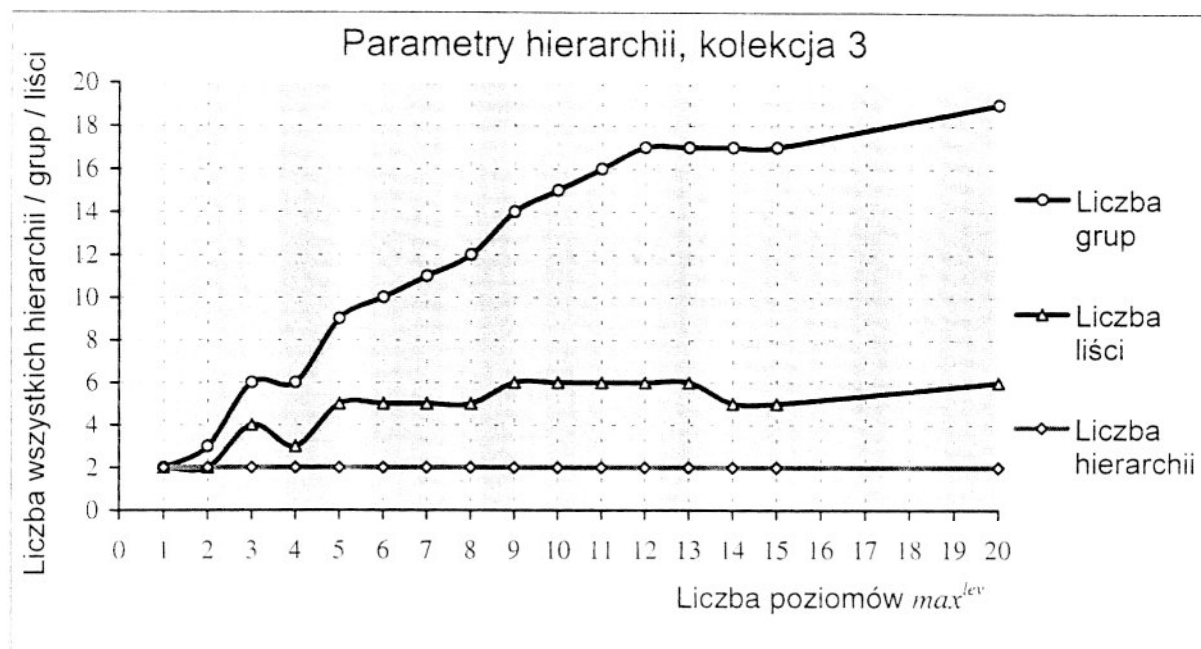
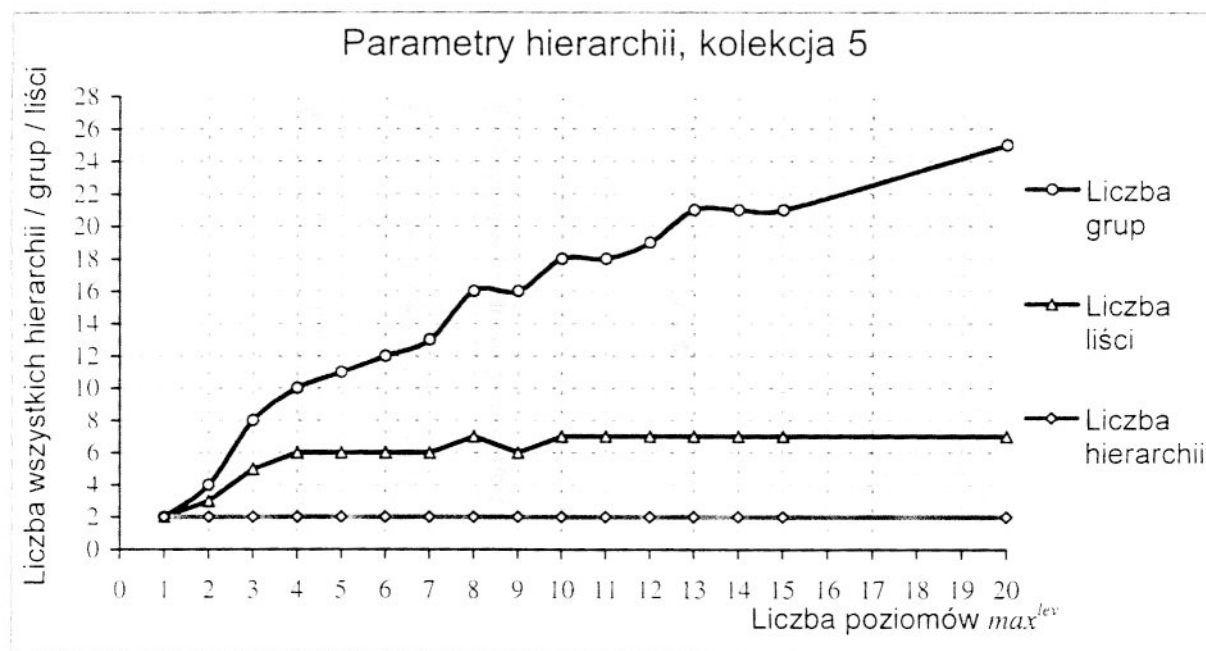
Dobór wartości maksymalnej liczby poziomów - max^{lev} zależy od wielkości kolekcji oraz zastosowania metody. W przypadku wykorzystania metody do wyszukiwania informacji i oceny wyników wyszukiwania przez użytkownika, liczba poziomów jest uzależniona od uwarunkowań psychologicznych człowieka. Użytkownik byłby zniechęcony, gdyby miał przeglądać 50 grup po średnio 2 dokumenty każda.

W konkretnej implementacji metody najlepiej, gdyby parametr max^{lev} był udostępniony użytkownikowi do zmiany, przykładowo poprzez opcje o nazwie „więcej poziomów”, „mniej poziomów”.

Zwiększanie parametru max^{lev} powoduje przede wszystkim „wydłużenie gałęzi” w hierarchii (tab. 6.1, rys. 6.6, 6.7 i 6.8). Przy dużych wzrostach max^{lev} , następuje także zwiększenie liczby gałęzi (liczby liści).

max^{lev}	Kolekcja 1			Kolekcja 3			Kolekcja 5		
	Liczba hierarchii ¹	Liczba grup ²	Liczba liści ³	Liczba hierarchii	Liczba grup	Liczba liści	Liczba hierarchii	Liczba grup	Liczba liści
1	5	5	5	2	2	2	2	2	2
2	5	7	5	2	3	2	2	4	3
3	5	10	6	2	6	4	2	8	5
4	5	12	6	2	6	3	2	10	6
5	5	13	6	2	9	5	2	11	6
6	5	14	7	2	10	5	2	12	6
7	5	14	7	2	11	5	2	13	6
8	5	16	7	2	12	5	2	16	7
9	5	16	7	2	14	6	2	16	6
10	5	19	7	2	15	6	2	18	7
11	5	18	7	2	16	6	2	18	7
12	5	19	7	2	17	6	2	19	7
13	5	20	8	2	17	6	2	21	7
14	5	22	8	2	17	5	2	21	7
15	5	23	8	2	17	5	2	21	7
20	5	25	8	2	19	6	2	25	7

Tab. 6.1. Parametry hierarchii, dla różnych wartości parametru max^{lev} .Rys. 6.6. Parametry hierarchii (bez uwzględniania dokumentów izolowanych) dla różnych wartości max^{lev} .¹ Liczba hierarchii nie uwzględnia dokumentów izolowanych² czyli wszystkich elementów wszystkich hierarchii (bez dokumentów izolowanych)³ tj. elementów wszystkich hierarchii, nie posiadających potomków (bez dokumentów izolowanych)

Rys. 6.7. Parametry hierarchii (bez uwzględniania dokumentów izolowanych) dla różnych wartości max^{lev} .Rys. 6.8. Parametry hierarchii (bez uwzględniania dokumentów izolowanych) dla różnych wartości max^{lev} .

Zmiana wartości max^{lev} nie wpływa ani na liczbę hierarchii, ani na liczbę dokumentów izolowanych.

W eksperymentacji polegających na ocenie przez ekspertów 25 kolekcji przyjęto $max^{lev} = 4$.

6.3.2. PRÓG ISTOTNOŚCI PODOBIENSTWA τ

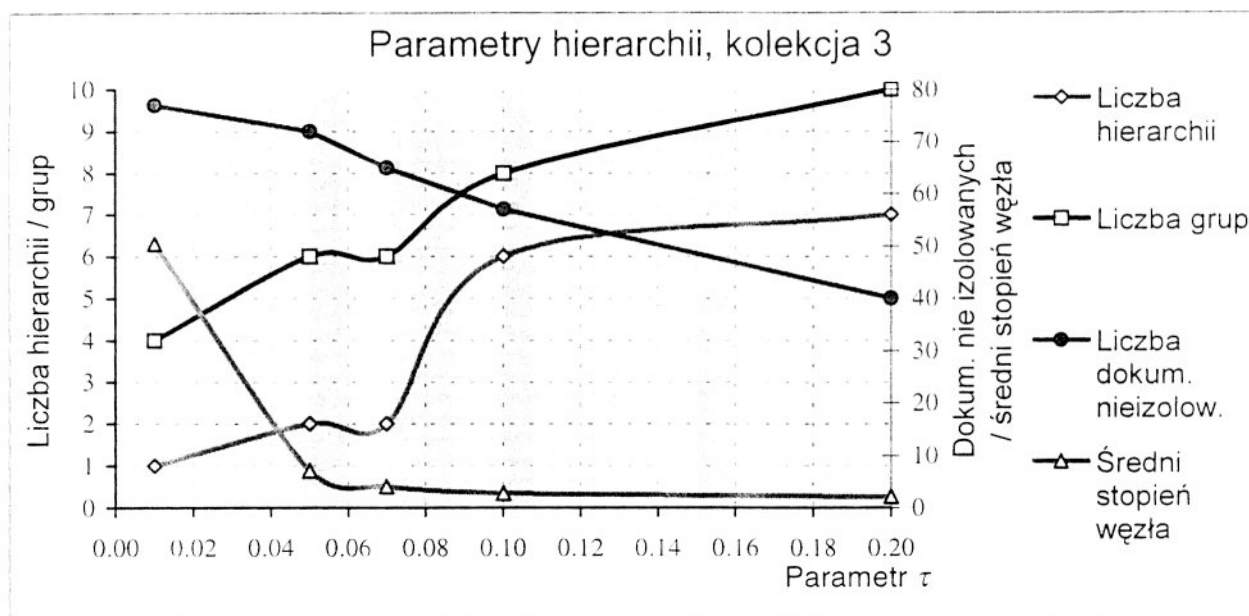
Próg istotności podobieństwa τ służy do zabezpieczenia się przed istnieniem w grafie ważonym krawędzi, które byłyby odzwierciedleniem bardzo małego podobieństwa między dokumentami. Wpływ parametru τ na postać i jakość hierarchii zbadano dla dwóch kolekcji: 3 i 4, przy pięciu różnych wartościach τ .

Zwiększanie τ prowadzi do zmniejszenia ogólnej liczby krawędzi w grafie. Jest to oczywiste, gdyż podobieństwo kompleksowe dla większej liczby par dokumentów nie przekracza zadanego

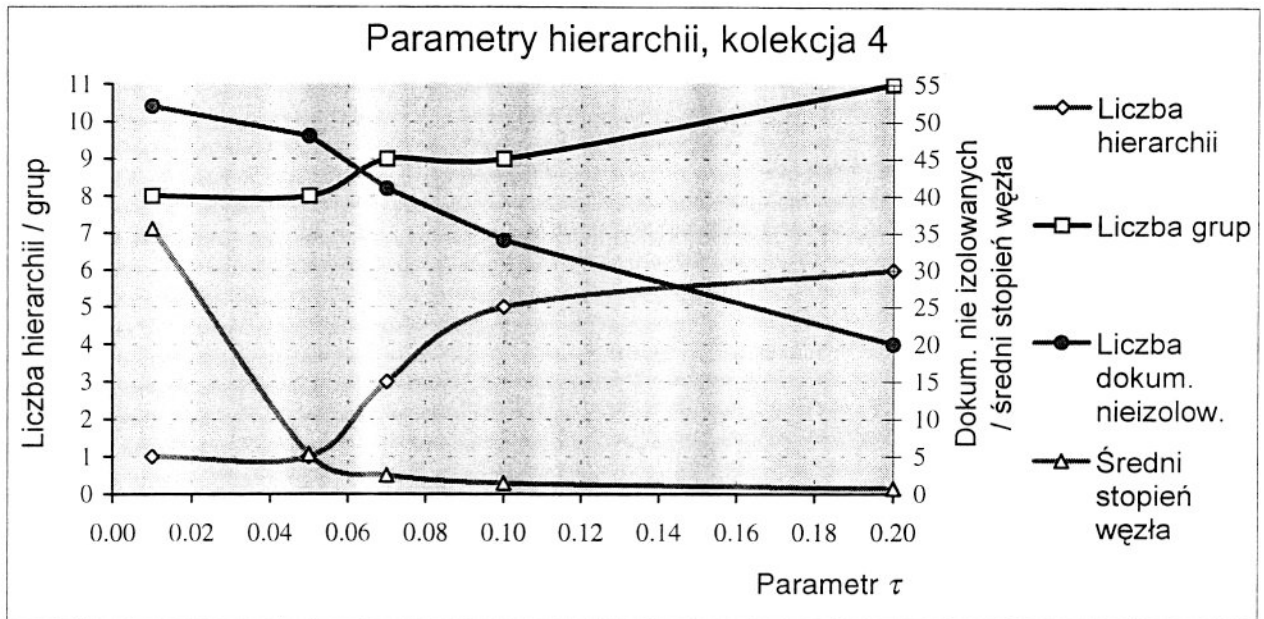
progu. Spada więc średni stopień węzła i rośnie liczba dokumentów izolowanych (tab. 6.2, rys. 6.9, 6.10). Rośnie także wtedy liczba hierarchii, gdyż z powodu mniejszej liczby krawędzi wydzielanych jest więcej składowych (każdej składowej odpowiada jedna hierarchia).

Nr kolekcji	τ	N	Liczba krawędzi grafu ważonego	Liczba hierarchii	Liczba grup (bez dok. izolowanych)	Liczba dok. nieizolowanych	Liczba dok. izolowanych	Średni stopień węzła w grafie	Średni stopień węzła, bez dok. izolowanych
3	0.01	78	1969	1	4	77	1	50.49	51.14
3	0.05	78	277	2	6	72	6	7.10	7.69
3	0.07	78	160	2	6	65	13	4.10	4.92
3	0.10	78	109	6	8	57	21	2.79	3.82
3	0.20	78	84	7	10	40	38	2.15	4.20
4	0.01	52	925	1	8	52	0	35.58	35.58
4	0.05	52	138	1	8	48	4	5.31	5.75
4	0.07	52	66	3	9	41	11	2.54	3.22
4	0.10	52	38	5	9	34	18	1.46	2.24
4	0.20	52	19	6	11	20	32	0.73	1.90

Tab. 6.2. Parametry hierarchii, dla różnych wartości progu τ .



Rys. 6.9. Zależność postaci hierarchii od parametru τ .

Rys. 6.10. Zależność postaci hierarchii od parametru τ .

Najgorsze oceny dokonane przez eksperta uzyskały hierarchie wygenerowane dla bardzo małych wartości τ (0.01) oraz stosunkowo dużych (0.20) – tab. 6.3, 6.4 oraz rys. 6.11, 6.12. Dotyczyło to prawie wszystkich parametrów tzn. $\%DNP$, $\%NG$, $\acute{S}PDP^1$, $\acute{S}PDK$. Wzrost τ , w 3 kolekcji powodował także (od wartości 0.10) zwiększanie $\%ZG$, co było związane z koniecznością przenosin jednoelementowych hierarchii, do innych.

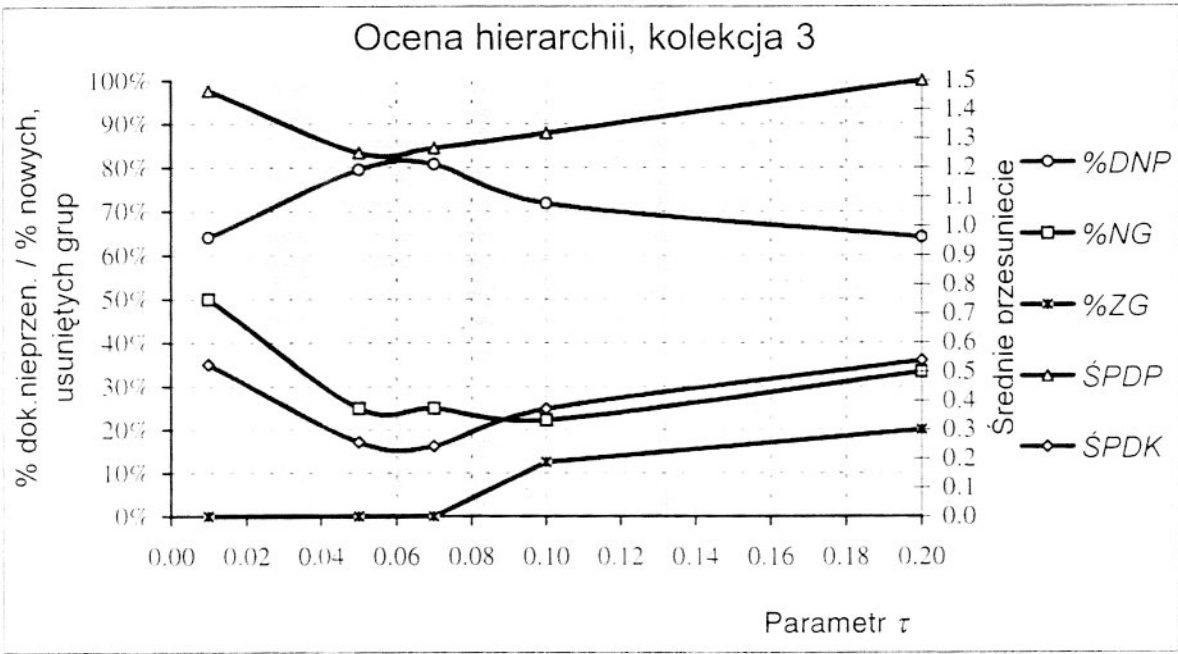
Nr kolekcji	τ	Przeniesienia do grup szczegółowszych (w dół)		Przeniesienia do grup ogólniejszych (w górę)		Przeniesienia do innych gałęzi lub innych hierarchii		Przeniesienia do grupy dokumentów izolowanych		Przeniesienia z grupy dokum. izolowanych	
		L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.
3	0.01	13	15	1	1	9	18	5	7	0	0
3	0.05	6	6	2	2	3	6	3	4	2	2
3	0.07	6	6	1	1	3	6	3	4	2	2
3	0.10	5	5	0	0	5	10	4	4	8	10
3	0.20	7	7	0	0	7	18	0	0	14	17
4	0.01	4	7	6	9	0	0	4	4	0	0
4	0.05	0	0	1	1	0	0	7	7	2	2
4	0.07	1	2	0	0	0	0	4	4	2	2
4	0.10	0	0	0	0	0	0	1	1	7	8
4	0.20	0	0	0	0	0	0	0	0	11	11

Tab. 6.3. Ocena hierarchii, dla różnych wartości parametru τ .

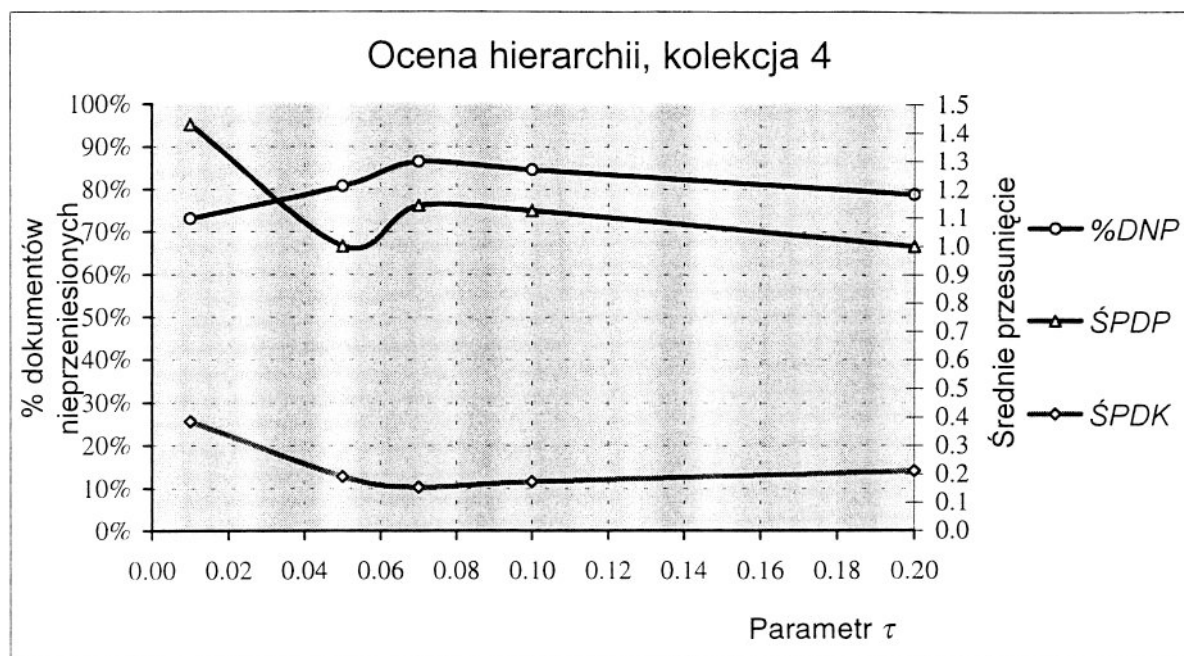
¹ Bardzo mała wartość $\acute{S}PDP=1$, dla kolekcji 4 i $\tau=0.20$, wynika z tego, że wszystkie przesunięcia związane były z przeniesieniami dokumentów izolowanych do korzeni czterech hierarchii.

Nr ko- lekcji	τ	NG	ZG	DP	WP	%DNP	%NG	%ZG	$\acute{S}PDP$	$\acute{S}PDK$
3	0.01	4	0	28	41	64.1%	50.0%	0.0%	1.46	0.53
3	0.05	2	0	16	20	79.5%	25.0%	0.0%	1.25	0.26
3	0.07	2	0	15	19	80.8%	25.0%	0.0%	1.27	0.24
3	0.10	2	1	22	29	71.8%	22.2%	12.5%	1.32	0.37
3	0.20	4	2	28	42	64.1%	33.3%	20.0%	1.50	0.54
4	0.01	0	0	14	20	73.1%	0.0%	0.0%	1.43	0.38
4	0.05	0	0	10	10	80.8%	0.0%	0.0%	1.00	0.19
4	0.07	0	0	7	8	86.5%	0.0%	0.0%	1.14	0.15
4	0.10	0	0	8	9	84.6%	0.0%	0.0%	1.13	0.17
4	0.20	0	0	11	11	78.8%	0.0%	0.0%	1.00	0.21

Tab. 6.4. Parametry zmian wynikające z oceny hierarchii, dla różnych wartości parametru τ .



Rys. 6.11. Ocena hierarchii zależna od parametru τ .

Rys. 6.12. Ocena hierarchii zależna od parametru τ .

Na podstawie powyższych badań, w eksperymentach polegających na ocenie hierarchii utworzonych dla 25 kolekcji przyjęto wartość $\tau=0.07$.

6.3.3. PARAMETR μ REGULUJĄCY ZNACZENIE ODSYŁACZY POŚREDNICH

Eksperyment przeprowadzono dla kolekcji 7. Wartości pozostałych parametrów były stałe i wynosiły: $\tau=0.15$, $\max^{lev}=4$, $\alpha=\beta=\gamma=0$. Kolekcja 7 charakteryzowała się stosunkowo dużą liczbą pośrednich odsyłaczy cytowania¹, a także różnorodnością tematyki poszczególnych dokumentów.

Kolekcja 7 powstała z odpowiedzi na pytanie „TREC”. Jest to skrót, dla którego w sieci WWW, istnieją różne rozwinięcia:

- a) Text REtrieval Conference
- b) Texas Real Estate Commission
- c) Training Resources for the Environmental Community
- d) C-TREC – centrum edukacyjne w Huston
- e) Transit Risk Environmental Control
- f) The Transportation Research and Education Center at Georgia Tech.
- g) “Trends in Electronic Commerce 98” Conference in Hamburg
- h) TREC Computers' Cyber Café
- i) The Real Estate Cafe
- j) Le TREC (Technique de Randonnée Équestre de Compétition)
- k) Timing Resource Electrophoresis Controller
- l) Testing, Research Support & Evaluation Center
- m) Tenderloin Reflection and Education Center
- n) The Rental Equipment Company

¹ Prawie $\frac{1}{3}$ stron była dokumentami źródłowymi dla pośrednich odsyłaczy cytowania.

o) Technical Resources Environmental Conservation

Większość odpowiedzi dotyczyła dwóch pierwszych znaczeń skrótu TREC.

Przyjęcie zerowej wartości α , β oraz γ służyło „osłabieniu” znaczenia zmodyfikowanego podobieństwa strukturalnego. Wybór¹ stosunkowo dużej wartości τ (0.15) miał na celu zwiększenie wpływu podobieństwa strukturalnego na podobieństwo kompleksowe¹.

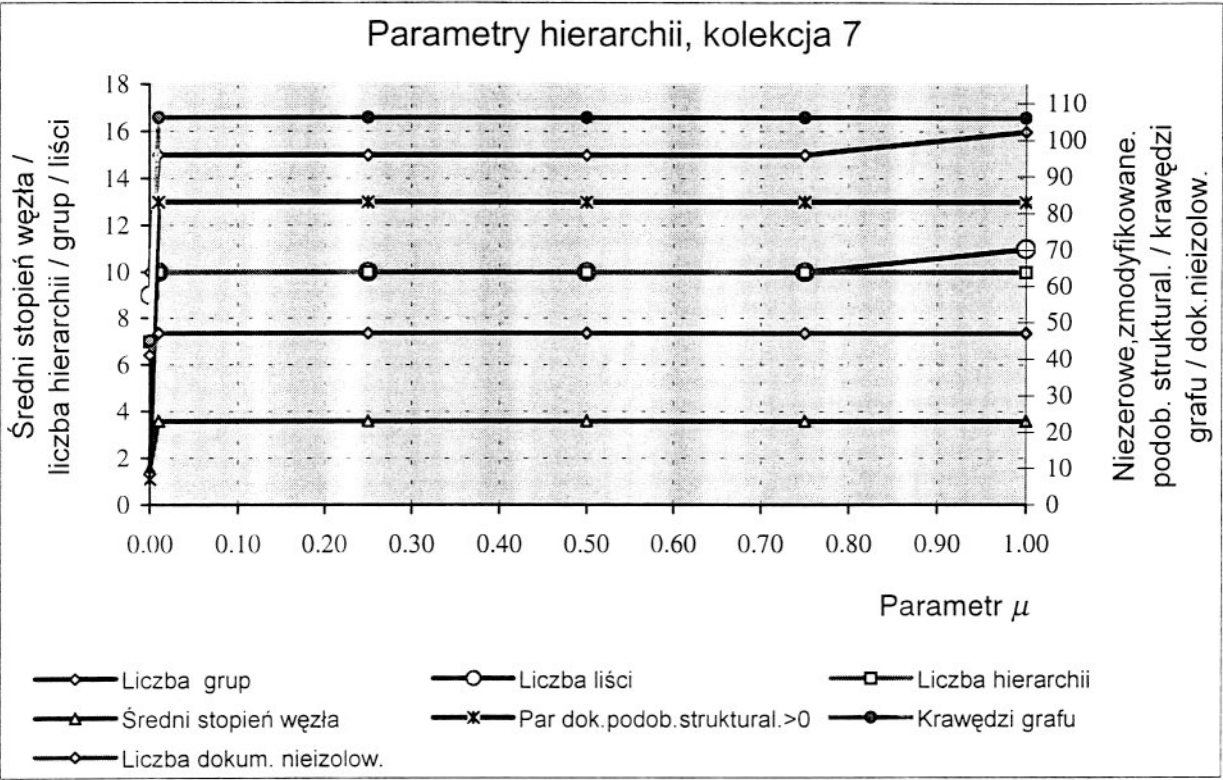
Dla $\mu=0$ graf ważony posiadał mało krawędzi, średni stopień wężła był mały: jedynie 1.53. Nadanie nieznacznej (0.01) wartości parametrowi μ , spowodowało utworzenie 2.4 razy więcej krawędzi (wzrost z 46 do 106). Średni stopień wężła zwiększył się do 3.59 (tab. 6.5, rys. 6.13). Wzrosła jednocześnie liczba grup (o $1/2$), przy czym grupy te były bardziej odseparowane od siebie (o 40% więcej hierarchii). Dalsze zwiększanie wartości parametru μ powodowało jedynie nieznaczne zmiany w wygenerowanych hierarchiach. Przeniesione zostawały jedynie pojedyncze dokumenty do grup niżej położonych. (zwiększenie μ , powoduje zwiększenie wartości podobieństwa kompleksowego). Dla $\mu \in [0.25, 0.75]$, zarówno postać hierarchii, jak i jej ocena (tab. 6.6, 6.7, rys. 6.14), była dla testowanych wartości taka sama. Przy $\mu=1$ w wygenerowanej hierarchii było tylko o jedną grupę więcej (wzrost jedynie o 7%), w stosunku do hierarchii, utworzonych dla $\mu=0.75$. Liczba hierarchii pozostała bez zmian.

Ocena hierarchii dla $\mu=1$ była, w ocenie eksperta, nieco gorsza ($\%DNP=78\%$, $\acute{S}PDK=0.37$), niż dla $\mu \in [0.25, 0.75]$ - $\%DNP=81.4\%$, $\acute{S}PDK=0.34$. Dla $\mu=0$ ocena hierarchii była zdecydowanie gorsza: $\%DNP=66.1\%$, $\acute{S}PDK=0.63$.

μ	N	Par dok. z $S^{cs}(d_i, d_j) > 0$	Dok. z $S^{cs}(d_i, d_j) > 0$	Liczba kra- wędzi grafu	Liczba dok. nie- izolo- wanych	Liczba dok. izolowa- nych	Liczba hie- rarchii	Liczba grup	Liczba liści	Średni stopień wężła w gra- fie	Średni stopień wężła, bez dok. izo- lowanych
0.00	59	7	8	45	41	18	7	10	9	1.53	2.20
0.01	59	83	26	106	47	12	10	15	10	3.59	4.51
0.25	59	83	26	106	47	12	10	15	10	3.59	4.51
0.50	59	83	26	106	47	12	10	15	10	3.59	4.51
0.75	59	83	26	106	47	12	10	15	10	3.59	4.51
1.00	59	83	26	106	47	12	10	16	11	3.59	4.51

Tab. 6.5. Charakterystyka hierarchii, dla różnych wartości μ .

¹ Dla $\tau=0.07$ i $\mu=0$, średni stopień wężła w grafie wynosił 4.25, a dla $\tau=0.15$ i $\mu=0$ jedynie 1.53 (tab. 6.5)



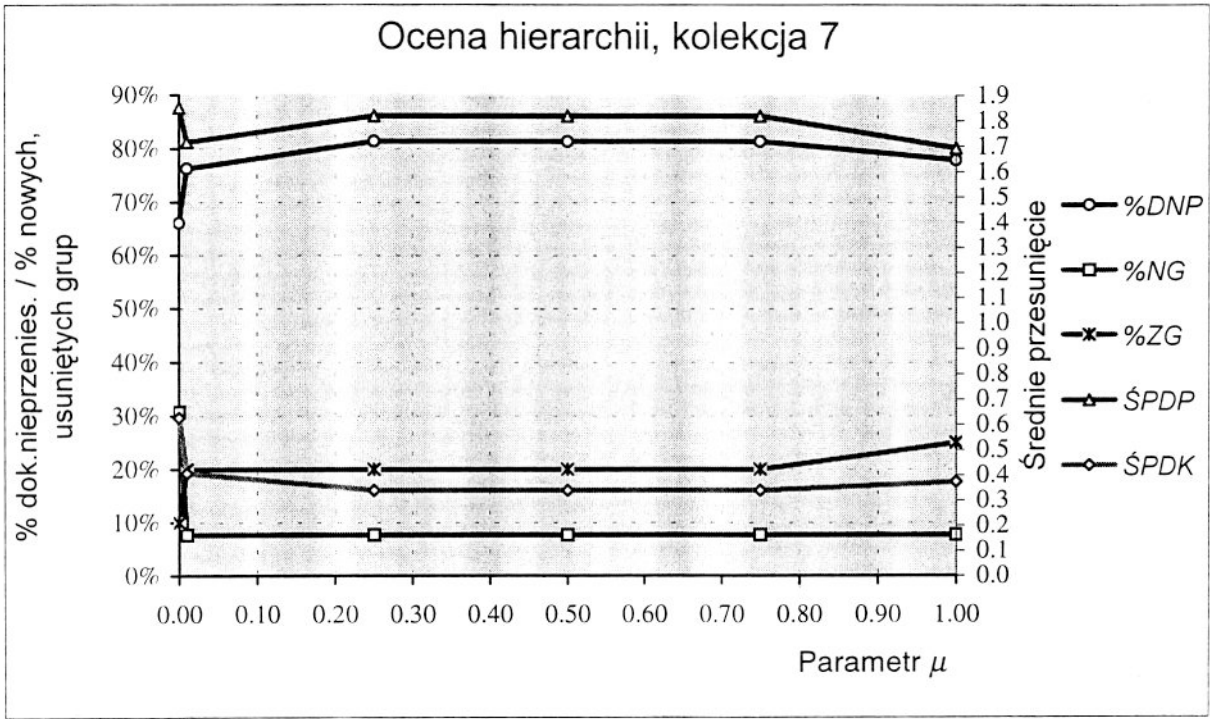
Rys. 6.13. Zależność postaci hierarchii od parametru μ .

μ	Przeniesienia do grup szczegółowszych (w dół)		Przeniesienia do grup ogólniejszych (w górę)		Przeniesienia do innych gałęzi lub innych hierarchii		Przeniesienia do grupy dokum. izolowanych		Przeniesienia z grupy dokum. izolowanych	
	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.
0.00	3	3	0	0	9	25	1	1	7	8
0.01	6	7	1	1	5	14	1	1	1	1
0.25	4	4	0	0	5	14	1	1	1	1
0.50	4	4	0	0	5	14	1	1	1	1
0.75	4	4	0	0	5	14	1	1	1	1
1.00	5	5	1	1	5	14	1	1	1	1

Tab. 6.6. Ocena hierarchii, dla różnych wartości parametru μ .

μ	NG	ZG	DP	WP	%DNP	%NG	%ZG	ŚPDP	ŚPDK
0.00	4	1	20	37	66.1%	30.8%	10.0%	1.85	0.63
0.01	1	3	14	24	76.3%	7.7%	20.0%	1.71	0.41
0.25	1	3	11	20	81.4%	7.7%	20.0%	1.82	0.34
0.50	1	3	11	20	81.4%	7.7%	20.0%	1.82	0.34
0.75	1	3	11	20	81.4%	7.7%	20.0%	1.82	0.34
1.00	1	4	13	22	78.0%	7.7%	25.0%	1.69	0.37

Tab. 6.7. Parametry zmian w hierarchiach wynikające z oceny eksperta, dla różnych wartości parametru μ .



Rys. 6.14. Ocena hierarchii zależna od parametru μ .

Zmiana wartości μ od 0.01 do 1.00 nie wpływała znacząco na strukturę i ocenę hierarchii. Można przypuszczać, że w wypadku kolekcji zawierających dużą liczbę odsyłaczy bezpośrednich ustalenie wartości μ (dla $\mu > 0$), ma jeszcze mniejsze znaczenie. Jednakże bardzo istotny wpływ na postać i ocenę hierarchii ma to czy $\mu = 0$, czy $\mu > 0$ (są lub nie są uwzględniane pośrednie odsyłacze cytowania).

W eksperymentach z 25 kolekcjami przyjęto wartość parametru μ równą 0.5.

6.3.4. PARAMETRY α , β , γ REGULUJĄCE ZNACZENIE TERMINÓW Z TYTUŁU, OPISU I SŁÓW KLUCZOWYCH

Dla parametrów α , β , γ , związanych z uwypuklaniem znaczenia wybranych elementów stron WWW, przeprowadzono eksperymenty dla kolekcji 7.

Badań związanych z parametrem α dokonano przy następujących wartościach pozostałych parametrów: $\beta = \gamma = 0$, $\tau = 0.1$, $max^{lev} = 4$. Dzięki temu, że μ nadano wartość 0, wpływ podobieństwa strukturalnego na podobieństwo kompleksowe dotyczył jedynie 7 par dokumentów.

W eksperymentach związanych z parametrem β , pozostałym parametrom nadano wartości: $\alpha = \gamma = 0$. Przy badaniach wpływu parametru γ na postać i jakość wygenerowanych hierarchii, α i β miały wartość 0. Innym parametrom nadano takie same wartości, jak przy wyznaczaniu najlepszej wartości dla α .

α	N	Dokumentów z tytułem	% dok. z tytułem	Liczba krańdziej grafu	Liczba dok. nieizolowanych	Liczba dok. izolowanych	Liczba hierarchii	Liczba grup	Liczba liści	Średni stopień wężła w grafie	Średni stopień wężła, bez dok. izolowanych
0	59	56	94.9%	86	45	14	6	10	7	2.92	3.82
5	59	56	94.9%	70	48	11	10	13	10	2.37	2.92
10	59	56	94.9%	59	49	10	10	13	9	2.00	2.41

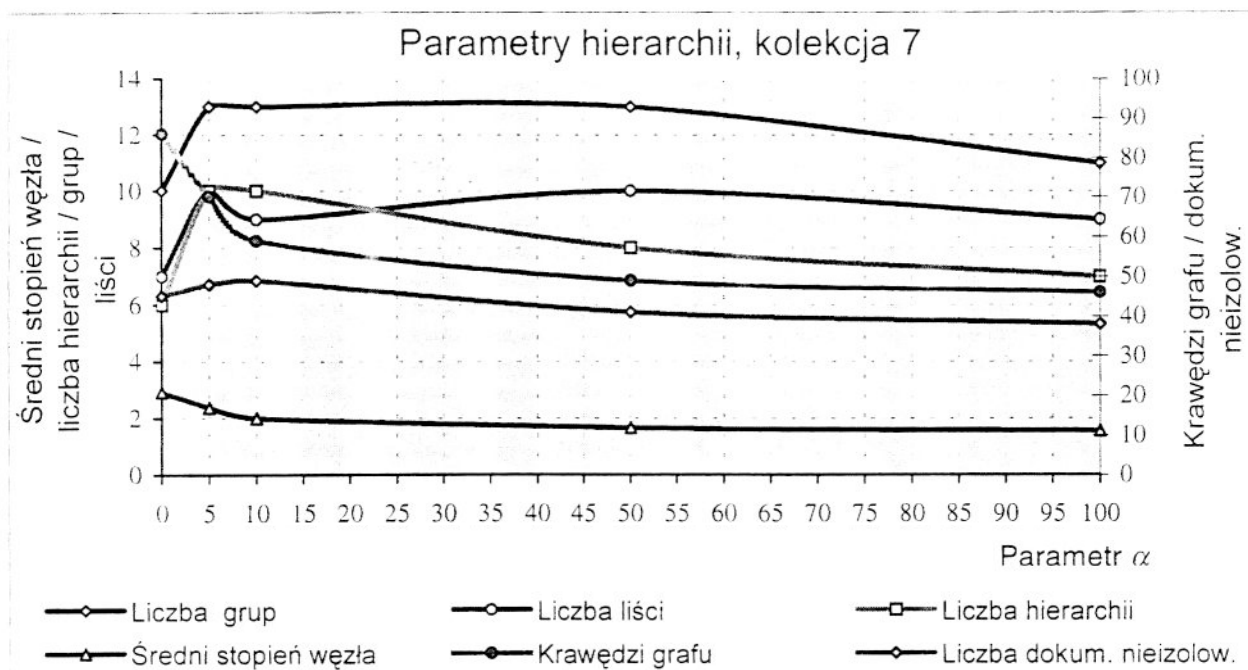
α	N	Dokumentów z tytułem	% dok. z tytułem	Liczba krawędzi grafu	Liczba dok. nieizolowanych	Liczba dok. izolowanych	Liczba hierarchii	Liczba grup	Liczba liści	Średni stopień wężła w grafie	Średni stopień wężła, bez dok. izolowanych
50	59	56	94.9%	49	41	18	8	13	10	1.66	2.39
100	59	56	94.9%	46	38	21	7	11	9	1.56	2.42

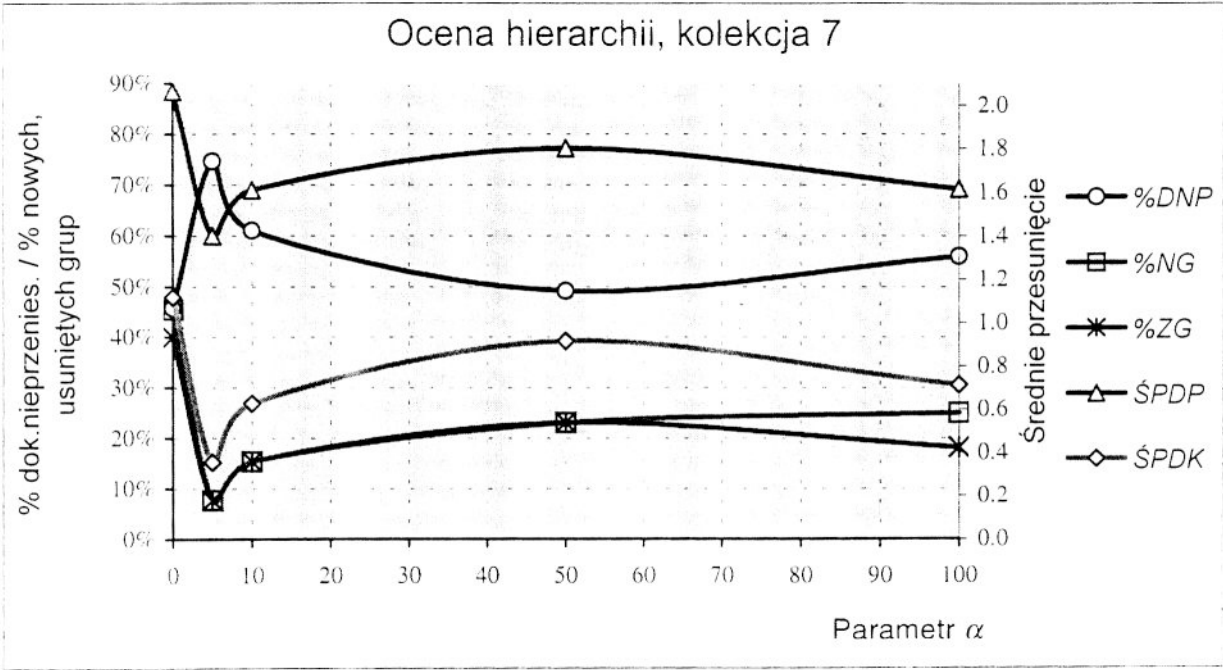
Tab. 6.8. Charakterystyka hierarchii, dla różnych wartości α .

α	Przeniesienia do grup szczegółowszych (w dół)		Przeniesienia do grup ogólniejszych (w górę)		Przeniesienia do innych gałęzi lub innych hierarchii		Przeniesienia do grupy dokum. izolowanych		Przeniesienia z grupy dokum. izolowanych	
	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.
0	2	2	0	0	15	49	9	9	6	6
5	0	0	0	0	5	10	7	7	3	4
10	0	0	0	0	10	24	8	8	5	5
50	0	0	0	0	15	38	7	8	8	8
100	0	0	0	0	8	22	7	8	11	12

Tab. 6.9. Ocena hierarchii, dla różnych wartości parametru α .

α	NG	ZG	DP	WP	%DNP	%NG	%ZG	ŚPD	ŚPDK
0	5	4	32	66	45.8%	45.5%	40.0%	2.06	1.12
5	1	1	15	21	74.6%	7.7%	7.7%	1.40	0.36
10	2	2	23	37	61.0%	15.4%	15.4%	1.61	0.63
50	3	3	30	54	49.2%	23.1%	23.1%	1.80	0.92
100	3	2	26	42	55.9%	25.0%	18.2%	1.62	0.71

Tab. 6.10. Parametry zmian w hierarchiach wynikające z oceny eksperta, dla różnych wartości parametru α .Rys. 6.15. Zależność postaci hierarchii od parametru α .



Rys. 6.16. Ocena hierarchii zależna od parametru α .

β	N	Dokumentów z opisem	% dok. z opisem	Liczba krawędzi grafu	Liczba dok. nieizolowanych	Liczba dok. izolowanych	Liczba hierarchii	Liczba grup	Liczba liści	Średni stopień wężła w grafie	Średni stopień wężła, bez dok. izolowanych
0	59	14	23.7%	86	45	14	6	10	7	2.92	3.82
2	59	14	23.7%	85	45	14	7	11	8	2.88	3.78
5	59	14	23.7%	84	46	13	8	12	9	2.85	3.65
10	59	14	23.7%	113	50	9	7	13	9	3.83	4.52
50	59	14	23.7%	519	53	6	1	4	1	17.59	19.58
100	59	14	23.7%	901	50	9	1	4	1	30.54	36.04

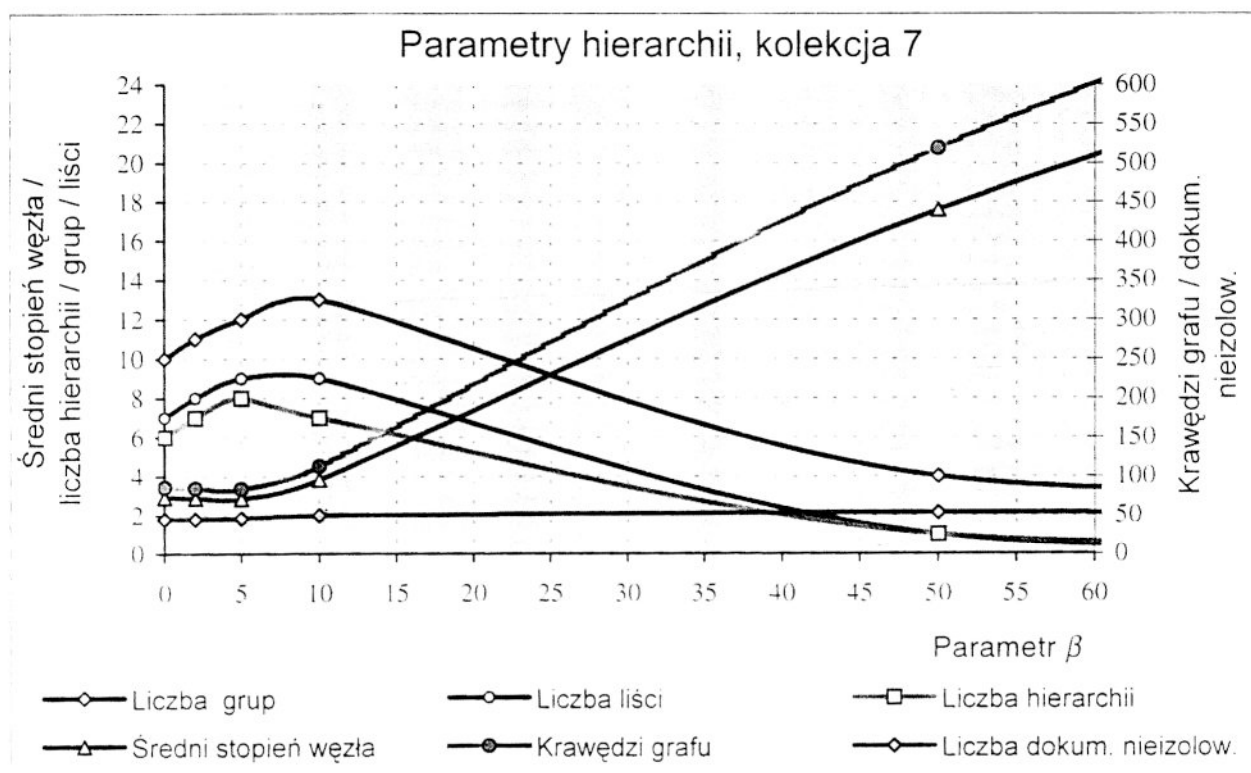
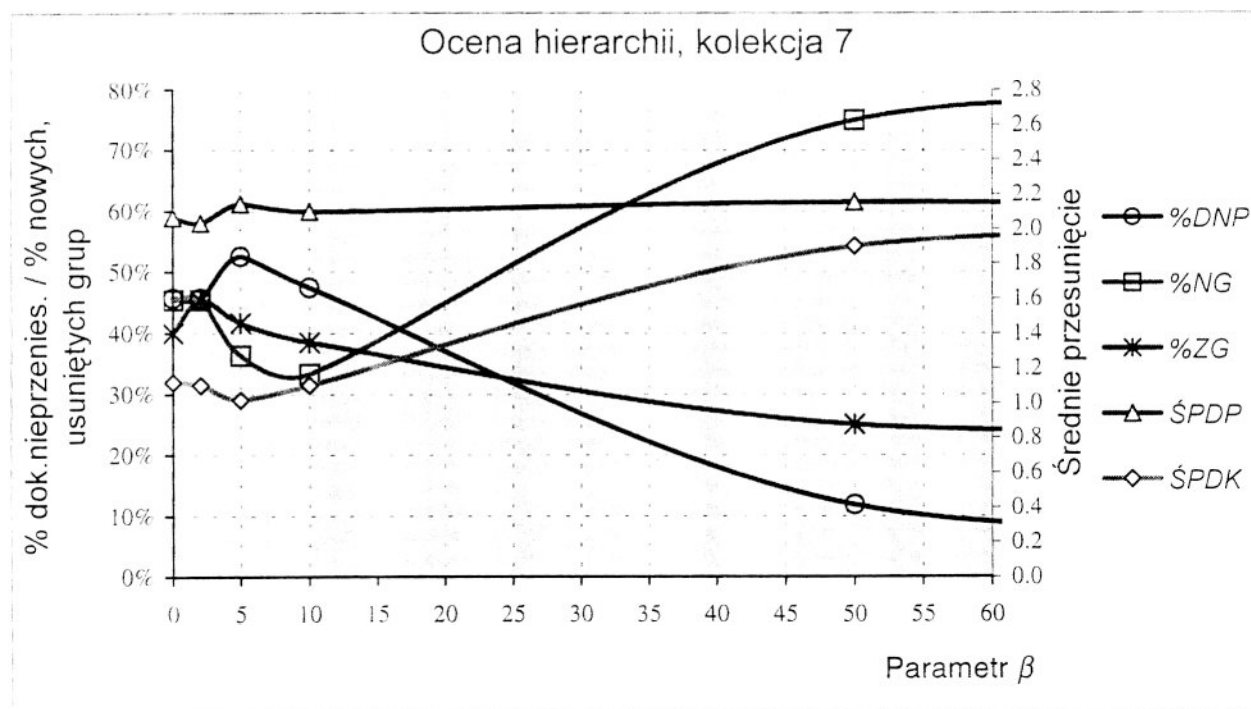
Tab. 6.11. Charakterystyka hierarchii, dla różnych wartości β .

β	Przeniesienia do grup szczegółowszych (w dół)		Przeniesienia do grup ogólniejszych (w górę)		Przeniesienia do innych gałęzi lub innych hierarchii		Przeniesienia do grupy dok. izolowanych		Przeniesienia z grupy dokum. izolowanych	
	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.
0	2	2	0	0	15	49	9	9	6	6
2	1	1	0	0	17	49	8	9	6	6
5	0	0	0	0	16	47	7	8	5	5
10	0	0	0	0	16	49	11	12	4	4
50	0	0	0	0	34	91	14	17	4	4
100	0	0	0	0	33	88	14	18	7	8

Tab. 6.12. Ocena hierarchii, dla różnych wartości parametru β .

β	NG	ZG	DP	WP	%DNP	%NG	%ZG	ŚPDP	ŚPDK
0	5	4	32	66	45.8%	45.5%	40.0%	2.06	1.12

β	NG	ZG	DP	WP	%DNP	%NG	%ZG	$\dot{S}PDP$	$\dot{S}PDK$
2	5	5	32	65	45.8%	45.5%	45.5%	2.03	1.10
5	4	5	28	60	52.5%	36.4%	41.7%	2.14	1.02
10	4	5	31	65	47.5%	33.3%	38.5%	2.10	1.10
50	9	1	52	112	11.9%	75.0%	25.0%	2.15	1.90
100	9	1	54	114	8.5%	75.0%	25.0%	2.11	1.93

Tab. 6.13. Parametry zmian w hierarchiach wynikające z oceny eksperta, dla różnych wartości parametru β .Rys. 6.17. Zależność postaci hierarchii od parametru β .Rys. 6.18. Ocena hierarchii zależna od parametru β .

γ	N	Dokumentów z opisem	% dok. z opisem	Liczba kra- wędzi grafu	Liczba dok. nie- izolowa- nych	Liczba dok. izolowa- nych	Liczba hie- rarchii	Liczba grup	Liczba liści	Średni stopień wężła w grafie	Średni stopień wężła, bez dok. izolowa-nych
0	59	16	27.1%	86	45	14	6	10	7	2.92	3.82
2	59	16	27.1%	88	46	13	6	10	7	2.98	3.83
5	59	16	27.1%	87	49	10	7	11	8	2.95	3.55
10	59	16	27.1%	111	51	8	6	11	8	3.76	4.35
50	59	16	27.1%	469	57	2	2	5	2	15.90	16.46
100	59	16	27.1%	825	54	5	2	5	7	27.97	30.56

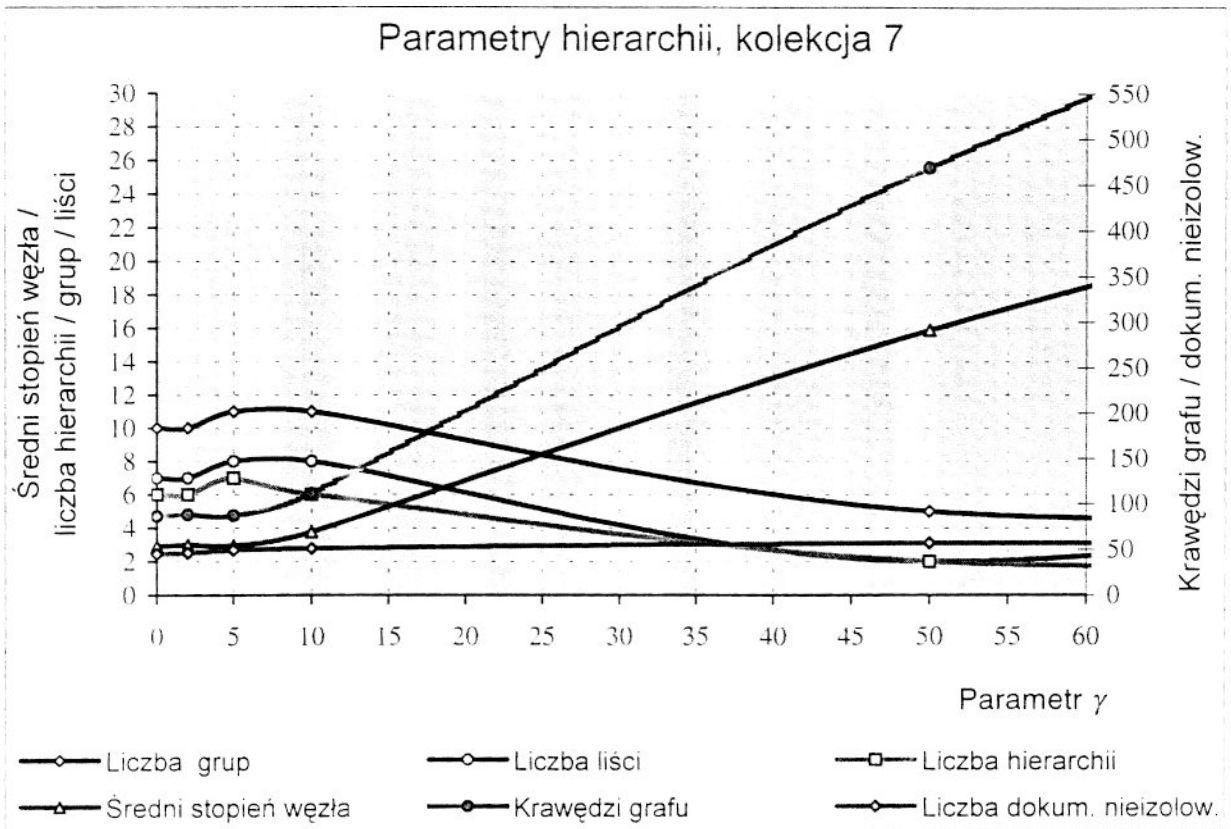
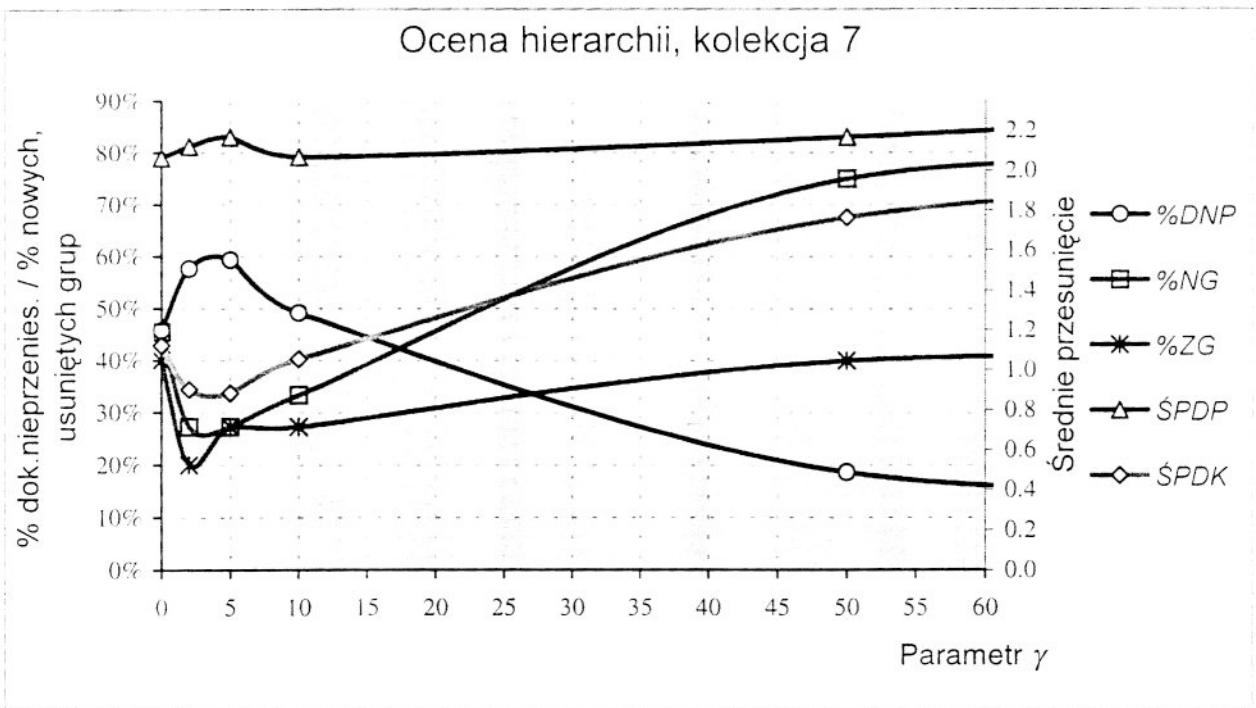
Tab. 6.14. Parametry hierarchii, dla różnych wartości γ

γ	Przeniesienia do grup szczegółowszych (w dół)		Przeniesienia do grup ogólniejszych (w górę)		Przeniesienia do innych gałęzi lub innych hierarchii		Przeniesienia do grupy dokum. izolowanych		Przeniesienia z grupy dokum. izolowanych	
	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.
0	2	2	0	0	15	49	9	9	6	6
2	0	0	0	0	14	41	6	6	5	6
5	0	0	0	0	14	41	6	6	4	5
10	0	0	0	0	16	46	10	12	4	4
50	0	0	2	3	32	85	13	15	1	1
100	0	0	5	6	30	92	12	16	3	3

Tab. 6.15. Ocena hierarchii, dla różnych wartości parametru γ

γ	NG	ZG	DP	WP	%DNP	%NG	%ZG	ŚPD _P	ŚPD _K
0	5	4	32	66	45.8%	45.5%	40.0%	2.06	1.12
2	3	2	25	53	57.6%	27.3%	20.0%	2.12	0.90
5	3	3	24	52	59.3%	27.3%	27.3%	2.17	0.88
10	4	3	30	62	49.2%	33.3%	27.3%	2.07	1.05
50	9	2	48	104	18.6%	75.0%	40.0%	2.17	1.76
100	9	2	50	117	15.3%	75.0%	40.0%	2.34	1.98

Tab. 6.16. Parametry zmian w hierarchiach wynikające z oceny eksperta, dla różnych wartości parametru γ

Rys. 6.19. Zależność postaci hierarchii od parametru γ .Rys. 6.20. Ocena hierarchii zależna od parametru γ .

Zwiększając wartości α , β , γ od 0 do 100, stwierdzono, że liczba wygenerowanych hierarchii, oraz grup rośnie by osiągnąć maksimum dla wartości 5 lub 10, a następnie spada. Podobnie zachowują się parametry związane z oceną hierarchii przez eksperta. Właściwie wszystkie parametry (%DNP, %NG, %ZG, SPDP, SPDK) osiągają najlepsze wartości dla $\alpha = \beta = \gamma = 5$.

Ogólnie stwierdzono, że nadanie parametrom α , β , γ wartości większych od zera poprawiało ocenę hierarchii. Największe pozytywne zmiany widać było w przypadku α : %DNP wzrasta z 45.8%

do 74.6%, a $\dot{S}PDK$ spada z 1.12 do 0.36. W przypadku β wskaźnik $\%DNP$ wzrastał jedynie do 52.5%, a $\dot{S}PDK$ spadał do 1.02. Dla γ było to odpowiednio 59.3% oraz 0.88. Zwiększanie α , β , γ od 5 do 100 powodowało gorsze oceny hierarchii, przy czym dla $\beta = \gamma = 100$, oceny były nawet gorsze niż dla $\beta = \gamma = 0$ (w przypadku α nie miało to miejsca). Takie wyniki były najprawdopodobniej związane z tym, że aż 95% dokumentów posiadało tytuł, a jedynie 24% opis i 27% słowa kluczowe.

W eksperymentacji polegających na ocenie przez ekspertów 25 kolekcji, parametrom α , β , γ nadano wartość 5.

6.4. OCENY HIERARCHII DOKONANE PRZEZ EKSPERTÓW

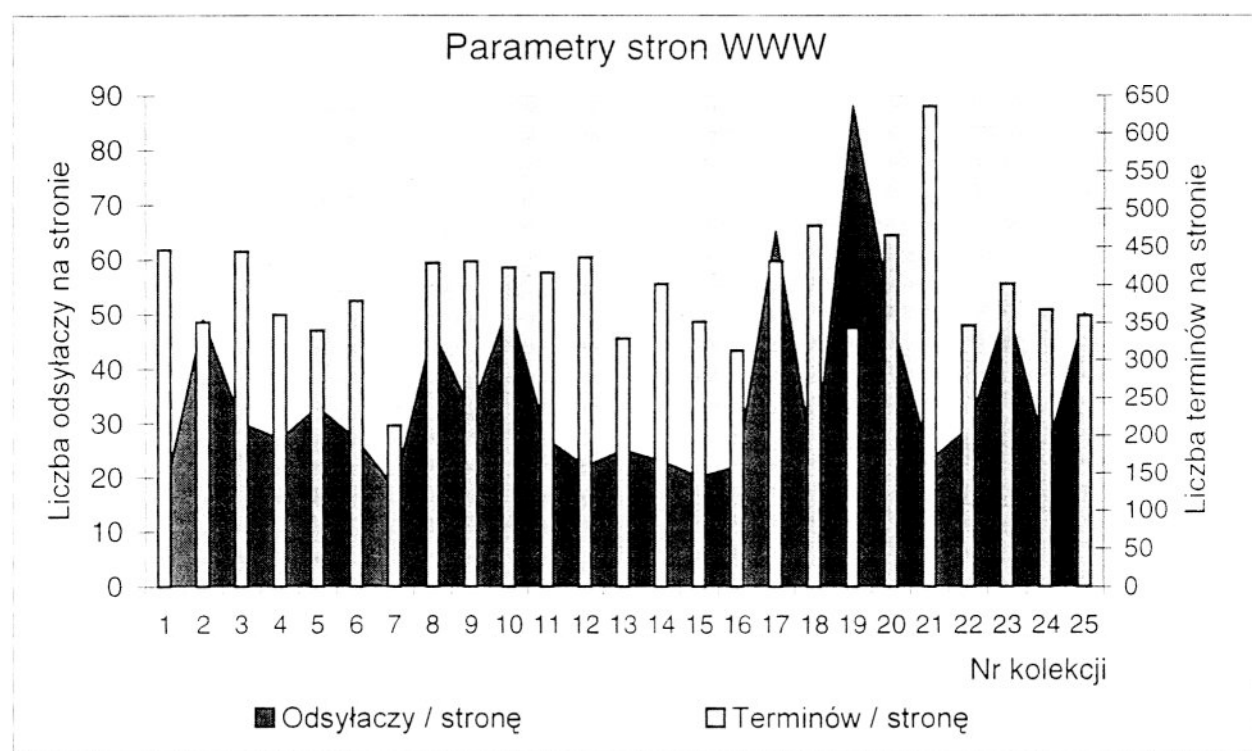
Podczas eksperymentu na 25 kolekcjach zbadano:

- Właściwości stron WWW wchodzących w skład poszczególnych kolekcji (tab. 6.17, rys. 6.21),
- Właściwości wygenerowanych przez system CIS grafów ważonych (tab. 6.18, rys. 6.22),
- Właściwości wygenerowanych hierarchii (tab. 6.19, rys. 6.23),
- Poprawność wygenerowanych hierarchii. Parametry zmian dokonanych przez ekspertów zostały zawarte w tab. 20-21 oraz rys. 6.24-25, natomiast wskaźniki ocen, których wartości wyznaczały poprawność hierarchii umieszczono w tab. 6.22 i rys. 6.26-27.

Nr kolekcji	Dok. w odpowiedzi Meta-Crawlera	N	Dok. niedostępnych		Liczba odsyłaczy <i>card(L)</i>	Unikato- wych odsy- łaczy w kolekcji	Odsyła- czy / dok.	Terminy	
			L. dok.	% odp.				<i>nt</i>	<i>nt / N</i>
1	119	80	39	32.8%	1131	821	17	4831	446
2	116	72	44	37.9%	2862	2763	49	4877	351
3	102	78	24	23.5%	1879	1702	30	4754	444
4	79	52	27	34.2%	1106	1091	27	3697	361
5	120	87	33	27.5%	2598	1510	33	3864	340
6	120	69	51	42.5%	1732	1368	27	4262	379
7	95	59	36	37.9%	879	644	18	3219	213
8	119	75	44	37.0%	3152	2800	47	4992	429
9	112	77	35	31.3%	1986	1642	32	4656	431
10	91	64	27	29.7%	2517	2191	52	4219	423
11	112	93	19	17.0%	1988	1570	27	4899	416
12	109	77	32	29.4%	1390	1185	22	3639	436
13	99	63	36	36.4%	1235	749	25	2538	329
14	105	61	44	41.9%	966	885	23	4141	401
15	82	58	24	29.3%	932	925	20	3616	351
16	120	96	24	20.0%	1659	1569	22	4901	313
17	105	82	23	21.9%	4065	4051	65	4221	431
18	120	82	38	31.7%	1522	1193	21	4440	478
19	119	87	32	26.9%	5178	5148	88	5181	343
20	83	56	27	32.5%	2604	1928	49	3691	465
21	120	87	33	27.5%	1766	1484	23	4992	635

Nr kolekcji	Dok. w odpowiedzi Meta-Crawlera	N	Dok. niedostępnych		Liczba odsyłaczy card(L)	Unikatowych odsyłaczy w kolekcji	Odsyłaczy / dok.	Terminy	
			L. dok.	% odp.				nt	nt / N
22	104	86	18	17.3%	1794	1563	29	4114	346
23	89	72	17	19.1%	3158	2762	51	4685	401
24	100	67	33	33.0%	1443	1185	24	4175	367
25	92	59	33	35.9%	2470	2465	50	4468	359
Suma	2632	1839	793		52012	45194		107072	
Średnia	105.3	73.6	31.7	30.1%	2080	1807.8	35.0	4283	398.6
Odchyl. standard.	13.6	12.4	8.8	7.3%	1034	1059.4	17.3	638	76.6

Tab. 6.17. Charakterystyka kolekcji testowych

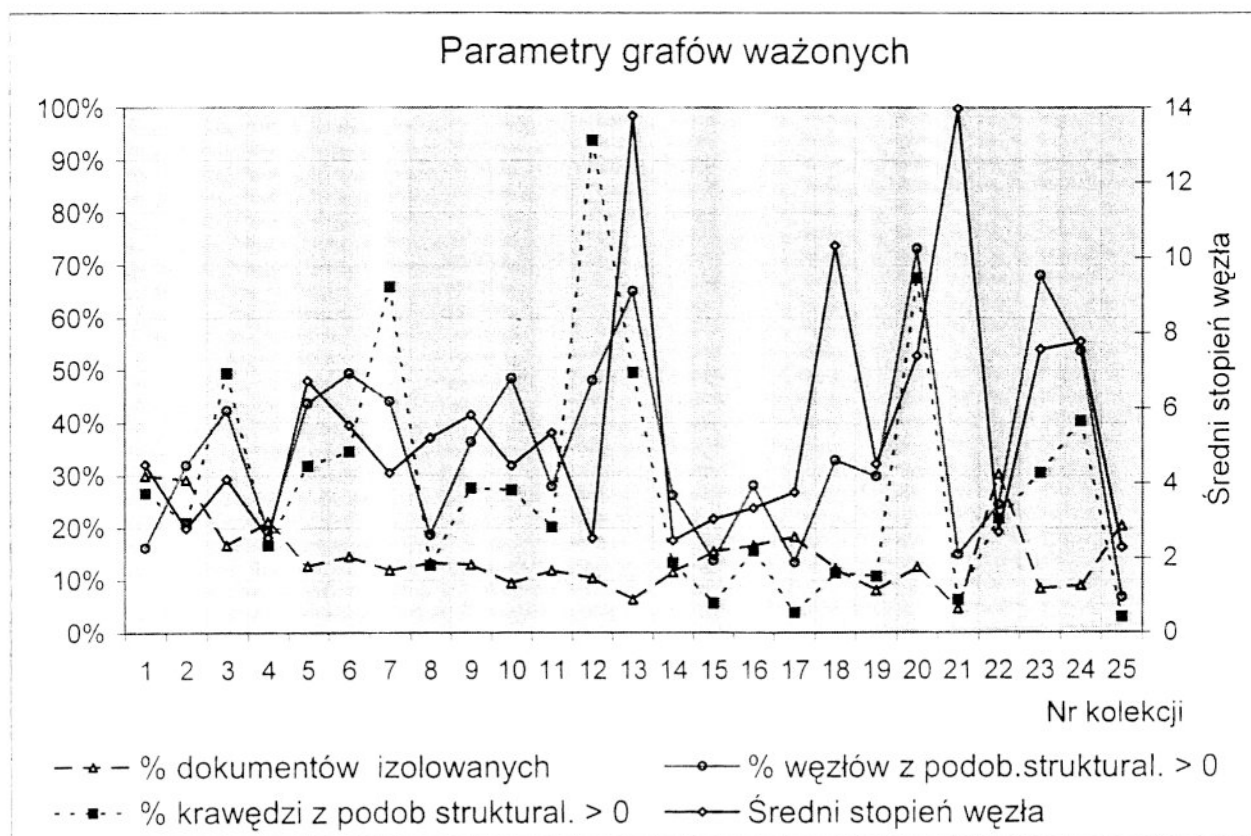


Rys. 6.21. Charakterystyka kolekcji stron WWW.

Nr kolekcji	N	Dok. nie-izolow.	Dok. izolowanych		Krawędzi card(E^G)	Średni stopień wężła		Dok. z $S^{cs}(d_i, d_j) > 0$			Krawędzi z $S^{cs}(d_i, d_j) > 0$	
			L. dok.	% z N		z izolow.	bez izolow.	L. dok.	% dok. (z izolow.)	% dok. (bez izolow.)	L. kraw.	%
1	80	56	24	30.0%	180	4.50	6.43	13	16.3%	23.2%	48	26.7%
2	72	51	21	29.2%	100	2.78	3.92	23	31.9%	45.1%	21	21.0%
3	78	65	13	16.7%	160	4.10	4.92	33	42.3%	50.8%	79	49.4%
4	52	41	11	21.2%	66	2.54	3.22	10	19.2%	24.4%	11	16.7%
5	87	76	11	12.6%	292	6.71	7.68	38	43.7%	50.0%	93	31.8%
6	69	59	10	14.5%	191	5.54	6.47	34	49.3%	57.6%	66	34.6%
7	59	52	7	11.9%	126	4.27	4.85	26	44.1%	50.0%	83	65.9%

Nr kolekcji	N	Dok. nie-izolow.	Dok. izolowanych		Krawędzi $card(E^G)$	Średni stopień wężła		Dok. z $S^{cs}(d_i, d_j) > 0$			Krawędzi z $S^{cs}(d_i, d_j) > 0$	
			L. dok	% z N		z izolow	bez izolow	L. dok.	% dok (z izolow)	% dok. (bez izolow)	L. kraw	%
8	75	65	10	13.3%	195	5.20	6.00	14	18.7%	21.5%	25	12.8%
9	77	67	10	13.0%	224	5.82	6.69	28	36.4%	41.8%	62	27.7%
10	64	58	6	9.4%	143	4.47	4.93	31	48.4%	53.4%	39	27.3%
11	93	82	11	11.8%	248	5.33	6.05	26	28.0%	31.7%	50	20.2%
12	77	69	8	10.4%	97	2.52	2.81	37	48.1%	53.6%	91	93.8%
13	63	59	4	6.3%	434	13.78	14.71	41	65.1%	69.5%	215	49.5%
14	61	54	7	11.5%	75	2.46	2.78	16	26.2%	29.6%	10	13.3%
15	58	49	9	15.5%	88	3.03	3.59	8	13.8%	16.3%	5	5.7%
16	96	80	16	16.7%	160	3.33	4.00	27	28.1%	33.8%	25	15.6%
17	82	67	15	18.3%	154	3.76	4.60	11	13.4%	16.4%	6	3.9%
18	82	72	10	12.2%	423	10.32	11.75	27	32.9%	37.5%	48	11.3%
19	87	80	7	8.0%	196	4.51	4.90	26	29.9%	32.5%	21	10.7%
20	56	49	7	12.5%	207	7.39	8.45	41	73.2%	83.7%	140	67.6%
21	87	83	4	4.6%	608	13.98	14.65	13	14.9%	15.7%	38	6.3%
22	86	60	26	30.2%	115	2.67	3.83	21	24.4%	35.0%	25	21.7%
23	72	66	6	8.3%	272	7.56	8.24	49	68.1%	74.2%	83	30.5%
24	67	61	6	9.0%	260	7.76	8.52	36	53.7%	59.0%	105	40.4%
25	59	47	12	20.3%	67	2.27	2.85	4	6.8%	8.5%	2	3.0%
Suma	1839	1568	271		5081			633			1391	
Śred.	73.56	62.7	10.8	14.7%	203.2	5.53	6.48	25.3	34.4%	40.4%	55.6	27.4%
Odch. stand.	12.4	11.7	5.7	7.0%	129	3.2	3.3	11.9	18.1%	19.6%	49.0	22.5%

Tab. 6.18. Charakterystyka grafu ważonego dla poszczególnych kolekcji.

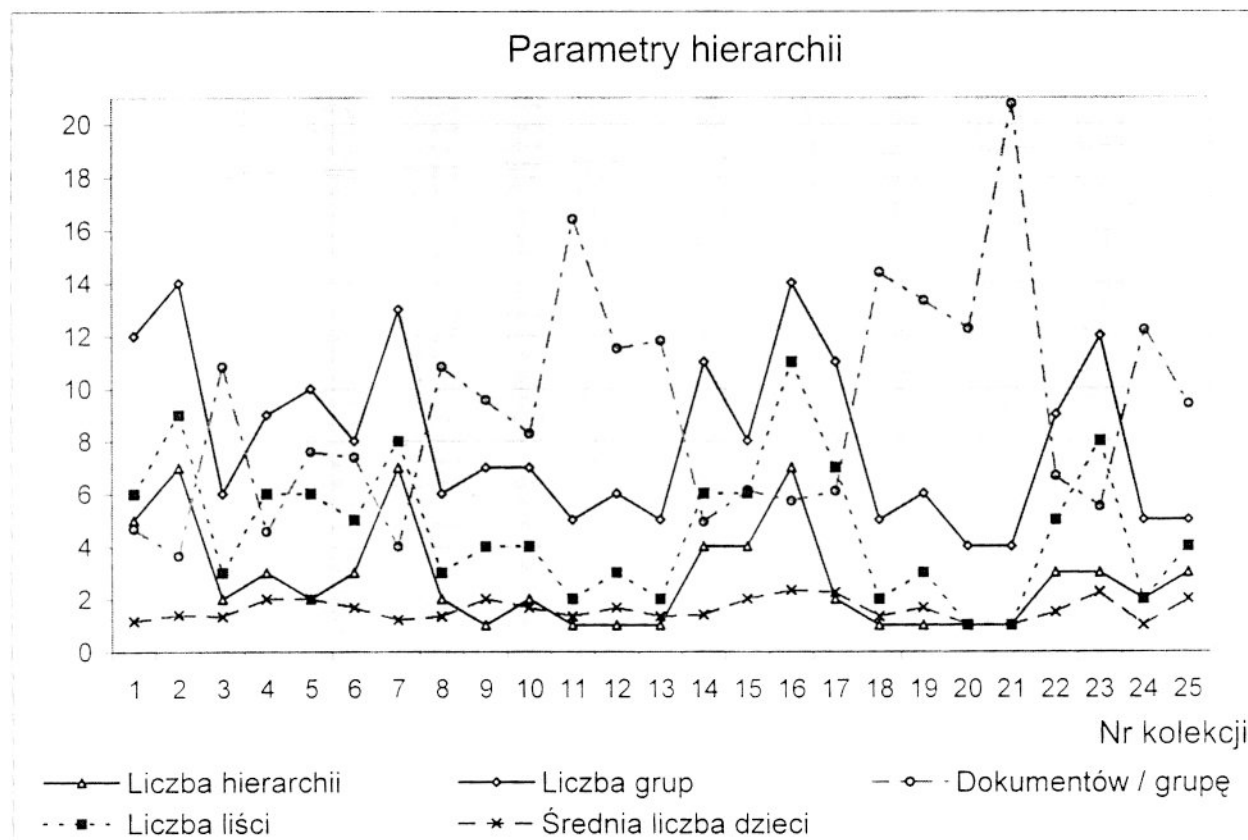


Rys. 6.22. Charakterystyka grafu ważonego.

Nr kolekcji	Liczba hierarchii	U	Liczba grup - liści	Średnia liczba dzieci	Średnio dokumentów w grupie
1	5	12	6	1.17	4.7
2	7	14	9	1.40	3.6
3	2	6	3	1.33	10.8
4	3	9	6	2.00	4.6
5	2	10	6	2.00	7.6
6	3	8	5	1.67	7.4
7	7	13	8	1.20	4.0
8	2	6	3	1.33	10.8
9	1	7	4	2.00	9.6
10	2	7	4	1.67	8.3
11	1	5	2	1.33	16.4
12	1	6	3	1.67	11.5
13	1	5	2	1.33	11.8
14	4	11	6	1.40	4.9
15	4	8	6	2.00	6.1
16	7	14	11	2.33	5.7
17	2	11	7	2.25	6.1
18	1	5	2	1.33	14.4
19	1	6	3	1.67	13.3
20	1	4	1	1.00	12.3
21	1	4	1	1.00	20.8
22	3	9	5	1.50	6.7

Nr kolekcji	Liczba hierarchii	U	Liczba grup - liści	Średnia liczba dzieci	Średnio dokumentów w grupie
23	3	12	8	2.25	5.5
24	2	5	2	1.00	12.2
25	3	5	4	2.00	9.4
Suma	69	202	117		
Średnia	2.8	8.1	4.7	1.56	7.8
Odchylenie standardowe	1.9	3.2	2.6	0.4	4.3

Tab. 6.19. Charakterystyka wygenerowanych hierarchii (bez dokumentów izolowanych).

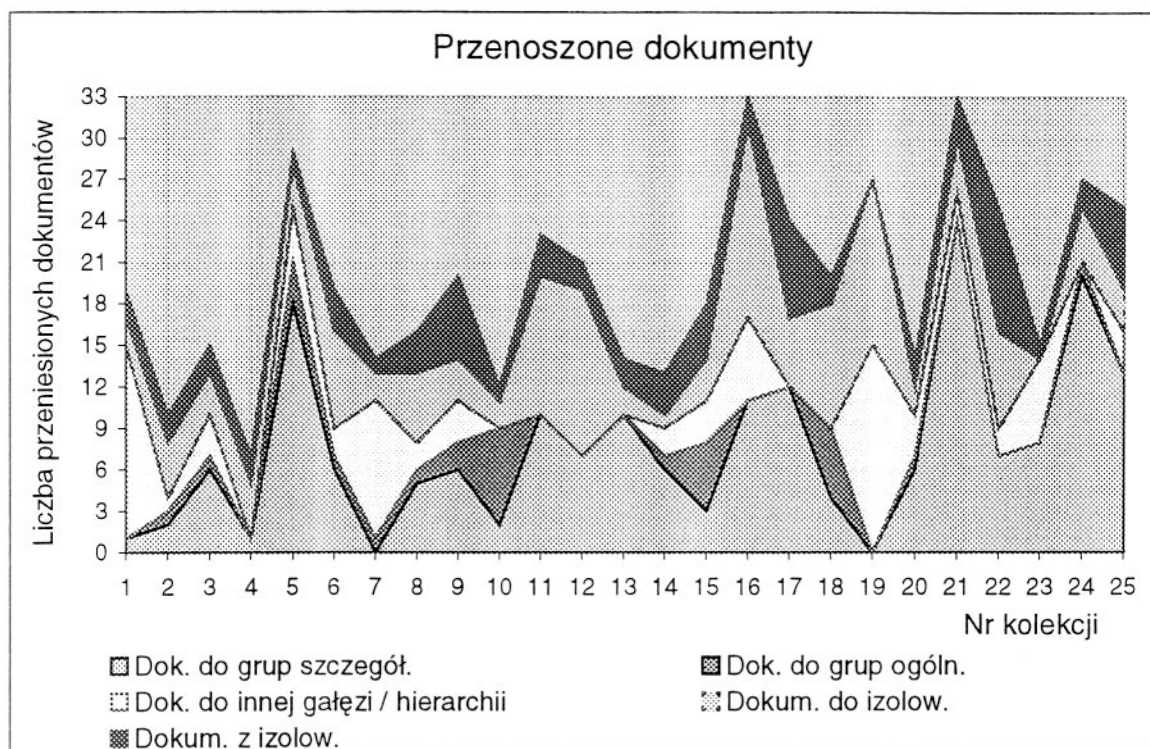


Rys. 6.23. Parametry wygenerowanych hierarchii (bez dokumentów izolowanych).

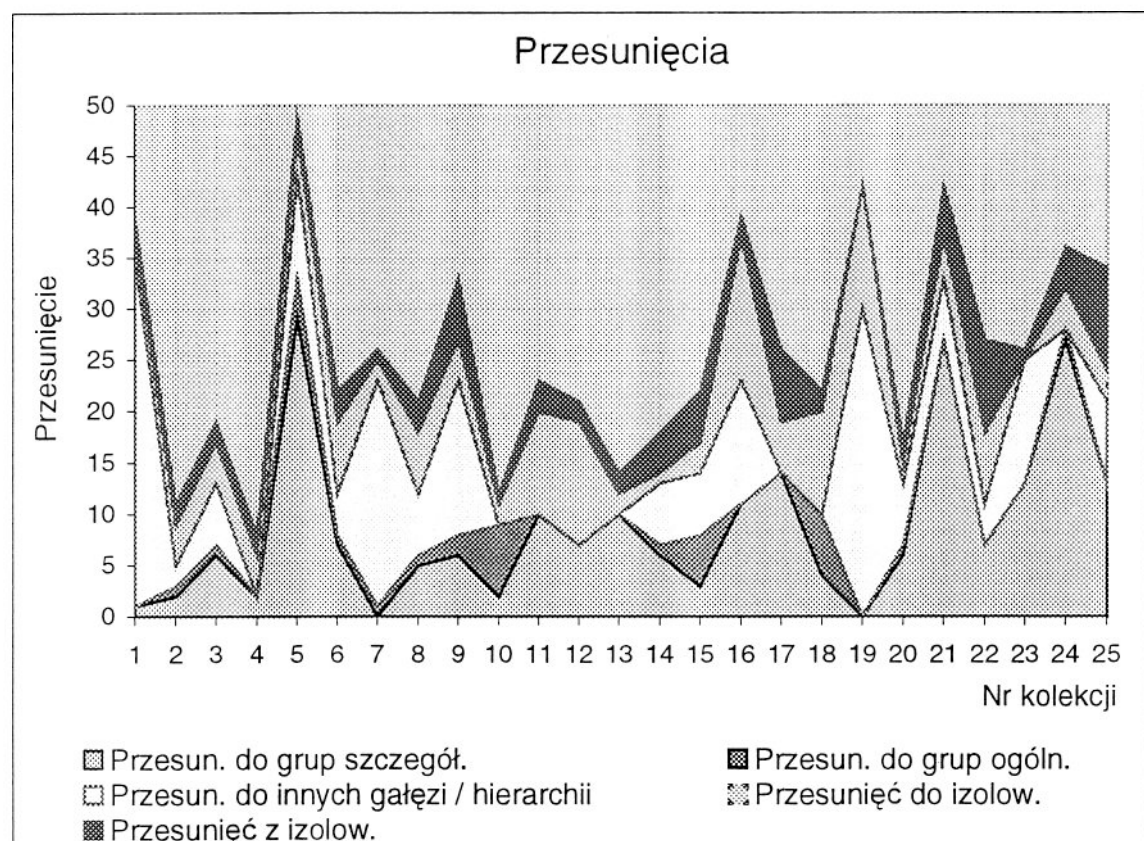
Nr kolekcji	Przeniesienia do grup szczegółowszych (w dół)		Przeniesienia do grup ogólniejszych (w górę)		Przeniesienia do innych gałęzi lub innych hierarchii		Przeniesienia do grupy dokum. izolowanych		Przeniesienia z grupy dokum. izolowanych	
	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.
1	1	1	0	0	14	34	2	2	2	2
2	2	2	1	1	1	2	4	4	2	2
3	6	6	1	1	3	6	3	4	2	2
4	1	2	0	0	0	0	4	4	2	2
5	18	29	3	4	4	10	3	4	1	2
6	6	7	1	1	2	4	7	7	3	3
7	0	0	1	1	10	22	2	2	1	1
8	5	5	1	1	2	6	5	6	3	3

Nr kolekcji	Przeniesienia do grup szczegółowszych (w dół)		Przeniesienia do grup ogólniejszych (w górę)		Przeniesienia do innych gałęzi lub innych hierarchii		Przeniesienia do grupy dokum. izolowanych		Przeniesienia z grupy dokum. izolowanych	
	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.	L. dok.	Łącznie przesun.
9	6	6	2	2	3	15	3	4	6	6
10	2	2	7	7	0	0	2	2	1	1
11	10	10	0	0	0	0	10	10	3	3
12	7	7	0	0	0	0	12	12	2	2
13	10	10	0	0	0	0	2	2	2	2
14	6	6	1	1	2	6	1	1	3	4
15	3	3	5	5	3	6	3	3	4	5
16	11	11	0	0	6	12	14	14	2	2
17	12	14	0	0	0	0	5	5	7	7
18	4	4	5	6	0	0	9	10	2	2
19	0	0	0	0	15	30	12	12	0	0
20	6	6	1	1	3	6	2	2	2	2
21	24	27	0	0	2	6	4	4	3	5
22	7	7	0	0	2	4	7	7	9	9
23	8	13	0	0	6	12	0	0	1	1
24	20	27	1	1	0	0	4	4	2	4
25	13	13	0	0	3	8	3	3	6	10
Suma	188	218	30	32	81	189	123	128	71	82
Średnia	7.5	8.7	1.2	1.3	3.2	7.6	4.9	5.1	2.8	3.3
Odchyl. standardowe	6.2	8.2	1.9	2.0	4.1	9.2	3.8	3.8	2.1	2.5
Procent z przeniesień	38.1%	33.6%	6.1%	4.9%	16.4%	29.1%	24.9%	19.7%	14.4%	12.6%
Procent kolekcji	10.2%		1.6%		4.4%		6.7%		3.9%	

Tab. 6.20. Charakterystyka przeniesień dokumentów dokonanych przez ekspertów, dla poszczególnych kolekcji



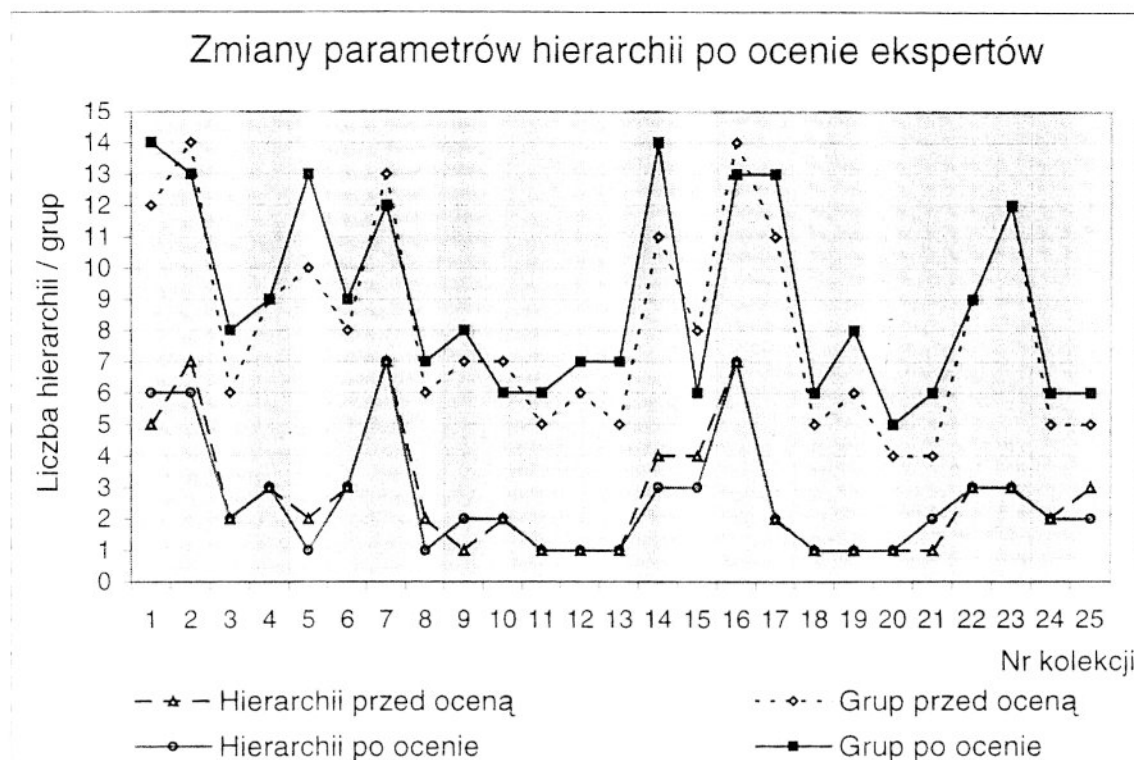
Rys. 6.24. Liczby przenoszonych dokumentów w poszczególnych rodzajach przenoszeń.



Rys. 6.25. Wartości przesunięć w poszczególnych rodzajach przenoszeń.

Nr kolekcji	Dok. izolowanych po ocenie		Grupy				Hierarchie			
	L. dok.	Przyrost	Nowe	Zlikwidowane	Przyrost	U'	Nowe	Zlikwidowane	Przyrost	Łącznie po ocenie
1	24	0	3	1	2	14	1	0	1	6
2	23	2	0	1	-1	13	0	1	-1	6
3	14	1	2	0	2	8	0	0	0	2
4	13	2	0	0	0	9	0	0	0	3
5	13	2	5	2	3	13	0	1	-1	1
6	14	4	2	1	1	9	0	0	0	3
7	8	1	2	3	-1	12	0	0	0	7
8	12	2	2	1	1	7	0	1	-1	1
9	7	-3	2	1	1	8	1	0	1	2
10	7	1	0	1	-1	6	0	0	0	2
11	18	7	1	0	1	6	0	0	0	1
12	18	10	1	0	1	7	0	0	0	1
13	4	0	2	0	2	7	0	0	0	1
14	5	-2	4	1	3	14	0	1	-1	3
15	8	-1	0	2	-2	6	0	1	-1	3
16	28	12	2	3	-1	13	0	0	0	7
17	13	-2	2	0	2	13	0	0	0	2
18	17	7	1	0	1	6	0	0	0	1
19	19	12	2	0	2	8	0	0	0	1
20	7	0	1	0	1	5	0	0	0	1
21	5	1	2	0	2	6	1	0	1	2
22	24	-2	1	1	0	9	0	0	0	3
23	5	-1	1	1	0	12	0	0	0	3
24	8	2	1	0	1	6	0	0	0	2
25	9	-3	2	1	1	6	0	1	-1	2
Suma	323	52	41	20	21	223	3	6	-3	66
Średnia	12.9	2.1	1.6	0.8	0.8	8.9	0.1	0.2	-0.1	2.6
Odchyl. Standard.	6.9	4.3	1.2	0.9	1.3	3.1	0.3	0.4	0.6	1.9

Tab. 6.21. Charakterystyka grup, hierarchii i dokumentów izolowanych dokonanych przez ekspertów.

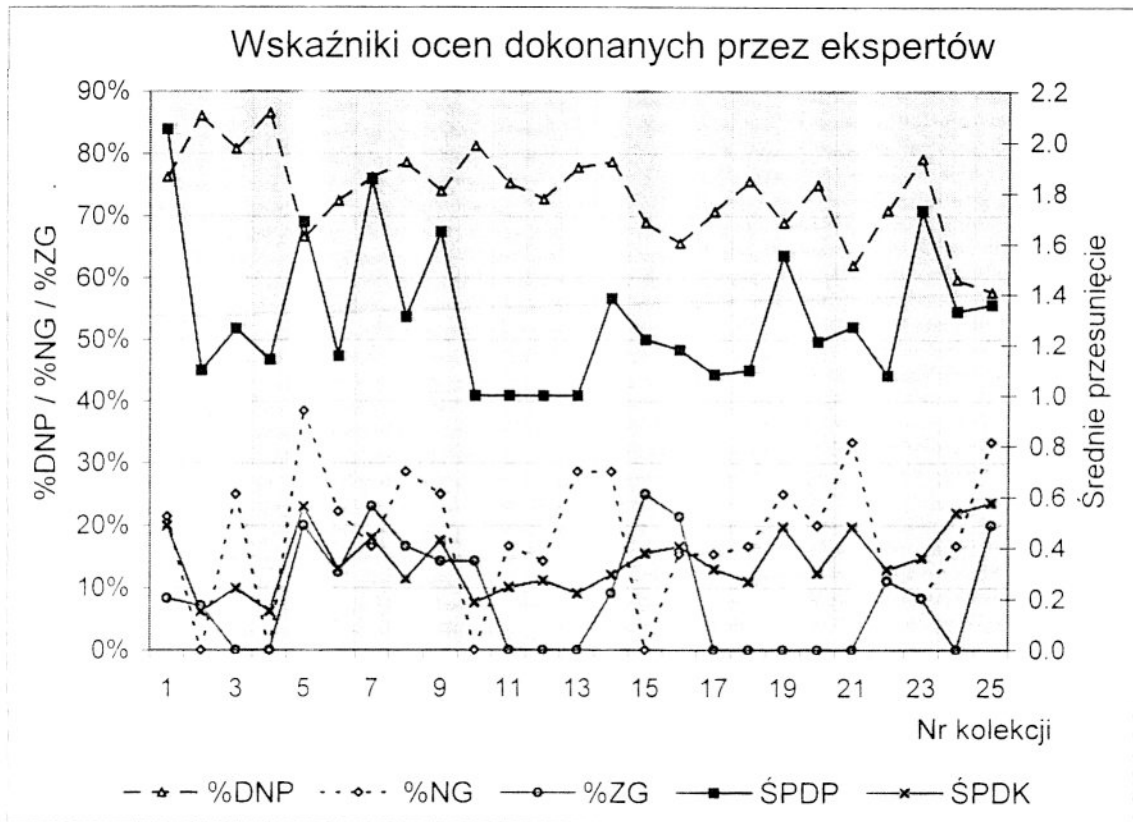


Rys. 6.26. Wpływ zmian dokonanych przez ekspertów na postać hierarchii.

Nr kolekcji	DP	WP	%DNP	%NG	%ZG	ŚPDP	ŚPDK
1	19	39	76.3%	21.4%	8.3%	2.05	0.49
2	10	11	86.1%	0.0%	7.1%	1.10	0.15
3	15	19	80.8%	25.0%	0.0%	1.27	0.24
4	7	8	86.5%	0.0%	0.0%	1.14	0.15
5	29	49	66.7%	38.5%	20.0%	1.69	0.56
6	19	22	72.5%	22.2%	12.5%	1.16	0.32
7	14	26	76.3%	16.7%	23.1%	1.86	0.44
8	16	21	78.7%	28.6%	16.7%	1.31	0.28
9	20	33	74.0%	25.0%	14.3%	1.65	0.43
10	12	12	81.3%	0.0%	14.3%	1.00	0.19
11	23	23	75.3%	16.7%	0.0%	1.00	0.25
12	21	21	72.7%	14.3%	0.0%	1.00	0.27
13	14	14	77.8%	28.6%	0.0%	1.00	0.22
14	13	18	78.7%	28.6%	9.1%	1.38	0.30
15	18	22	69.0%	0.0%	25.0%	1.22	0.38
16	33	39	65.6%	15.4%	21.4%	1.18	0.41
17	24	26	70.7%	15.4%	0.0%	1.08	0.32
18	20	22	75.6%	16.7%	0.0%	1.10	0.27
19	27	42	69.0%	25.0%	0.0%	1.56	0.48
20	14	17	75.0%	20.0%	0.0%	1.21	0.30
21	33	42	62.1%	33.3%	0.0%	1.27	0.48
22	25	27	70.9%	11.1%	11.1%	1.08	0.31
23	15	26	79.2%	8.3%	8.3%	1.73	0.36
24	27	36	59.7%	16.7%	0.0%	1.33	0.54
25	25	34	57.6%	33.3%	20.0%	1.36	0.58

Nr kolekcji	DP	WP	%DNP	%NG	%ZG	ŚPDP	ŚPDK
Suma	493	649					
Średnia	19.7 (26.8%)	26.0	73.2%	18.4%	9.9%	1.32	0.35
Odchylenie standardowe	6.9	10.8	7.4%	10.9%	8.8%	0.29	0.12

Tab. 6.22. Parametry ocen dokonanych przez ekspertów.



Rys. 6.27. Wskaźniki ocen dokonanych przez ekspertów.

Niektórzy eksperci zauważali, że może istnieć kilka różnych hierarchii grup, które byłyby przez akceptowane. Wynika to z możliwości przyjęcia różnych kryteriów, przy ustalaniu tego, co jest tematem dokumentu.

6.5. OMÓWIENIE WYNIKÓW

Średni stopień węzła w wygenerowanym przez system CIS grafie ważonym wyniósł 5.53¹ (tab. 6.18), a bez uwzględniania dokumentów izolowanych: 6.48. Oznacza to, że każdy z dokumentów nieizolowanych był połączony krawędzią ze średnio ponad 6 innymi dokumentami.

Liczba dokumentów izolowanych nie przekroczyła w żadnej kolekcji 1/3 wszystkich dokumentów (średnio 14.7%).

Liczba dokumentów, które były podobne do innych na podstawie struktury hipertekstowej ($S^{cs}(d_i, d_j) > 0$) wynosiła średnio 34.4% (40.4% bez uwzględniania dokumentów izolowanych), przy występujących wartościach od 6.8% (kolekcja 25) do 73.2% (kolekcja 20). Liczba krawędzi grafu, łączących dokumenty, dla których zmodyfikowane podobieństwo strukturalne było większe od

¹ Tylko w dwóch kolekcjach (14 i 25) średni stopień węzła był mniejszy od 2.5, a w trzech (13, 18, 21) przekraczał 10.

zera, wyniosła średnio 27.4%, przy wartościach od 3% (kolekcja 25) do 93.8% (kolekcja 12). Dokumenty o największej wartości tego współczynnika (większej od 45%, kolekcje: 3, 7, 12, 13, 20) były przy tym stosunkowo dobrze oceniane przez ekspertów (wartości $\%DNP$ wynosiły dla nich odpowiednio 80.8%, 76.3%, 72.7%, 77.8%, 75%)

Wygenerowane przez system CIS hierarchie były bardzo różnorodne. Jednakże zauważono, że duża część dokumentów była umieszczana w korzeniu hierarchii, o największej liczbie grup. W niektórych kolekcjach w korzeniu tym znajdowało się nawet 50% wszystkich dokumentów. Wynikało to z tego, że w badanych kolekcjach większa część dokumentów była związana ze sobą tematycznie, chociaż na niezbyt dużym poziomie. Szczegółowsze grupy zawierały zwykle mniejszą liczbę dokumentów. Średnia liczba dokumentów w grupie wyniosła 7.8 (tab. 6.19).

Wygenerowane hierarchie były stosunkowo mało rozbudowane wszcz, o czym świadczy wartość średniej liczby dzieci (tab. 6.19). W trzech kolekcjach (20, 21, 24) liczba dzieci nie przekraczała 1, co świadczy o tym, że w hierarchiach została wydzielona tylko jedna szczegółowsza tematyka. Ocena takich hierarchii w przypadku kolekcji 21 i 24 była zdecydowanie gorsza od średniej ($\%DNP$ wyniósł odpowiednio 62.1% i 59.7%, a $\dot{S}PDK$: 0.48 i 0.54). Dla kolekcji 20 ocena była bliska średniej ($\%DNP=75\%$, $\dot{S}PDK=0.3$).

Jednym z najważniejszych wniosków wypływających z przeprowadzonych eksperymentów jest to, że wygenerowany przez system CIS podział kolekcji stron WWW na hierarchie grup w prawie $\frac{3}{4}$ był prawidłowy, tzn. tylko 26.8% wszystkich dokumentów (tab. 6.22) eksperci zaproponowali przenieść do innych grup. Odchylenie standardowe dla $\%DNP$ było przy tym stosunkowo małe – 7.4%. Średnie przesunięcie dokumentów w kolekcji ($\dot{S}PDK$), dla wszystkich kolekcji wyniosło 0.35 (odchylenie standardowe 0.12). Innymi słowy, średnio co trzeci dokument z wszystkich 1839, został przez ekspertów przeniesiony o 1 poziom (do grupy niżej lub wyżej położonej w hierarchii rozszerzonej).

Stosunkowo niewielkie zmiany wprowadzali eksperci w odniesieniu do całych grup dokumentów. W końcowych hierarchiach (powstałych po ocenach ekspertów) średnio 18.4% grup ($\%NG$) zostało utworzonych przez ekspertów (odchylenie standardowe 10.9%), a tylko średnio 9.9% wygenerowanych przez system CIS grup ($\%ZG$) zostało przez ekspertów usuniętych. Ogólna liczba hierarchii zmieniała się wskutek oceny stosunkowo niewiele, jedynie o ± 1 , a grup w granicach $+3/-2$ (tab. 6.21).

Większa część przeniesień (tab. 6.20) dotyczyła przeniesień do grup szczegółowszych (38.1% wszystkich przenoszonych dokumentów). Tylko w 2 kolekcjach nie było żadnych przeniesień do grup szczegółowszych. Na drugim miejscu znalazły się przeniesienia do grupy dokumentów izolowanych (24.9%) Przeniesienia tego typu nie występowały tylko w jednej kolekcji. Łącznie wartość przesunięć była także największa w kategorii przeniesień do grup szczegółowszych (33.6% WP). Drugie miejsce, jednak, należało do przeniesień do innych gałęzi lub innych hierarchii (29.1% WP , przy 16.4% dokumentów przenoszonych). Wynika to z tego, że przesunięcie do innej gałęzi lub hierarchii ma wartość co najmniej 2. Najrzadziej występowały przeniesienia do grup ogólniejszych (tylko 6.1% dokumentów i 4.9% WP).

Najwyżej ocenione kolekcje ($\%DNP > 80\%$, $\dot{S}PDK < 0.2$, kolekcje 2, 4, 10) miały także małą wartość przyrostów liczby grup i hierarchii po ocenie ($-1/0$ – tab. 6.21), a także małe wartości $\%NG$ (0%) oraz $\%ZG$ (0% i 14.3%).

W najgorzej ocenionych kolekcjach ($\%DNP < 60\%$, $\dot{S}PDK > 0.5$, kolekcje 24 i 25) ponad połowa dokumentów była przeniesiona do grup szczegółowszych (dla kolekcji 24: 20 z 27 dokumentów przeniesionych, a dla kolekcji 25: 13 z 25).

Biorąc pod uwagę kolekcje najgorzej (24, 25) i najlepiej ocenione (2, 4, 10) nie można wysnuć żadnych wniosków dotyczących zależności wskaźników oceny z postacią strony (odsyłaczy / stronę, terminów / stronę), postacią grafu (odsetek dokumentów izolowanych, stopień węzła, odsetek węzłów i krawędzi z niezerowym podobieństwem strukturalnym).

Przeprowadzone badania dostarczyły także informacji o środowisku systemu WWW. Liczba dokumentów z odpowiedzi, dla których nie udało się uzyskać treści była znaczna: średnio 30.1% (tab. 6.17). Jest to o tyle zastanawiające, że metawyszukiwarka MetaCrawler sama sprawdza odpowiedzi zwracane przez wyszukiwarki, tzn. bada, czy dany odsyłacz jest prawidłowy. Brak dostępu do treści stron był spowodowane kilkoma czynnikami:

- e) przeciążeniem serwerów i sieci, w wyniku których następowało przekroczenie czasów oczekiwania (ang. *time out*) i zrywanie połączeń – najczęściej występujący problem;
- f) błędami zwracanymi przez serwery WWW (często występujący problem), spowodowanymi zwykle przez zmiany nazewnictwa kartotek i zbiorów na serwerze, a także przenosinami serwisów lub ich części,
- g) niedostępnością serwerów: wyłączenia, brak połączenia sieciowego, itp.;
- h) brakiem zarejestrowania serwera w systemie DNS (*Domain Name Service/Server*), wynikłymi przede wszystkim ze zmian w nazewnictwie serwerów (rzadko występująca sytuacja, gdyż MetaCrawler zwykle sprawdza zarejestrowanie serwera, i odrzuca odpowiedzi z serwerów niezarejestrowanych).

Średnia liczba odsyłaczy na stronie WWW wyniosła 35 (średnio dla 1839 dokumentów) i wahała się dla poszczególnych kolekcji od 17 do 88, przy odchyleniu standardowym 17.3. Stwierdzono więc, że średnia liczba odsyłaczy na stronie podlegała stosunkowo dużym wahaniom w zależności od kolekcji. Jeszcze większym wahaniom podlegała liczba odsyłaczy zawartych na poszczególnych stronach. W niektórych dokumentach liczba odsyłaczy przekraczała 300!

Średnia liczba różnych terminów w dokumencie wyniosła 398.9 (łącznie dla wszystkich kolekcji), przy wartościach od 213 do 635, w zależności od kolekcji. Innymi słowy, dokument WWW składał się z tekstu równoważnego ok. 1 stronie maszynopisu. Średnia wielkość tekstu dokumentu w mniejszym stopniu była zależna od kolekcji: odchylenie standardowe wyniosło 76.6.

7. PODSUMOWANIE

Głównym celem badań było opracowanie metody grupowania dokumentów hipertekstowych (w szczególności stron WWW). W metodzie tej miarą powiązań między dokumentami jest wartość maksymalnego przepływu w grafie, w którym węzły odpowiadają dokumentom, a krawędzie - bezpośredniemu podobieństwu między dwoma dokumentami. Wagi krawędzi wyznacza, opracowana dla potrzeb metody grupowania, funkcja podobieństwa kompleksowego. Uwzględnia ona teksty dokumentów oraz strukturę istniejących w hipertekście powiązań odsyłaczowych. W funkcji podobieństwa kompleksowego wykorzystano elementy charakterystyczne dla dokumentów WWW (tytuł, słowa kluczowe i opis będące częściami nagłówka strony).

Przed opracowaniem funkcji podobieństwa wykonano analizę systemów hipertekstowych. Dokonano zarówno ich klasyfikacji jak i klasyfikacji stron oraz odsyłaczy występujących w systemie WWW.

Informację, którą niesie ze sobą maksymalny przepływ w grafie, wykorzystano poprzez zastosowanie drzewa maksymalnych przepływów. Można z niego uzyskać wartości maksymalnego przepływu między dowolną parą węzłów grafu źródłowego. Do wyznaczenia drzewa zastosowano algorytm Gusfielda.

Drzewo maksymalnych przepływów jest wykorzystywane przez opracowany algorytm TC, który przekształca je w hierarchię grup dokumentów. Algorytm TC tworzy zawsze tę samą hierarchię grup dla danego grafu (składowej grafu), bez względu na postać drzewa utworzonego przez algorytm Gusfielda, który może generować różne postacie drzewa maksymalnych przepływów (na przykład w zależności od numeracji węzłów, czy wyboru minimalnego przekroju). Tę oraz inne własności algorytmu TC hierarchii sformułowano w postaci kilkunastu twierdzeń, lematów i wniosków.

Badania teoretyczne zostały potwierdzone przez przeprowadzone i zaprezentowane w rozdziale 6 eksperymenty, w których eksperci oceniali i przekształcali utworzone hierarchie. Do syntetycznego przedstawienia dokonanych przez ekspertów zmian oraz porównania ocen hierarchii dla różnych kolekcji stosowano wskaźniki zmian w hierarchiach: procent dokumentów nieprzesuniętych ($%DNP$), średnie przesunięcie dokumentu w kolekcji ($\bar{SPDK}$), procent nowych ($%NG$) i zlikwidowanych grup ($%ZG$). Z przeprowadzonych eksperymentów wynika, że eksperci przenosili jedynie nieco ponad $\frac{1}{4}$ wszystkich dokumentów (średnia wartość $%DNP=73.2\%$), a średnie przesunięcie dokumentu dla wszystkich dokumentów z wszystkich kolekcji wyniosło niewiele ponad $\frac{1}{3}$ (średnie $\bar{SPDK}=0.35$). Średnia wartość $%NG$ wyniosła 18.4%, a $%ZG$ - tylko 9.9%.

Można więc uznać, że wyniki eksperymentów potwierdzają poprawność hierarchii tworzonych za pomocą metody grupowania na podstawie drzewa maksymalnych przepływów.

Najważniejsze wyniki pracy to:

1. Opracowanie grafowej, hierarchicznej metody grupowania (podziału) dokumentów na podstawie drzewa maksymalnych przepływów, w której wykorzystano wartość maksymalnego przepływu jako nośnik informacji o powiązaniu dokumentów. W tym między innymi:
 - a) opracowanie algorytmu TC tworzenia hierarchii grup z drzewa maksymalnych przepływów;
 - b) zbadanie własności hierarchii utworzonej przez algorytm TC (niezależność od postaci drzewa maksymalnych przepływów, rozkład dokumentów hierarchii).

2. Opracowanie czterech wersji funkcji podobieństwa, wykorzystujących strukturę oraz treść dokumentów hipertekstowych, ze szczególnym uwzględnieniem elementów charakterystycznych dla stron WWW.

Poza tym w pracy przedstawiono:

1. Klasyfikację elementów struktury systemów hipertekstowych.
2. Definicję środowiska hipertekstowego i kolekcji hipertekstowej.
3. Klasyfikację i ocenę metod grupowania, ze szczególnym uwzględnieniem metod stosowanych do dokumentów hipertekstowych.
4. Implementację metody grupowania¹, która służy do jej weryfikacji, a także może być wykorzystywana do przeprowadzania innych badań w środowisku WWW.
5. Sposób wyznaczania reprezentantów grup za pomocą słów kluczowych.

Można wyróżnić trzy zasadnicze dziedziny, w których hierarchiczne grupowanie dokumentów na podstawie drzewa maksymalnych przepływów ma największe zastosowanie:

1. Wyszukiwanie informacji.
2. Prezentowanie zbioru dokumentów użytkownikowi.
3. Wspomaganie automatycznego tworzenia katalogów tematycznych.

W wyszukiwaniu informacji grupowanie jest stosowane do modyfikacji kwerend. Użytkownik po zadaniu pytania otrzymuje w odpowiedzi nie listę dokumentów, ale hierarchię grup dokumentów. Po wyrażeniu przez użytkownika zainteresowania wybranymi grupami w hierarchii, system automatycznie (z wykorzystaniem słów kluczowych będących reprezentantami grup) przekształca początkowe pytanie. W wyniku zadania zmodyfikowanego pytania otrzymuje się bardziej relevantną listę odpowiedzi.

Prezentacja zbioru odpowiedzi w postaci hierarchii grup dokumentów, zamiast (jak to jest najczęściej – listy) umożliwia realizację innej strategii przekazywania informacji. Użytkownik nie czyta po kolei wszystkich dokumentów, lecz przegląda reprezentantów grup najniżej położonych w hierarchii (liście). Jeżeli znajdzie wśród nich grupy odpowiadające jego potrzebom informacyjnym, to czyta dokumenty w nich zawarte i ewentualnie przegląda także dokumenty z grup wyżej położonych w hierarchii (ogólniejszych względem tej grupy, która okazała się dla niego interesująca). Inne hierarchie, inne gałęzie danej hierarchii oraz dokumenty izolowane mogą być przez użytkownika pominięte. Nawet jeżeli znajdują się w nich dokumenty interesujące, to jest ich mało. Dzięki temu możliwe jest szybsze dotarcie do dokumentów relevantnych.

Inne zastosowanie grupowania w systemach prezentacyjnych, to wspomaganie generowania struktur systemów informacyjnych WWW, tzw. map serwisów. Zawierają one hierarchię stron dostępnych w danym serwisie. Aktualnie najczęściej tworzone są na podstawie „ręcznego” indeksowania poszczególnych stron.

Kolejnym obszarem zastosowań hierarchicznego grupowania dokumentów jest wspomaganie tworzenia katalogów tematycznych. System grupujący podpowiada w takim przypadku podział zadanej kolekcji dokumentów na hierarchie grup, zaś użytkownik jedynie przeprowadza ewentualne korekty.

Badania związane z proponowaną metodą grupowania na podstawie drzew maksymalnych przepływów można dalej prowadzić w trzech kierunkach:

1. Dalszego testowania metody. Można tutaj wymienić:
 - a) Badania dla większej liczby kolekcji testowych o bardziej różnorodnej tematyce.

¹ W wyniku współpracy ze studentami specjalności Systemy Informacyjne Wydziału Informatyki i Zarządzania Politechniki Wrocławskiej powstał system CIS.

- b) Porównanie ocen hierarchii wygenerowanych na podstawie zaprezentowanej metody grupowania oraz wygenerowanych z wykorzystaniem wybranych metod przedstawionych w rozdziale 3.¹
 - c) Analiza zależności pomiędzy wartościami parametrów metody (τ , max^{lev} , α , β , γ) a charakterystyką kolekcji (na przykład kolekcje o ściśle lub luźno związanych ze sobą dokumentach).
 - d) Ocena wpływu zastosowania grupowania hierarchicznego zbioru dokumentów i automatycznej redefinicji zapytań na dokładność wyszukiwania.
2. Modyfikacje niektórych elementów metody, w szczególności:
- a) Zbadanie wpływu zastosowania innych funkcji podobieństwa na jakość tworzonych hierarchii. Można tutaj wykorzystać wersje A, B, C funkcji podobieństwa kompleksowego a także niektóre funkcje zaprezentowane w pkt. 3.1.
 - b) Zastosowanie innych sposobów skalowania (niż te, przedstawione w pkt. 5.5), w których szerokości przedziałów nie będą równe.
 - c) Zbadanie możliwości wykorzystania innych sposobów wyznaczania reprezentantów w postaci słów kluczowych.
 - d) Przekształcenie metody tak, aby w hierarchiach nie było grup pustych.
3. Rozszerzenia metody grupowania wykorzystującego maksymalny przepływ w grafach. Ważnym uogólnieniem byłoby opracowanie metody grupowania dla grafów zorientowanych. Wprowadzenie tego rozszerzenia ma duże konsekwencje dla całej metody, gdyż nie jest wtedy możliwe zastosowanie drzewa maksymalnych przepływów. Innym, istotnym rozszerzeniem jest wprowadzenie mechanizmów aktualizacji hierarchii przy zmianach w kolekcji wejściowej (chodzi szczególnie o dodawanie dokumentów do kolekcji).

¹ Badania w tym zakresie są już aktualnie prowadzone w ramach pracy magisterskiej.

LITERATURA

- [Abr96] Abramowicz W.: *Retoryka hipertekstu*. Informatyka, Nr 3, 1996, s. 4-7.
- [All93] Allen R.B., Obry P., Littman M.: *An interface for navigating clustered document sets returned by queries*. W: Proceedings of the Conference on Organizational Computing Systems, Milpitas, CA, USA, November 1 - 4, 1993, s. 166-171.
- [And73] Andenberg M.R.: *Cluster Analysis for Applications*. New York, Academic Press, 1973.
- [Ani94] Anick P.G.: *Adapting a full-text information retrieval system to the computer troubleshooting domain*. W: Croft W.B., van Rijsbergen C.J. (eds.): Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 3-6 July 1994, Springer Verlag, 1994, s. 349 - 358.
- [Ash94] Ashman H.L., Verbyla J.L.M., Cawley T.: *Hypermedia Management in Large-Scale Information Systems Using the Functional Model of the Link*. W: Proceedings of the Fifth Australasian Database Conference, New Zealand, January 1994, s. 247-257. <http://www.cs.flinders.edu.au/Research/LINKS/adc94.html>.
- [Bar96] Baron L., Tague-Sutcliffe J., Kinnucan M.T., Carey T.: *Labeled, typed links as cues when reading hypertext documents*. Journal of the American Society for Information Science, Volume 47, Number 12, December 1996, s. 896-908.
- [Ber91] Bernstein M., Bolter J.D., Joyce M., Mylonas E.: *Architectures for volatile hypertext*. W: Proceedings of the Third Annual ACM Conference on Hypertext '91, San Antonio, Texas, December 15-18, 1991, New York, ACM Press, 1991, s. 243-260.
- [Ber94] Berners-Lee T., Masinter L., McCahill M., eds: *Uniform Resource Locators (URL). Request for Comments: 1738*. Network Working Group, December 1994, <http://www.cis.ohio-state.edu/htbin/rfc/rfc1738.html>.
- [Bie92] Bieber M.P., Kimbrough S.O.: *On Generalizing the Concept of Hypertext*. MIS Quarterly, vol. 16, no. 1, 1992, s. 77-93. <http://www.cba.uga.edu/misq/MISQ/V16/V16n1art4.html>.
- [Bie97] Bieber M., Vitali F., Ashman H., Balasubramanian V., Oinas-Kukkonen H.: *Fourth generation hypermedia: some missing links for the World Wide Web*. International Journal of Human-Computer Studies, Special Issue: World Wide Web Usability, Vol. 47, No. 1, 1997, s. 31-65, <http://ijhcs.open.ac.uk/bieber/bieber.html>.
- [Bha98a] Bharat K., Broder A.: *A technique for measuring the relative size and overlap of public web search engines*. Computer Networks and ISDN Systems, Vol. 30, Issue 1-7, 01-April-1998, s. 379-388. Także: Processing of the 7th International World Wide Web Conference, Brisbane, Australia, 14-18 April, 1998. <http://www7.scu.edu.au/programme/fullpapers/1937/com1937.htm>; także uzupełnienie: <http://www.research.digital.com/SRC/whatsnew/sem.html>.
- [Bha98b] Bharat K., Henzinger M.R.: *Improved algorithms for topic distillation in a hyper-linked environment*. W: SIGIR '98. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28, Melbourne, Australia, 1998, s. 104-111.
- [Blo94] Bloomfield H.: *Links in Hypertext: An investigation into how they can provide information on inter-node relationship*. Ph.D. Thesis, Queen Mary and Westfield College, University of London, September 1994.

- [Bol98] Bollacker K.D., Lawrence S., Giles C.L.: *CiteSeer: an autonous Web agent for automatic retrieval and identification of interesting publications*. W: Sycara K.P., Wooldridge M. (eds.): AGENTS'98. Proceedings of the Second International Conference on Autonomous Agents, New York, ACM Press, 1998, s. 116-123.
- [Bot92] Botafogo R.A., Rivlin E., Shneiderman B.: *Structural analysis of hypertexts: identifying hierarchies and useful metrics*. ACM Transaction on Information Systems, Vol. 10, No. 2, April 1992, s. 142-180.
- [Bot93] Botafogo R.A.: *Cluster analysis for hypertext systems*. W: Korfhage R., Rasmussen E., Willett P. (eds.): Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, ACM, New York, 1993, s. 116-125.
<http://jambo.caa.uff.br/~botafogo/papers/sigir/sigir.html>.
- [Bri98] Brin S. Page L.: *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Computer Networks and ISDN Systems, Vol. 30, Issue 1-7, 01-April-1998, s. 107-117. Także: Processing of the Seventh International World Wide Web Conference, Brisbane, Australia, 14-18 April, 1998.
- [Bro97] Broder A., Glassman S.C., Manasse M.S., Zweig G.: *Syntactic clustering of the Web*. Computer Networks and ISDN Systems, Vol. 29, Issue 8-13, 30 September 1997, s. 1157-1166. Także: Proceedings of the Sixth International WWW Conference, Santa Clara, California USA, April 7-11, 1997.
<http://www.scope.gmd.de/info/www6/technical/paper205/paper205.html>.
- [Bur95] Burgin R.: *The retrieval effectiveness of five clustering algorithms as a function of indexing exhaustivity*. Journal of the American Society for Information Science, Volume 46, Number 8, 1995, s. 562-572.
- [Bus45] Bush V.: *As we may think*. The Atlantic Monthly, 176, no. 1, July 1945, s. 101-108, <http://www.isg.sfu.ca/~duchier/misc/vbush>.
- [Car97] Carrière S.J., Kazman R.: *WebQuery: searching and visualizing the Web through connectivity*. Computer Networks and ISDN Systems, Vol. 29, Issue 8-13, 30 September 1997, s. 1257-1267. Także: Proceedings of the Sixth International WWW Conference, Santa Clara, California USA, April 7-11, 1997.
<http://www.scope.gmd.de/info/www6/technical/paper096/paper96.html>.
- [Chak98a] Chakrabarti S., Dom B., Raghavan P., Rajagopalan S., Gibson D., Kleinberg J.: *Automatic resource compilation by analyzing hyperlink structure and associated text*. Computer Networks and ISDN Systems Vol. 30, Issue 1-7, 01-April-1998, s. 65-74. Także: Processing of the 7th International World Wide Web Conference, Brisbane, Australia, 14-18 April, 1998.
<http://www.cs.cornell.edu/home/kleinber/www98/438.html>.
- [Chak99] Chakrabarti S., van den Berg M., Dom B.: *Focused Crawling: A New Approach to topic-specific web resource discovery*. Computer Networks, Volume 31, Issue 11-16, 17-May-1999, s. 1623-1640. Także: Mendelzon A. (ed.): The 8th International WWW Conference Proceedings, Toronto, Canada, May 11-14, 1999, University of Toronto, Canada. <http://www8.org/w8-papers/5a-search-query/crawling/index.html>.
- [Chan98] Chang C.H., Hsu C.C.: *Hypertext information retrieval for short queries*. W: Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop, Taipei, Taiwan, November 1998.
<http://www.csie.ntu.edu.tw/~f2506005/pub/kdex98.ps.gz>.
- [Chan99] Chang C.H., Hsu C.C.: *Enabling concept-based relevance feedback on World Wide Web*. IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 4, July/August 1999, s. 595-609. <http://www.iis.sinica.edu.tw/~ellie/pub/tkde.zip>.

- [Chen97] Chen C.: *Structuring and visualising the WWW by generalised similarity analysis*. W: Bernstein M., Carr L., Østerbye K. (eds.): *Hypertext 97*. The Eighth ACM Conference on Hypertext, University of Southampton, UK, April 6-11, 1997, ACM Press, 1997, s. 177-186.
- [Con87] Conklin J.: *Hypertext: an introduction and survey*. Computer, vol. 20, no. 9, September 1987, IEEE Publications, s. 17-41.
- [Cor98] Cormen T.H., Leiserson C.E., Rivest R.L.: *Wprowadzenie do algorytmów*. WNT, Warszawa, 1998.
- [Cro93] Croft W.B., Turtle H.R.: *Retrieval strategies for hypertext*. Information Processing and Management, Vol. 29, No. 3, 1993, s. 313-324.
- [Cro95] Croft B., Cook R., Wilder D.: *Providing Government Information on the Internet: Experiences with THOMAS*. W: Proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries, June 11-13, 1995, Austin, Texas, USA, Hypermedia Research Lab, Computer Science Department, Texas A&M University, s. 19-24. <http://www.csd.tamu.edu/csd/DL95/contents.html>.
- [Cut92] Cutting D.R., Pedersen J.O., Karger D., Tukey J.W.: *Scatter/Gather: a cluster-based approach to browsing large document collections*. W: Belkin N.J., Ingwersen P., Pejtersen A.M. (eds.): Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Copenhagen, Denmark, June 21-24, 1992. ACM, New York, s. 318-329. <http://corp.excite.com/people/cutting/papers/sigir92.ps>.
- [Cut93] Cutting D.R., Karger D.R., Pedersen J.O.: *Constant interaction-time scatter/gather browsing of very large document collections*. W: Korfhage R., Rasmussen E., Willett P. (eds.): Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, ACM, New York, 1993, s. 126 - 134. <http://corp.excite.com/people/cutting/papers/sigir93.ps>.
- [Dan98] Daniłowicz Cz.: *Dokumenty elektroniczne*. Zagadnienia Informacji Naukowej, Nr 1, 1998, s. 55-63.
- [Dan00a] Daniłowicz Cz.: *Porządkowanie dokumentów na podstawie grafu powiązań*. Raport Zakładu Systemów Informacyjnych Politechniki Wrocławskiej SPR 29, 2000.
- [Dan00b] Daniłowicz Cz.: *Wyszukiwanie informacji w otwartym systemie WWW*. Informatyka, Nr 3, Marzec 2000, s. 11-19.
- [Dav96] Davies N.J., Weeks R., Revett M.C.: *Information agents for the World Wide Web*. BT Technology Journal, Vol. 14, No. 4, October 1996. <http://www.bt.com/btj/vol14no4/index.htm>.
- [Dąb78] Dąbrowski M., Laus-Maczyńska K.: *Metody wyszukiwania i klasyfikacji informacji*. Wydawnictwa Naukowo-Techniczne; Warszawa, 1978.
- [Def77] Defays D.: *An Efficient Algorithm for a Complete Link Method*. Computer Journal, Volume 20, Number 4, 1977, s. 364-366.
- [Deo80] Deo N.: *Teoria grafów i jej zastosowania w technice i informatyce*. Państwowe Wydawnictwo Naukowe, Warszawa, 1980.
- [DeR89] DeRose S.J.: *Expanding the notion of links*. W: Second Annual ACM Conference Proceedings on Hypertext '89, November 5-8, 1989, Pittsburgh, Pennsylvania, USA, ACM, 1989, s. 249-257.
- [Dij78] Dijkstra E.: *Zadanie o najkrótszym drzewie powiązań*. Umiejętność programowania. WNT, 1978, str. 156-162.
- [Dur00] Durka E.: *Ilu badaczy, tyle Internetu*. Gazeta Wyborcza, 6 III 2000.

- [Etz95] Etzioni O., Selberg E.: *Multi-service search and comparison using the MetaCrawler*. W: Proceedings of the 4th International World Wide Web Conference '95; Boston; Massachusetts, USA, December 11-14, 1995.
<http://draz.cs.washington.edu/papers/www4/html/Overview.html>.
- [Fre95] Frei H.P., Stieger D.: *The use of semantic links in hypertext information retrieval*. Information Processing & Management, Vol. 31, No. 1, 1995, s. 1-13.
- [Gar79] Garfield E.: *Citation indexing*. A Wiley-Interscience Publication, New York, 1979.
- [Gar93] Garzotto F., Paolini P., Schwabe D.: *HDM - A Model-Based Approach to Hypertext Application Design*. ACM Transactions on Information Systems, Vol. 11, No. 1, January 1993, s. 1-26.
- [Gdu99] Gdula Ł., Kossowski K.: *Wyszukiwanie stron WWW z wykorzystaniem grupowania dokumentów*. Praca magisterska. Zakład Systemów Informacyjnych Wydziału Informatyki i Zarządzania, Politechnika Wrocławska, Wrocław 1999.
- [Gib98] Gibson D., Kleinberg J., Raghavan P.: *Inferring Web communities from link topology*. W: Proceedings of the ninth ACM conference on hypertext and hypermedia: links, objects, time and space structure in hypermedia systems, Pittsburgh, PA, USA, June 20-24, ACM Press, 1998, s. 225-234.
<http://simon.cs.cornell.edu/home/kleinber/ht98.ps>.
- [Gol86] Goldberg A.V., Tarjan R.E.: *A new approach to the maximum-flow problem*. W: STOC '86. Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, 1986, s. 136-146.
- [Gol88] Goldberg A.V., Tarjan R.E.: *A new approach to the maximum-flow problem*. Journal of the ACM, Vol. 35, No. 4, October 1988, s. 921-940.
- [Gol96] Goldberg J.L.: *CDM: An approach to learning in text categorization*. International Journal on Artificial Intelligence Tools, Vol. 5, Nos. 1&2, 1996, s. 229-253.
- [Gol97] Golovchinsky G.: *What the query told the link: the integration of hypertext and information retrieval*. W: Bernstein M., Carr L., Østerbye K. (eds.): Hypertext 97, The Eighth ACM Conference on Hypertext, University of Southampton, UK, April 6-11, 1997. ACM Press, 1997, s. 67-74.
- [Gra86] Granot F., Hassin R.: *Multi-terminal maximum flows in node capacitated networks*. Discrete Applied Mathematics, Vol. 13, 1986, s. 157-163.
- [Gri84] Griffiths A., Robinson L.A., Willett P.: *Hierarchic agglomerative clustering methods for automatic document classification*. Journal of Documentation, Vol. 40, 1984, s. 175-205.
- [Gri86] Griffiths A., Luckhurst H.C., Willett P.: *Using interdocument similarity information in document retrieval systems*. Journal of the American Society for Information Science, Volume 37, Number 1, 1986, s. 3-11.
- [Gui00] Guillaume D., Murtagh F.: *Clustering of XML documents*. Computer Physics Communications, 2000, w druku. <http://strule.cs.qub.ac.uk/~fmurtagh/kd-ir/kd-ir-dg-fm.html>.
- [Gus87] Gusfield D.: *Very Simple Algorithms and Programs for All Pairs Network Flow Analysis*. Technical Report CSE-87-1, Computer Science Division, University of California, Davis, April 16, 1987.
- [Gus90] Gusfield D.: *Very Simple Methods for All Pairs Network Flow Analysis*. SIAM Journal on Computing, Vol. 19, No. 1, February 1990, s. 143-155.
- [GVU98] *GVU's 9th WWW User Survey*. Georgia Tech Research Corporation, April 1998.
http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-04.

- [Haa98] Haas S.W., Grams E.S.: *Page and link classifications: connecting diverse resources*. W: Proceedings of the Third ACM Conference on Digital Libraries, June 23-26, 1998, Pittsburgh, PA, USA, ACM Press, 1998, s. 99-107.
- [Hal93] Hall W., Hill G., Davis H.: *The microcosm link service*. W: Hypertext '93. Proceedings of the Fifth ACM Conference on Hypertext, Seattle, WA, November 14-18, 1993, ACM, 1993, s. 256-259.
- [Hal94] Halasz F., Schwartz M.: *The Dexter hypertext reference model*. Communications of the ACM, Vol. 37, No. 2, Feb. 1994, s. 30-39.
- [Ham89] El-Hamdouchi A., Willett P.: *Comparison of hierarchic agglomerative clustering methods for document retrieval*. Computer Journal, Vol. 32, No.3, 1989, s. 220-227.
- [Har99] Adam Harriss A., Huneke C.: *English will dominate Web for only three more years*. Computer Economics, June 9, 1999, <http://www.computereconomics.com/new4/pr/pr990610.html>.
- [HTM98a] *HTML 4.0 Specification*. W3C, World Wide Web Consortium, 24-April-1998. <http://www.w3.org/TR/REC-html40>.
- [HTM98b] *B.4 Notes on helping search engines index your Web site*. W: *HTML 4.0 Specification*. W3C, World Wide Web Consortium, 24-April-1998. <http://www.w3.org/TR/REC-html40/appendix/notes.html#h-B.4>.
- [Hu61] Gomory R.E., Hu T.C.: *Multi-Terminal Network Flows*. SIAM Journal on Applied Mathematics, Vol. 9, No. 4, 1961, s. 551-570.
- [Hu82] Hu T.C.: *Combinatorial algorithms*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1982.
- [Hu92] Cheng C.K., Hu T.C.: *Maximum Concurrent Flows and Minimum Cuts*. Algorithmica, Vol. 8, No. 3, 1992, s. 233-249.
- [Ink00] *Web Surpasses One Billion Documents*. Inktomi News & Events, January 18, 2000. <http://www.inktomi.com/new/press/billion.html>.
- [IPM97] Information Processing & Management, Vol. 33, No. 2, 1997, Special issue on methods and tools for the automatic construction of hypertext.
- [Jar71] Jardine N., Sibson R.: *Mathematical Taxonomy*. John Wiley & Sons Ltd.; London; 1971.
- [Kac91] Kacmar C.J., Leggett J.J.: *PROXHY: a process-oriented extensible hypertext architecture*. ACM Transactions on Information Systems, Vol. 9, No. 4, October 1991, s. 399-419.
- [Kaz96] Kazienko P., Karpiak-Kozińska B.: *Wyszukiwanie i transfer zbiorów*. W: Daniłowicz Cz. (red.): *Sieciowe i multimedialne systemy informacyjne*. Zakład Systemów Informacyjnych Wydziału Informatyki i Zarządzania Politechniki Wrocławskiej, Wrocław, 1996, s. 69-80.
- [Kaz97] Katarzyniak R., Kazienko P., Sobecki J.: *Wybrane problemy wyszukiwania informacji w sieci Internet*. W: IV Krajowe Forum Informacji Naukowej i Technicznej. Materiały konferencyjne. Polskie Towarzystwo Informacji Naukowej, Zakopane, 2-5 września 1997, Warszawa, Pol. Tow. Inf. Nauk., s. 286-295. <ftp://ftp.zsi.pwr.wroc.pl/publications/Kazienko/Zakopane97>.
- [Kaz98a] Kazienko P.: *Grupowanie stron WWW na podstawie odsyłaczy hipertekstowych*. W: MiSSI. Multimedialne i Sieciowe Systemy Informacyjne. Materiały konferencyjne, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 1998, s. 119-129. <ftp://ftp.zsi.pwr.wroc.pl/publications/Kazienko/MISSI98>.

- [Kaz98b] Kazienko P.: *Struktura hipertekstu a struktura systemu WWW*. Zagadnienia Informatyki Naukowej, Nr 2 (72), 1998, s. 36-56.
<ftp://ftp.zsi.pwr.wroc.pl/publications/Kazienko/ZIN1998>.
- [Kaz99a] Kazienko P.: *Rodzaje stron i odsyłaczy w systemie WWW*. Informatyka, Nr 2, Luty 1999, s. 24-35. <ftp://ftp.zsi.pwr.wroc.pl/publications/Kazienko/Informatyka2-99>.
- [Kaz99b] Kazienko P.: *Algorytm Gusfielda tworzenia drzewa maksymalnych przepływów*. Raport Zakładu Systemów Informacyjnych Politechniki Wrocławskiej SPR 19, 1999.
- [Kaz99c] Kazienko P.: *Podobieństwo dokumentów hipertekstowych w systemie WWW*. Raport Zakładu Systemów Informacyjnych Politechniki Wrocławskiej SPR 20, 1999.
- [Kaz00a] Kazienko P., Prusiewicz P.: *Przegląd metod grupowania dokumentów hipertekstowych*. Raport Zakładu Systemów Informacyjnych Politechniki Wrocławskiej SPR 28, 2000.
- [Kaz00b] Kazienko P.: *Źródła podobieństwa stron WWW*. Artykuł przyjęty na konferencję MiSSI 2000.
- [Kaz00c] Kazienko P.: *Hierarchiczna metoda podziału kolekcji stron WWW na podstawie wartości maksymalnego przepływu*. Artykuł przyjęty na konferencję MiSSI 2000.
- [Kle99] Kleinberg J.: *Authoritative Sources in a Hyperlinked Environment*. Journal of the ACM, Vol. 46, No. 5, September 1999, s. 604-632.
- [Koh99] Koehler W.: *An analysis of Web page and Web site constancy and permanence*. Journal of the American Society for Information Science, Volume 50, Number 2, 1999, s. 162-180.
- [Kor78] Korzan B.: *Elementy teorii grafów i sieci*. Wydawnictwa Naukowo-Techniczne, Warszawa 1978.
- [Kuch82] Kucharczyk J.: *Algorytmy analizy skupień w języku ALGOL 60*. Państwowe Wydawnictwo Naukowe, Warszawa, 1982.
- [Kul86] Kulikowski J.L.: *Zarys teorii grafów. Zastosowania w technice*. Państwowe Wydawnictwo Naukowe, Warszawa, 1986.
- [Kur80] Kuratowski K.: *Wstęp do teorii mnogości i topologii*. Wydanie 8, PWN, Warszawa, 1980.
- [Lan66] Lance G.N. Williams W.T.: *A General Theory of Classificatory Sorting Strategies. I. Hierarchical systems*. Computer Journal, Vol. 9, 1966, s. 373-380.
- [Lar96] Larson R.R.: *Bibliometrics of the World Wide Web: An Exploratory Analysis of the Intellectual Structure of Cyberspace*. W: Hardin S. (ed.): ASIS'96. Proceedings of the 59th Annual Meeting of the American Society of Information Science, Baltimore, MD., October 19-24, 1996, Information Today, Medford, NJ, 1996. s.71-77.
<http://sherlock.berkeley.edu/asis96/asis96.html>.
- [Law98] Lawrence S., Lee Giles C.: *Searching the World Wide Web*. Science, Volume 280, Number 5360, April 3 1998, p. 98.
<http://www.neci.nj.nec.com/homepages/lawrence/websize.html>.
- [Law99a] Lawrence S., Giles C.L., Bollacker K.D.: *Autonomous citation matching*. W: AGENTS '99. Proceedings of the Third Annual Conference on Autonomous Agents, Seattle, USA, May 1 - 5, 1999, ACM, 1999, s. 392-393.
- [Law99b] Lawrence S., Lee Giles C.: *Accessibility of information on the web*. Nature, Vol. 400, 8 July 1999, s. 107-109.
- [Lip89] Lipski W.: *Kombinatoryka dla programistów*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1989.

- [Maa96] Maarek Y.S., Ben-Shaul I.Z.: *Automatically Organizing Bookmarks per Contents*. Computer Networks and ISDN Systems Vol. 28, Issue 7-11, 01-May-1996, s. 1321-1334. Także w: Proceedings of the Fifth International WWW Conference, May 6-10, 1996, Paris, France, Elsevier Science.
http://www5conf.inria.fr/fich_html/papers/P37/Overview.html.
- [Mar97] Marchiori M.: *The quest for correct information on the Web: hyper search engines*. Computer Networks and ISDN Systems, Vol. 29, Issue 8-13, 30 September 1997, s. 1225-1236. Także w: Proceedings of the Sixth International WWW Conference, Santa Clara, California USA, April 7-11, 1997.
<http://www.scope.gmd.de/info/www6/technical/paper222/paper222.html>.
- [Mat77] Matula D.W.: *Graph theoretic techniques for cluster analysis algorithms*. W: van Ryzin J. (ed.): Classification and clustering. Academic Press, Inc., 1977. Proceedings of an advanced seminar conducted by the Mathematics Research Center, The University of Wisconsin at Madison, May 3-5, 1976, s. 95-129.
- [Muk94] Mukherjea S., Foley J., Hudson S.: *Interactive Clustering for Navigating in Hypermedia Systems*. W: ECHT '94. Proceedings of the ACM European Conference on Hypermedia Technology, Edinburgh, Scotland, UK, Sept. 18-23, 1994, ACM, 1994, s. 136-144.
- [Mur83] Murtagh F.: *A Survey of Recent Advances in Hierarchical Clustering Algorithms*. Computer Journal, Vol. 26, No. 4, 1983, s. 354-359.
- [Ols95] Olson C.F.: *Parallel Algorithms for Hierarchical Clustering*. Parallel Computing, Vol. 21, No 8, August 1995, s. 1313-1325.
<http://robotics.jpl.nasa.gov/people/olson/abstracts/csd-94-786.html>.
- [Pau98] Paulo F.B., Turine M.A.S., de Oliveira M.C.F., Masiero P.C.: *XHMBS: a formal model to support hypermedia specification*. W: Proceedings of the ninth ACM conference on hypertext and hypermedia: links, objects, time and space structure in hypermedia systems, Pittsburgh, PA, June 20-24, 1998, s. 161-170.
- [Pin94] Pinkerton B.: *Finding what people want: experiences with the WebCrawler*. W: Proceedings of the 2nd International WWW Conference '94, Chicago, July 1994.
<http://info.webcrawler.com/bp/WWW94.html>.
- [Pir96] Pirolli P., Pitkow J., Rao R.: *Silk from a sow's ear: extracting usable structures from the Web*. W: CHI '96. Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground. April 13-18, 1996, Vancouver, British Columbia, Canada, s. 118-125.
- [Pit96] Pitkow J.E., Kehoe C. M.: *Emerging trends in the WWW user population*. Communications of the ACM, Vol. 39, No. 6, June 1996, s. 106-108.
- [Pit97] Pitkow J., Pirolli P.: *Life, death, and lawfulness on the electronic frontier*. W: Steven P. (ed.): CHI 97: Conference Proceedings on Human Factors in Computing Systems, Atlanta, Georgia, 22-27 March 1997. ACM/Addison-Wesley, 1997, s. 383-390.
- [Ras92] Rasmussen E.: *Clustering algorithms*. W: Frakes W., Baeza-Yates R. (eds.): Information retrieval: data, structures & algorithms. Chap 16, Englewood Cliffs, NJ: Prentice Hall, 1992, s. 419-442.
- [Rij71] van Rijsbergen C.J.: *An Algorithm for Information Structuring and Retrieval*. Computer Journal, Vol. 14, No. 4, 1971, s. 407-412.
- [Rij79] van Rijsbergen C.J.: *Information retrieval*. Butterworths, London, second edition, 1979. <http://www.dcs.glasgow.ac.uk/Keith/Preface.html>.

- [Riv94] Rivlin E., Botafogo R.A., Shneiderman B.: *Navigating in Hyperspace: Designing a Structure-Based Toolbox*. Communications of the ACM, Volume 37, Number 2, February 1994, s. 87-96.
- [Rob98] Robinson D.A., Lester C.R., Hamilton N.M.: *Delivering computer assisted learning across the WWW*. Computer Networks and ISDN Systems, Vol. 30, Issue 1-7, 01-April-1998, s. 301-307. Także w: Proceedings of the 7th WWW Conference, Brisbane, Australia, 14-18 April, 1998. <http://www7.scu.edu.au/programme/fullpapers/1876/com1876.htm>.
- [Sal75a] Salton G., pod redakcją: *SMART automatyczny system wyszukiwania informacji*. Warszawa, WNT, 1975.
- [Sal75b] Salton G., Wong A., Yang C.S.: *A vector space model for information retrieval*. Journal of the American Society for Information Science, Volume 18, Number 11, November 1975, s. 613-620.
- [Sal83] Salton G., McGill M.J.: *Introduction to Modern Information Retrieval*. New York, McGraw-Hill Book Co., 1983.
- [Sal88] Salton G., Buckley C.: *Term-Weighting Approaches in Automatic Text Retrieval*. Information Processing and Management, Vol.24, No.5, 1988, s. 513-523.
- [Sal99] Salampasis M., Tait J.: *A link-based collection fusion strategy*. Information Processing and Management, Volume 35, Number 5, 1999, s. 691-711.
- [Sib73] Sibson R.: *SLINK: an Optimally Efficient Algorithm for the Single-Link Cluster Method*. Computer Journal, Vol. 16, No. 1, 1973, s. 30-34.
- [Sin96] Singhal A., Buckley C., Mitra M.: *Pivoted Document Length Normalization*. W: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96, August 18-22, 1996, Zurich, Switzerland. Special Issue of the SIGIR Forum, ACM, New York, s. 21-29.
- [Shn89] Shneiderman B., Kearsley G.: *Hypertext Hands On!*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
- [Sko] Skorobohatyj G.: *MathPROG Code for Finding a Minimum Cut between all Node Pairs in an Undirected Graph*. Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB). <http://elib.zib.de/pub/Packages/mathprog/mincut/all-pairs>.
- [Soe67] Soergel D.: *Mathematical analysis of documentation systems. An attempt to a theory of classification and search request formulation*. Information Storage and Retrieval, Vol. 3, No. 3, July 1967, s. 129-173.
- [Spä77] Spät H.: *Cluster-Analyse-Algorithmen zur Objektklassifizierung und Datenreduktion*. R. Oldenbourg Verlag, München – Wien, 1977.
- [Spe97] Spertus E.: *ParaSite: mining structural information on the Web*. Computer Networks and ISDN Systems, Vol. 29, Issue 8-13, 30 September 1997, s. 1205-1215. Także w: Proceedings of the 6th WWW Conference, Santa Clara, California, USA, 7-11 April, 1997. <http://www.cs.washington.edu/homes/ellens/w3/parasite.html>.
- [Sud97] Sudhakar V., Murthy C.S.R.: *A modified noising algorithm for the graph partitioning problem*. Integration, The VLSI Journal, Vol. 22, Issue 1-2, 1997, 22, s. 101-113.
- [Sul00] Sullivan D.: *Search Engine Sizes*. Search Engine Watch, Number 39, February 3, 2000. <http://searchenginewatch.com/reports/sizes.html>.
- [Świ00] Świderek T.: *Ile to jest warte. Internet. Pierwsze próby wyceny polskiego rynku*. Rzeczpospolita, Nr 28 (5498), 3 lutego 2000.

- [Ter99] Terveen L., Hill W., Amento B.: *Constructing, organizing, and visualizing collections of topically related Web resources*. ACM Transactions on Computer-Human Interaction, Volume 6, Issue 1, 1999, s. 67-94.
- [Tri86] Trigg R.H., Weiser M.: *TEXTNET: a network-based approach to text handling*. ACM Transactions on Information Systems, Vol. 4, No. 1, January 1986, s. 1-23.
- [Voo86a] Voorhees E.M.: *The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval*. Ph.D. Dissertation., Department of Computer Science, Cornell University, Ithaca, N.Y., 1986.
- [Voo86b] Voorhees E.M.: *Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval*. Information Processing & Management, Vol. 22, No. 6, 1986, s. 465-476.
- [War63] Ward J.H. Jr.: *Hierarchical Grouping to Optimize an Objective Function*. Journal of the American Society for Information Science, Vol. 58, 1963, s. 235-244.
- [Wei96] Weiss R., Velez B., Sheldon M.A., Namprempre C., Szilagyi P., Duda A., Gifford D.K.: *HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering*. W: Hypertext'96. Proceedings of the 7th ACM Conference on Hypertext, Washington DC, March 16-20, 1996. ACM Press, 1996, s. 180-193. <http://www.cs.unc.edu/~barman/HT96/index.html>.
- [Wil85] Wilson R.J.: *Wprowadzenie do teorii grafów*. Państwowe Wydawnictwo Naukowe, Warszawa, 1985.
- [Wul97] Wulfekuhler M.R., Punch W.F.: *Finding salient features for personal web pages categories*. Computer Networks and ISDN Systems, Vol. 29, Issue 8-13, 30 September 1997, s. 1147-1156. Także w: Proceedings of the Sixth International WWW Conference, Santa Clara, California USA, April 7-11, 1997.
- [XLINK00] *XML Linking Language (XLink)*. World Wide Web Consortium, Working Draft 19-January-2000. <http://www.w3.org/TR/xlink>.
- [Zam98] Zamir O., Etzioni O.: *Web document clustering: a feasibility demonstration*. W: SIGIR '98. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, August 24-28, Melbourne, Australia, 1998, ACM, 1998, s. 46-54. <http://www.cs.washington.edu/homes/zamir/papers/sigir98.doc>.
- [Zam99] Zamir O., Etzioni O.: *Grouper: A Dynamic Clustering Interface to Web Search Results*. Computer Networks Volume 31, Issue 11-16, 17-May-1999, s. 1361-1374. Także w: The 8th International WWW Conference Proceedings, Toronto, Canada, May 11-14, 1999 Edited by A. Mendelzon, University of Toronto, Canada. <http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html>.

ZAŁĄCZNIK A. ALGORYTM GUSFIELDA

Pod koniec lat 80-tych Gusfield opracował metodę uzyskiwania drzewa maksymalnych przepływów, niekoniecznie będącego zarazem drzewem minimalnych przekrojów, która wymaga dokładnie $(N-1)$ rozwiązań problemu minimalnych przekrojów. W wyniku tego powstał algorytm EQ opisany w pracach [Gus87, Gus90, Kaz99b] i przedstawiony poniżej.

Algorytm EQ Gusfielda

Wejście: spójny, niezorientowany, ważony graf $G^S(V^S, E^S)$.

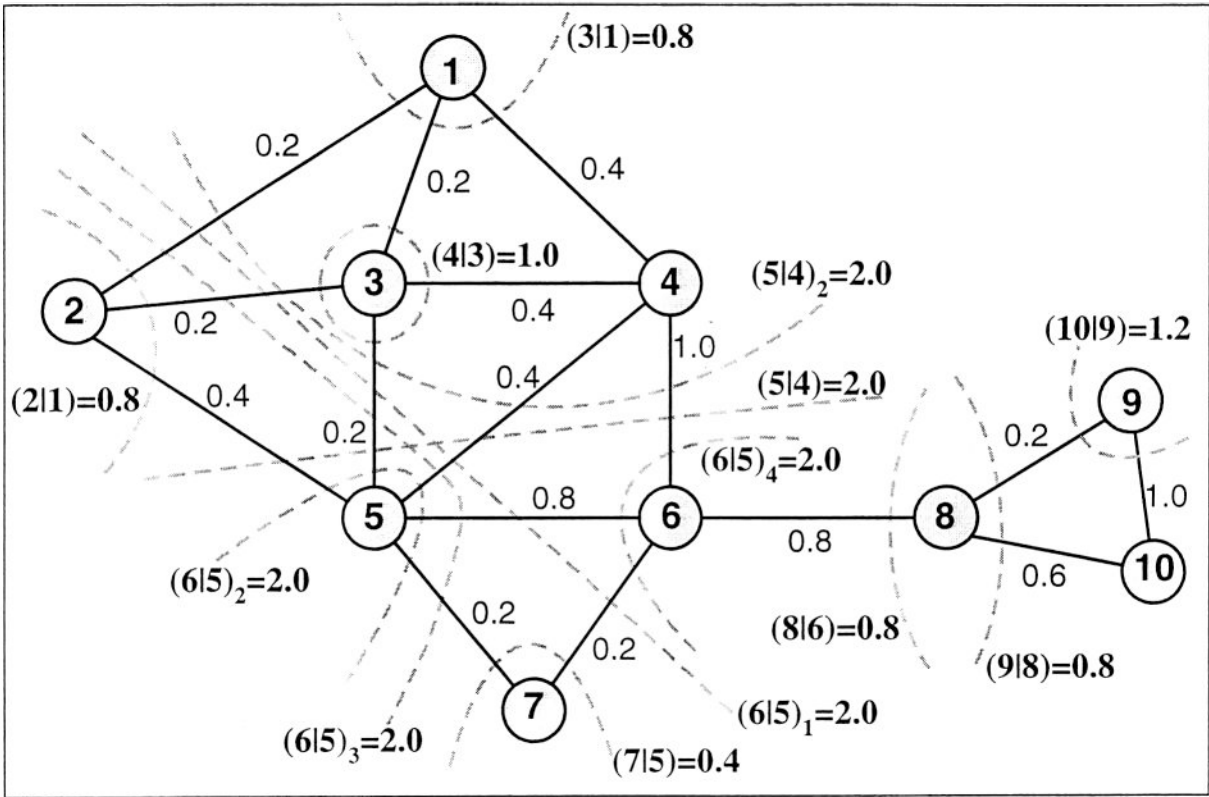
Wyjście: drzewo maksymalnych przepływów $T(V^S, E^T)$.

Algorytm wykorzystuje zewnętrzną procedurę obliczania minimalnego przekroju między dwoma węzłami w grafie $G^S(V^S, E^S)$.

- Krok 1. Utwórz drzewo (gwiazdę) $T(V^S, E^T)$ o N węzłach, z węzłem nr 1 jako centrum oraz węzłami o numerach od 2 do N jako liśćmi.
- Krok 2. Dla s od 2 do N wykonaj kroki 3 i 4
- Krok 3. Oblicz minimalny przekrój (XLY) w grafie $G^S(V^S, E^S)$ pomiędzy węzłem źródłowym v_s (liściem drzewa $T(V^S, E^T)$) a jego (jedynym) sąsiadem – ujściem v_t w $T(V^S, E^T)$. Oznacz krawędź $\{v_s, v_t\}$ w $T(V^S, E^T)$ wartością pojemności przekroju (XLY).
- Krok 4. Dla każdego węzła v_i , dla którego i jest większe niż s , jeśli v_i jest sąsiadem v_t oraz v_i jest po stronie źródła v_s przekroju (XLY), wtedy zmodyfikuj drzewo $T(V^S, E^T)$ poprzez odłączenie v_i z ujścia v_t i połączenie v_i do źródła v_s .

W kroku 1, ze zbioru węzłów V^S grafu $G^S(V^S, E^S)$ tworzone jest drzewo – gwiazda o środku w węźle o numerze 1 i wszystkich pozostałych połączonych z nim nieważoną krawędzią. Następnie z tej gwiazdy przenoszone (lub pozostawiane) są węzły o kolejnych numerach począwszy od 2 (krok 3). W każdej iteracji jeden raz wywoływana jest zewnętrzna procedura obliczania minimalnego przekroju i maksymalnego przepływu a wartość maksymalnego przepływu przyporządkowywana jest kolejnym krawędziom.

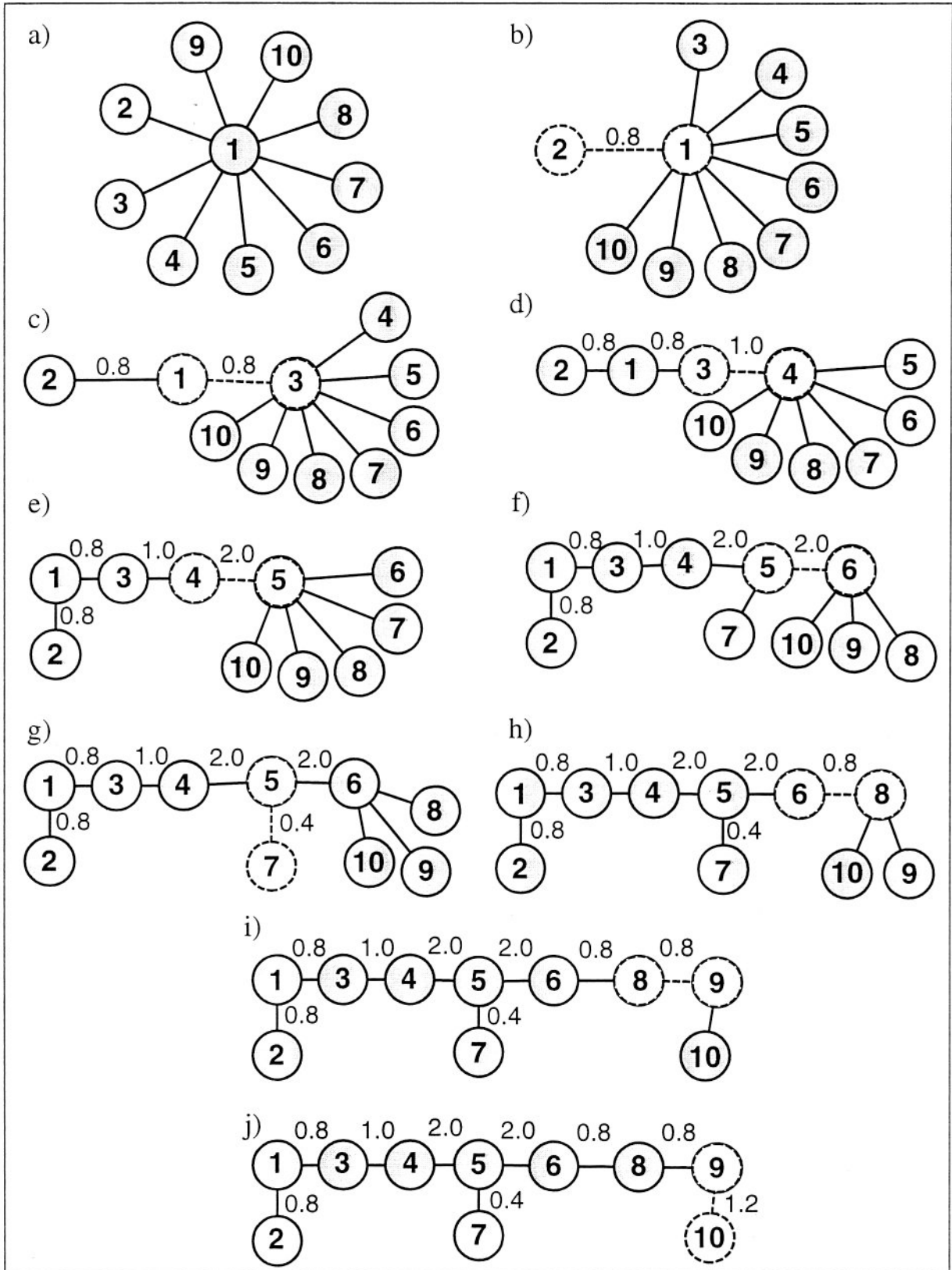
Łatwo zauważyć, że przy każdej iteracji, węzeł v_s i wszystkie węzły o numerze większym niż s są liśćmi w drzewie $T(V^S, E^T)$, a więc każdy kolejny – pobrany w kroku 2 – węzeł v_s posiada tylko jednego sąsiada, jak wymaga algorytm (krok 3).



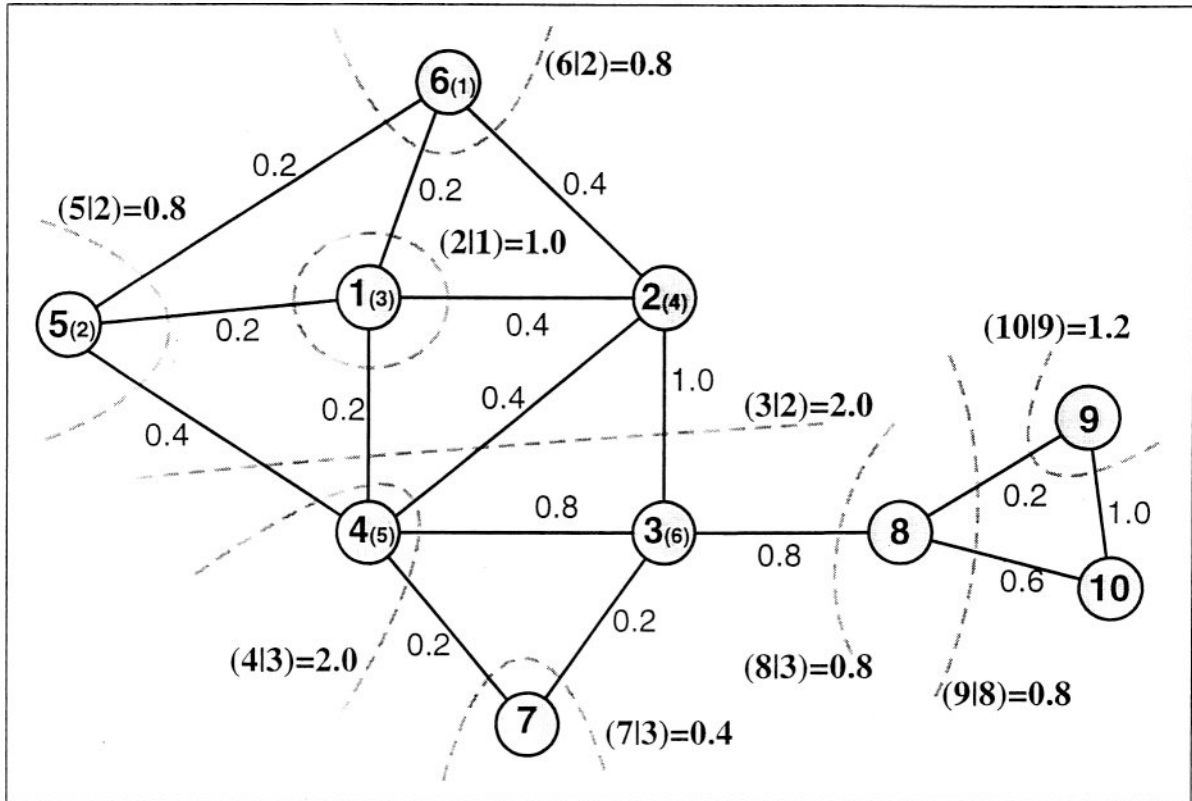
Rys. A.1. Graf z rys. 1.1 z zaznaczonymi minimalnymi przekrojami.

Działanie algorytmu EQ wyjaśnimy na przykładzie. Drzewo maksymalnych przepływów będziemy budować dla spójnego, niezorientowanego, ważonego grafu $G^S(V^S, E^S)$ przedstawionego na rys. A.1. Kolejne iteracje algorytmu EQ pokazano na rys. A.2. Początkowo wszystkie węzły o numerze większym od 1 (od 2 do 10) są połączone z środkiem gwiazdy – węzłem 1 (rys. A.2a). W pierwszej iteracji algorytmu ($v_s=2$), w trzecim kroku, obliczany jest minimalny przekrój między węzłem 2 oraz jego sąsiadem czyli węzłem 1 ($v_r=1$) – przekrój $(2|1)$, zaznaczony na rys. A.1. Wartość jego pojemności wynosi 0.8. Krawędź $\{2,1\}$ na rys. A.2b otrzymuje więc wagę 0.8. W czwartym kroku algorytmu sprawdzane jest, czy któryś z węzłów o numerach większych od 2 (czyli od 3 do 10) znajduje się w przekroju $(2|1)$ po stronie $v_s=2$. Wszystkie węzły od 3 do 10 są po stronie węzła 1 (v_r), czyli w czwartym kroku nieprzenoszone są żadne węzły. Po pierwszej iteracji algorytmu drzewo wygląda tak, jak na rys. A.2b. W kolejnej iteracji ($v_s=3$, a jego sąsiad to $v_r=1$) obliczany jest minimalny przekrój $(3|1)=0.8$. Krawędź $\{1,3\}$ otrzymuje wagę 0.8. Ponieważ w przekroju $(3|1)$ wszystkie węzły większe od 3, sąsiadujące z ujściem przekroju ($v_r=1$) – wszystkie od 4 do 10 – znajdują się po stronie źródła przekroju, więc są one w czwartym kroku „przenoszone” z ujścia do źródła, czyli do węzła 3 (rys. A.2c). Podobnie dzieje się w następnych dwóch iteracjach (rys. A.2d i e) – wszystkie węzły o numerach większych niż źródło przenoszone są do kolejnych źródeł przekrojów. W iteracji piątej ($v_s=6$) obliczany jest minimalny przekrój $(6|5)=2.0$. Istnieje kilka minimalnych przekrojów $(6|5)=2.0$. Cztery z sześciu zaznaczone są na rys. A.1 jako: $(6|5)_1$, $(6|5)_2$, $(6|5)_3$ i $(6|5)_4$. Dwa pozostałe to $(\{5,7\} | \{1,2,3,4,6,8,9,10\})$ oraz $(\{1,2,3,4,5\} | \{6,7,8,9,10\})$. Przekroje $(6|5)_1$ oraz $(6|5)_3$ przecinają przekrój poprzedni $(5|4)$. Istnieje także inny przekrój między węzłem 5 oraz 4, na przykład $(5|4)_2$, który nie przecina przekrojów $(6|5)_1$ oraz $(6|5)_3$. Załóżmy, że procedura obliczania minimalnych przekrojów jako przekrój $(6|5)$ zwróciła $(6|5)_1$. W przekroju tym węzeł 7 znajduje się po stronie ujścia (węzła 5), a pozostałe, większe od 6 (czyli od 8 do 10), są po stronie źródła. W związku z tym w czwartym kroku algorytmu tylko węzły od 8 do 10 „przenoszone” są do węzła 6 – rys. A.2f. Węzeł 7 pozostaje. W kroku 4 algorytmu może więc nastąpić „rozgałęzienie” drzewa. W iteracji szóstej ($v_s=7$) obliczany jest minimalny przekrój $(7|5)=0.4$. Ponieważ żaden z węzłów większych od 7 (od 8 do 10) nie sąsiaduje z węzłem 5, więc w kroku 4 nie przenoszone są żadne węzły. Krawędź $\{7,5\}$ otrzymuje jedynie wagę 2 – rys. A.2g. Następne, dwie iteracje powodują przenoszenie wszystkich pozostałych węzłów do kolejnych, źródeł o rosnących numerach (rys. A.2h oraz i). W ostatniej iteracji

($v_s=10$) obliczana jest jedynie wartość minimalnego przekroju. W kroku 4 algorytmu, w ostatniej iteracji, nie ma potrzeby sprawdzania sąsiedztwa, gdyż nie istnieją węzły o numerach większych od ostatniego (największego).



Rys. A.2. Kolejne etapy powstawania drzewa maksymalnych przepływów dla grafu z rys. A.1.



Rys. A.3. Graf z rys. A.1, ze zmienioną numeracją węzłów od 1 do 6.

Na rys. A.3 przedstawiono taki sam graf, jak ten na rys. A.1. Zmieniono w nim jednak numerację węzłów od 1 do 6. Numerację z rys. A.1 umieszczono na rys. A.3 mniejszą czcionką, w nawiasach. Wybrano także inny przekrój (4|3), nie przecinający innych przekrojów (odpowiada on przekrojowi (6|5)₂ z rys. A.1). Przebieg algorytmu EQ jest inny. W pierwszej iteracji algorytmu ($v_s=2$) węzły od 3 do 10 są przenoszone do węzła 2 (po jego stronie przekroju (2|1) się znajdują) – rys. A.4b. W drugiej iteracji ($v_s=3$) obliczany jest minimalny przekrój (3|2)=2.0. Z węzłów większych od 3 węzeł 5 i 6 znajdują się po stronie ujścia (2), więc przy nim pozostają, a węzły 4, 7, 8, 9, 10 znajdują się po stronie źródła przekroju (3|2), czyli „przenoszone” są do węzła 3 – rys. A.4c.

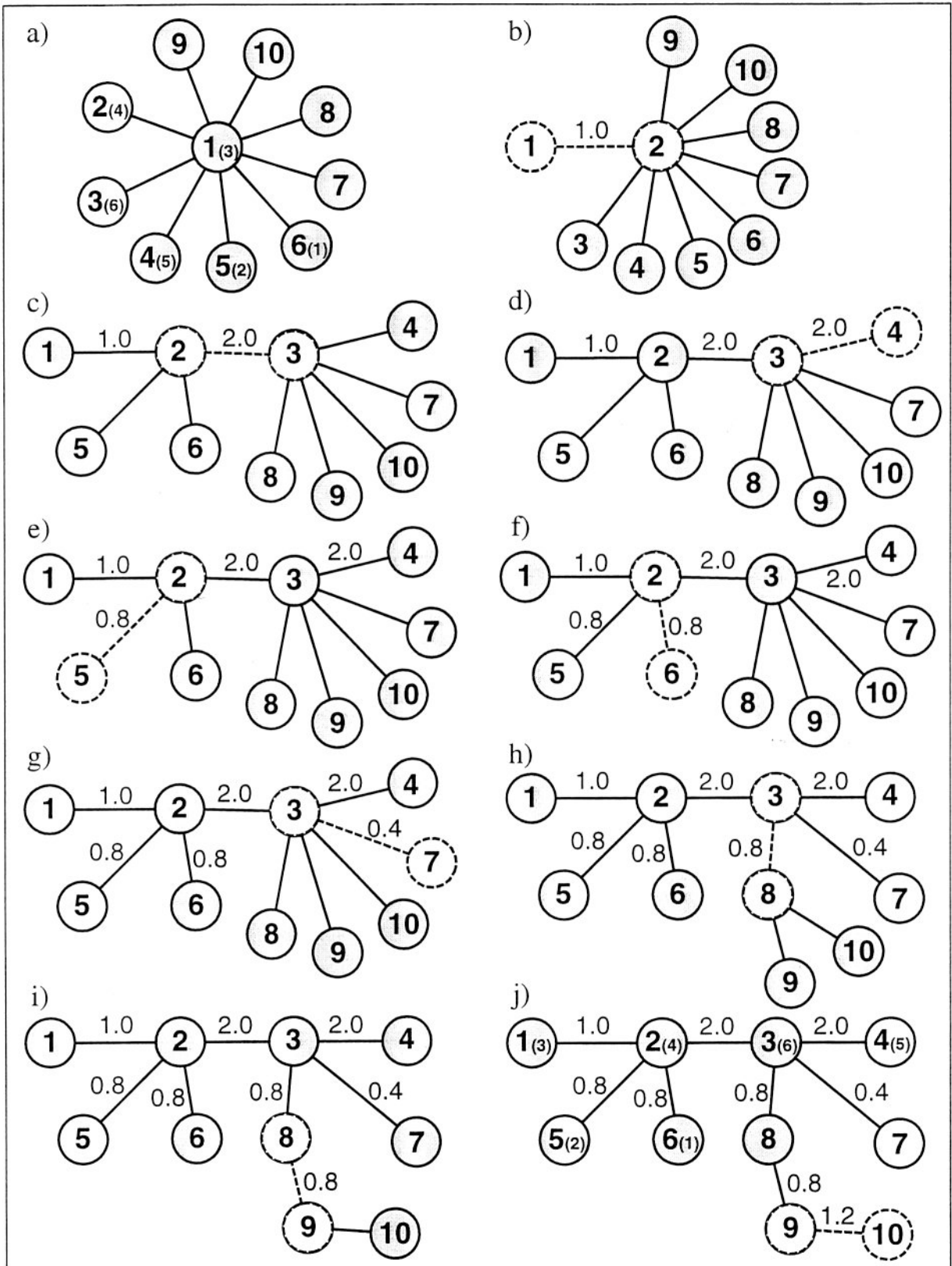
Dla takiego samego grafu (rys. A.1 i A.3), ale ze zmienioną numeracją węzłów¹, w efekcie zastosowania algorytmu powstają inne drzewa maksymalnych przepływów (porównaj rys. A.2j i A.4j). Fakt utworzenia dwóch różnych drzew maksymalnych przepływów, jest związany z arbitralnym wyborem środka gwiazdy (zawsze węzeł 1), w kroku 1 algorytmu, a także możliwością istnienia wielu różnych – i wyboru dowolnego – minimalnych przekrojów, dzielących te same węzły. Dla przykładu z rys. A.1, w pierwszej iteracji jako minimalny przekrój dzielący węzeł 2 i 1, zamiast zaznaczonego na rys. przekroju (2|1), mógłby być wybrany przekrój (3|1), który także jest minimalny dla węzłów 2 i 1. W drugiej iteracji, natomiast, algorytm szukałby minimalnego przekroju między węzłem 3 i 2 i wybrałby zaznaczony na rys. (2|1). Spowodowałoby to zamianę miejscami węzłów 1 i 2 w końcowym drzewie maksymalnych przepływów z rys. A.2j.

Oba drzewa, z rys. A.2j oraz A.4j, są drzewami maksymalnych przepływów, czyli zgodnie z definicją 1.12 maksymalny przepływ pomiędzy dowolną parą węzłów w drzewie jest taki sam, jak pomiędzy odpowiadającą im parą węzłów w grafie.

Drzewa maksymalnych przepływów uzyskiwane algorytmem EQ nie muszą być zarazem drzewami minimalnych przekrojów. Można porównać na przykład drzewa z rys. 1.2b i c. Drugie z tych dwóch drzew (c) powstało w wyniku zastosowania algorytmu EQ do grafu z rys. 1.2a i będąc drzewem maksymalnych przepływów nie jest drzewem minimalnych przekrojów. Także drzewo z rys. A.2j. nie jest drzewem minimalnych przekrojów. Krawędź łącząca węzły 3 oraz 1 tego drze-

¹ Na rys. A.4a oraz j zaznaczono mniejszą czcionką, w nawiasach, numerację węzłów z rys. A.2j.

wa nie spełnia warunku drzewa przekrojów, gdyż jej usunięcie spowoduje powstanie 2 składowych $\{1,2\}$ oraz $\{3,4,5,6,7,8,9,10\}$ z odpowiadającym im przekrojem o pojemności 1.2, który nie jest minimalny dla żadnej pary węzłów.



Rys. A.4. Etapy tworzenia drzewa maksymalnych przepływów dla grafu z rys. A.3. W nawiasach, mniejszą czcionką podano numerację węzłów z rys. A.1.

Krótki, bezpośredni dowód poprawności algorytmu EQ podano w pracy [Gus90]. Inny, oparty na porównaniu z metodą Gomory-Hu, zawiera praca [Gus87].

Niżej przedstawiony program EQ jest realizacją algorytmu EQ Gusfielda. Zmienna p jest w nim N -elementowym wektorem (tablicą), zainicjowaną wartością 1, dla wszystkich jej elementów. Przy każdej iteracji, każdy węzeł o numerze i większym lub równym s jest liściem drzewa, a wartość $p[i]$ wskazuje na numer jego jedynego sąsiada. Początkowo, dla wszystkich węzłów, $p[i]$, wskazuje na węzeł 1 (tworzona jest gwiazda taka, jak na rys. rys. A.2a i A.4a), czyli p jest wypełniona wartością 1. Na wejściu programu podawany jest graf $G^S(V^S, E^S)$, a na wyjściu otrzymujemy zbiór ważonych krawędzi, które tworzą drzewo maksymalnych przepływów $T(V^S, E^T)$ dla grafu $G^S(V^S, E^S)$.

Program EQ

```
for s:=2 to N do
```

```
  begin
```

Wywołaj procedurę obliczania minimalnego przekroju pomiędzy węzłem źródłowym v_s a ujściem v_t gdzie $t:=p[s]$ (czyli jedynym sąsiadem źródła v_s). Niech X będzie zbiorem węzłów po stronie źródła v_s tego przekroju.

Wyprowadź krawędź $\{v_s, v_t\}$ oraz wartość maksymalnego przepływu między v_s i v_t czyli $f_{s,t}$, jako ważoną krawędź będącą elementem drzewa $T(V^S, E^T)$ maksymalnych przepływów.

```
    for i:=s to N do
```

```
      if ( $v_i$  is in  $X$  and  $p[i]=t$ ) then
```

```
         $p[i]:=s$ ;
```

```
  end;
```

Aby otrzymać wszystkie $\frac{N(N-1)}{2}$ wartości maksymalnych przepływów należy utworzyć dodatkową $N \times N$ -wymiarową tablicę F zainicjowaną nieskończonością. Będzie ona przechowywała wartości maksymalnych przepływów między wszystkimi parami węzłów. Następnie, do wyżej przedstawionego programu EQ, należy dodać poniższe linie przed końcową linią "end".

```
F[s,t] := F[t,s] :=  $f_{s,t}$ ;
```

```
for i:=1 to s-1 do
```

```
  if ( $i \neq t$ ) then
```

```
    F[s,i] := F[i,s] :=  $\min(f_{s,t}, F[t,i])$ 
```

Przetwarzanie wykorzystujące graf $G^S(V^S, E^S)$ pojawia się w algorytmie EQ jedynie w momencie obliczania minimalnego przekroju. Algorytm ten może więc być traktowany jako stosujący $N-1$ wywołań procedury „znającej” graf $G^S(V^S, E^S)$. Co więcej, dla danej pary węzłów v_s i v_t , jeśli istnieje więcej niż jeden, minimalny przekrój dzielący v_s i v_t , wtedy procedura obliczania minimalnego przekroju może wybrać dowolny z nich (nie ma znaczenia fakt przecinania się z innymi przekrojami). W związku z tym, procedura obliczania minimalnego przekroju nie musi „szukać” przekroju nieprzecinającego poprzednich, jak w metodzie Gomory-Hu [Hu61, Hu82]. Algorytm EQ zadaje więc dokładnie $N-1$ pytań o minimalny przekrój.

Oprogramowanie zarówno algorytmu EQ jak i jego rozszerzonej wersji umożliwiającej uzyskanie drzewa minimalnych przekrojów dostępne jest w Konrad-Zuse-Zentrum für Informationstechnik w Berlinie, opracowane przez Georga Skorobohatj [Sko]. Zostało ono wykorzystane przy opracowywaniu systemu CIS, użytego w przeprowadzonych eksperymentach. Do obliczania minimalnego przekroju oraz wartości maksymalnego przepływu zastosowano w nim algorytm Goldberga i Tarjana, przedstawiony pracach [Gol86, Gol88].

ZAŁĄCZNIK B. LISTA PYTAŃ TESTOWYCH

Nr pytania / kolekcji	Lista słów kluczowych, połączonych operatorem logicznym „i”	Imię i nazwisko eksperta
1	median distance	T.Nguen
2	temporal database consensus	T.Nguen
3	information retrieval ranking	J.Baliński
4	ergodic markov chain	J.Baliński
5	interface agents	J.Sobecki
6	multimedia retrieval	J.Sobecki
7	TREC	P.Kazienko
8	roles object oriented databases	D.Król
9	dynamic object oriented databases	D.Król
10	bayesian networks information retrieval probabilistic reasoning	A.Zgrzywa
11	neural networks information retrieval	A.Zgrzywa
12	JSP	B.Trawiński
13	JDBC	B.Trawiński
14	speech recognition algorithm	E.Kukla
15	multimedia intelligent tutoring	E.Kukla
16	music information retrieval	J.Wójcik
17	telecommunication trade fairs	J.Wójcik
18	information retrieval agent	A.Piasecka
19	query expansion user profile	A.Piasecka
20	software agent intelligent	D.Dudek
21	time management	D.Dudek
22	distributed information management intranet	M.Lenar
23	multiprocessor mainboard celeron	M.Lenar
24	avatars virtual reality	K.Choroś
25	translation semantic analysis statistical method	A.Siemiński

Tab. B.1. Lista pytań / kolekcji wykorzystywanych podczas eksperymentów.

ZAŁĄCZNIK C. FORMULARZ OCENY HIERARCHII

OCENA WYNIKÓW GRUPOWANIA

Nr kolekcji: **9**..... Nr pytania: **322**..... Data odpowiedzi: **22.02.00** Nr strony: **1/1**

Słowa kluczowe (pytanie): **dynamic object oriented databases**

Imię i nazwisko eksperta: **Dariusz Król**

Parametry: τ **0.07**, lev^{max} : **4**, α : **5**, β : **5**, γ : **5**, μ : **0.5**

Lp	Numer dokumentu	Do grupy szczegółowej (numer starej → nowej grupy / ile poziomów)	Do grupy ogólniejszej (numer starej → nowej grupy / ile poziomów)	Do innej gałęzi / innej hierarchii (numer starej → nowej gr./ile poz)	Do grupy dok. izolowanych (numer starej grupy / ile poz.)	Z grupy dok. izolowanych (numer nowej grupy / ile poz.)
1.	2924	usunięta grupa (6)		6 → krótka/5	nowa grupa i litera - (9)	
2.	2941			6 → krótka/5		
3.	2944			6 → krótka/5		
4.	2949					
5.	2907	5 → 7/1	6 → 5/1			
6.	2948		3 → 1/1			
7.	2928	3 → 5/1				
8.	2914	3 → 5/1				
9.	2906				3 → 1/2	
10.	2887	3 → 5/1				
11.	2970				1 → 1/1	
12.	2916				1 → 1/1	
13.	2900	1 → Allepro/1	Nowa grupa „Allepro” (8)			
14.	2915	1 → Allepro/1				
15.	2934					→ 1/1
16.	2942					→ 1/1
17.	2965					→ 1/1
18.	2971					→ 1/1
19.	2974					→ 1/1
20.	2980					→ 1/1
21.						
22.						
23.						
24.						
25.						
26.						
27.						
28.						

	Dok.	Przesun.	Dok.	Przesun.	Dok.	Przesun.	Dok.	Przesun.	Dok.	Przesun.
Suma:	6	6	2	2	3	15	3	4	6	6

Liczba odpowiedzi metawyszukiwarki: **112**..... Liczba dokum. w kolekcji (N): **77**.....

Liczba różnych terminów w kolekcji: **4656**..... Średnio różnych terminów/dok: **157.6**.....

Średnio odsyłaaczy/dok: **25.8**..... Liczba par z $S^{cs}(d_i, d_j) > 0$: **62**.....

Liczba dok. z $S^{cs}(d_i, d_j) > 0$: **28**..... Liczba krawędzi grafu: **224**.....

Liczba, przed oceną (bez dok.izolow.), hierarchii: **1**....., grup: **7**....., liści: **4**.....

Liczba nowych grup: **2 (1 hierarchia)**..... Liczba usuniętych grup: **1**.....

Liczba dokumentów izolowanych przed oceną: **10**....., po ocenie: **7**.....

Rys. C.1. Wypełniony formularz oceny grupowania dla kolekcji 9 (hierarchia z rys. 6.1).