



Imię i nazwisko autora rozprawy: **mgr inż. Kanstantsin Myslitski**
Dyscyplina naukowa: Informatyka Techniczna i Telekomunikacja

ROZPRAWA DOKTORSKA

Tytuł rozprawy w języku polskim: **Zastosowanie metod transformacji grafu topologii sieci teleinformatycznej w wyznaczaniu niezawodnych tras transmisji ukierunkowane na redukcję czasu obliczeń**

Tytuł rozprawy w języku angielskim: **Application of methods of network topology graph transformation in calculation of reliable transmission paths aimed at reducing the computational time**

Promotor <i>podpis</i>	Drugi promotor <i>podpis</i>
Dr hab. inż. Jacek Rak, prof. PG	
Promotor pomocniczy <i>podpis</i>	Kopromotor <i>podpis</i>



Niniejszą pracę dedukuję mojej Mamie – Śp. Natalii Myslitskiej.



Za powstanie tej pracy dziękuję mojemu promotorowi Dr. hab. inż. Jackowi Rakowi, prof. PG za wspólną pracę, za całą przekazaną mi przez te lata wiedzę oraz wszelką pomoc.

Chciałbym również podziękować całemu zespołowi pracowników Katedry Teleinformatyki Politechniki Gdańskiej za kształcenie oraz osobisty rozwój.

Osobne, równie ważne, podziękowania składam na ręce mojej żony Mariany Myslitskiej, na którą zawsze mogłem liczyć, która mnie wspierała przez cały ten okres.



Streszczenie

Od początku powstania sieci teleinformatycznych obserwujemy nieustanny wzrost ich znaczenia praktycznie w każdym obszarze działalności ludzkiej. Sieci te odgrywają ważną rolę nie tylko w kontekście zastosowań, dla których zostały pierwotnie opracowane (czyli transmisji/dostępu do informacji tekstowej). Umożliwiają ponadto realizację zadań w wielu kluczowych obszarach, jak np. w odniesieniu do ochrony zdrowia (transmisja danych medycznych), pracy zdalnej (np. w sytuacji pandemii COVID-19 czy podczas ekstremalnych warunków pogodowych), zapewniania bezpieczeństwa obywateli (np. dając możliwość rozpowszechniania informacji dotyczących zagrożeń atmosferycznych, stanów kryzysowych związanych z pożarami, powodzią, itp.), stanowiąc tym samym ważny element infrastruktury krytycznej.

Obserwowany wzrost ilości danych transmitowanych w sieciach teleinformatycznych nie byłby możliwy bez równoległego rozwoju technologii, w szczególności w obszarze sieci szkieletowych dzięki zastosowaniu techniki zwielokrotnienia z podziałem długości fal – WDM (ang. Wavelength Division Multiplexing) i jej kolejnych wariantów (np. DWDM – ang. Dense Wavelength Division Multiplexing) oferujących na jednym łączy optycznym nawet kilkadziesiąt kanałów transmisji, każdy o przepustowości rzędu kilkudziesięciu Gb/s. W tym kontekście, awaria nawet pojedynczego elementu sieci może być katastrofalna w skutkach. Niedostępność łącza/węzła obsługującego wiele tysięcy strumieni ruchu użytkowników końcowych może prowadzić zarówno do istotnych strat finansowych właścicieli usług sieciowych oraz operatorów, jak i zwiększyć stopień zagrożenia zdrowia czy życia ludzi (np. w sytuacji niemożności transmisji informacji w przypadku usług wspomagających prowadzenie operacji ratunkowych).

W literaturze dostępnych jest wiele metod ochrony transmisji przed awarią elementów sieci bazujących na koncepcji tras zapasowych (ang. backup paths), tj. alternatywnych tras transmisji nieposiadających wspólnych elementów tranzytowych z ochranianymi trasami podstawowymi. W obliczu coraz częstszych scenariuszy awarii wielokrotnych, ochrona przed jednoczesną awarią kilku (k) elementów sieci typowo implikuje konieczność ustanowienia

k tras zabezpieczających wykorzystywanych głównie w krótkim okresie pomiędzy wystąpieniem awarii a fizyczną naprawą uszkodzonych elementów sieci (co jest mało efektywne dla operatorów z punktu widzenia stopnia zapotrzebowania na przepustowości łączy sieci).

W odniesieniu do wielu przypadków awarii klasyfikowanych jako awarie wielokrotne (scenariuszy jednoczesnych awarii k elementów sieci) istotne jest, że znaczna ich część stanowi w rzeczywistości sekwencję awarii pojedynczych, które wystąpiły w krótkich odstępach czasu zanim możliwa była fizyczna naprawa uszkodzonych elementów. W takich przypadkach zastosowanie algorytmów doboru tras, które byłyby w stanie odpowiednio szybko wyznaczyć pary tras: podstawową i zabezpieczającą dla każdego żądania zaraz po awarii danego elementu sieci mogłoby w istocie sprowadzić zadanie ochrony przed awarią wielokrotną do problemu ochrony przed kolejnymi awariami pojedynczymi (angażującego jedynie pojedyncze trasy zabezpieczające zamiast k tras zabezpieczających). To z kolei skutkowałoby istotnie zmniejszonym zapotrzebowaniem na zasoby sieci (przepustowości łączy) w celu instalacji tras. Mimo dostępności w literaturze szeregu rozwiązań z zakresu doboru tras niezawodnych, niewiele z nich jest ukierunkowanych na jednoczesną redukcję czasu obliczeń.

Celem pracy jest zaprezentowanie autorskich metod zapewniania niezawodności transmisji wieloskokowej (od krańca do krańca) przy wykorzystaniu proponowanych technik transformacji ukierunkowanych na ograniczenie czasu wyznaczania tras, jak i na umożliwienie obsługi przez sieć większej liczby żądań dzięki redukcji zapotrzebowania na zasoby sieci w scenariuszach ochrony przed awarią wielokrotną oraz opracowaniu mechanizmów doboru tras ukierunkowanych na redukcję współczynnika blokowania żądań. Weryfikacja charakterystyk metod opisanych w niniejszej rozprawie była możliwa dzięki wykorzystaniu zaimplementowanego autorskiego środowiska badań symulacyjnych i miała na celu wykazanie tezy rozprawy mówiącej o tym, że zastosowanie technik transformacji grafu topologii sieci umożliwia redukcję czasu niezbędnego do wyznaczenia tras niezawodnych od krańca do krańca.

Niniejsza rozprawa obejmuje pięć rozdziałów numerowanych, dziesięć aneksów oraz trzy części nienumerowane (spis symboli, spis skrótów, wykaz

literatury). Rozdział 1 stanowiący wstęp zawiera zarys charakterystyk optycznych sieci transportowych i uwypukla podatność elementów sieci rozległych na awarie. Przedstawia on także cel pracy oraz najważniejsze założenia metod proponowanych w rozprawie. W rozdziale 2 zostały opisane dostępne w literaturze strategie zapewniania ochrony transmisji w sieciach optycznych i ich podział ze względu na chwilę wyznaczania tras zabezpieczających, zasięg ochrony tras zabezpieczających, scenariusze awarii elementów sieci oraz sposób rezerwacji zasobów sieci na potrzeby tras zabezpieczających. Rozdział 2 nasświetla także problemy z zakresu wdrożenia tych rozwiązań w optycznych sieciach rozległych oraz uzasadnia wybór strategii ochrony transmisji przyjętej w dalszej części rozprawy.

Rozdział 3 prezentuje autorską koncepcję doboru tras ukierunkowaną na redukcję prawdopodobieństwa blokowania żądań poprzez dążenie do zrównoważenia obciążenia łączy sieci. Rozwiązania prezentowane w rozdziale 3 są zainspirowane założeniami znanymi z mechaniki kwantowej operującej koncepcją kwantu – najmniejszej jednostki, która może istnieć jednocześnie w kilku lokalizacjach przestrzeni z określonym prawdopodobieństwem. W szczególności, metoda zaprezentowana w rozdziale 3.1 adaptuje właściwości kwantu w propozycji metryki kwantowej uwzględniającej stopień prognozowanych konfliktów o zasoby sieci (a więc analizuje a priori stopień wzrostu ryzyka całkowitego wykorzystania zasobów danego łącza) po jego ewentualnym wykorzystaniu przez aktualnie wyznaczaną trasę. Wyniki badań symulacyjnych wskazują na istotną przewagę proponowanego rozwiązania (nawet do 53.6%) nad rozwiązaniem referencyjnym w kontekście stopnia blokowania żądań. W rozdziale 3.2 ukazana jest modyfikacja metody z rozdziału 3.1 ukierunkowana na redukcję czasu obliczeń kwantów (tras reprezentowanych w postaci probabilistycznej). Stanowi ona w istocie pewien kompromis pomiędzy wartością stopnia blokowania żądań (tutaj lepszą o maksymalnie 8.24% w porównaniu z rozwiązaniem referencyjnym) a czasem obliczeń.

Głównym celem rozdziału 4 jest przedstawienie autorskiej metody szybkiego wyznaczania tras transmisji, która byłaby możliwa do wykorzystania w scenariuszu sekwencji awarii pojedynczych (występujących jedna po drugiej w krótkich odstępach czasu), aby wyznaczyć trasy transmisji po wystąpieniu danej

awarii zanim wystąpi kolejna awaria. Proponowana metoda w celu ograniczenia czasu obliczeń tras wykorzystuje autorską koncepcję tzw. mapy komórkowej będącej transformacją struktury grafu topologii sieci, w której wierzchołki reprezentują nie węzły sieci lecz cykle bazowe grafu topologii sieci. Jako że dla grafu topologii sieci liczba cykli bazowych jest zazwyczaj mniejsza od liczby węzłów tej sieci, zadanie wyznaczenia par rozłącznych tras (podstawowej i zabezpieczającej) operując na strukturze mapy komórkowej trwa znacząco krócej niż w przypadku wykorzystania klasycznego grafu topologii sieci. Istotnie, dowodzą tego wyniki symulacji, podczas których uzyskana została redukcja czasu obliczeń nawet o 73.54% w porównaniu z podejściem referencyjnym.

Rozdział 5 zawiera podsumowanie pracy wraz z głównymi wnioskami oraz wskazaniem potencjalnych obszarów dalszej pracy badawczej. Szereg załączników rozprawy zawiera rozszerzenie jej wybranych aspektów, m.in. w zakresie dowodów poprawności proponowanych algorytmów, analizy złożoności obliczeniowej metod oraz prezentacji szczegółowych wyników symulacyjnych oraz rozszerzeń autorskich metod w wybranych kontekstach.

Uzyskane wyniki dowodzą słuszności tezy rozprawy i potwierdzają przydatność metod transformacyjnych w uzyskaniu istotnej redukcji czasu wyznaczania tras transmisji, jak i redukcji ryzyka blokowania żądań.

Summary

Since the emergence of communication networks, we have observed a continuous increase in their importance in practically every area of human activity. These networks play an important role not only in the context of utilization for which they were originally developed (i.e. transmission / access to text information). Moreover, they enable the implementation of tasks in many key areas, such as in relation to healthcare (transmission of medical data), remote work (e.g. in the situation of the COVID-19 pandemic or during extreme weather conditions), ensuring the safety of citizens (e.g. enabling the dissemination of information on atmospheric hazards, crisis states related to fires, floods, etc.), thus forming an important element of the critical infrastructure.

The observed increase in the amount of data transmitted by communication networks would not be possible without the parallel development of technology, in particular in the area of backbone networks, thanks to the application of the WDM (Wavelength Division Multiplexing) technique and its subsequent variants (e.g. DWDM – Dense Wavelength Division Multiplexing) offering up to several tens of transmission channels per single optical link, each channel with a capacity of several tens of Gb/s. In this context, failure of even a single network element can be catastrophic. Unavailability of a link / node serving many thousands of end-user traffic flows may lead to significant financial losses for service owners and operators, and increase the degree of threat to human health or life (e.g. in the event of inability to transmit information in the case of services supporting emergency operations).

In the literature, there are many methods for protection of transmission against failures of network elements available based on the concept of backup paths, i.e. alternative transmission paths that do not have common transit elements with protected primary paths. In the face of more and more frequent scenarios of multiple failures, protection against a simultaneous failure of several (i.e. k) network elements typically implies the need to establish k backup paths used mainly in the short period between the failure occurrence and the physical repair of the damaged network elements (which is ineffective for operators in terms of the amount of link capacity reserved for backup path purposes).

Concerning many failure cases classified as multiple failures (i.e. scenarios of simultaneous failures of k network elements), it is important to notice that a significant part of them are, in fact, sequences of single failures that occurred at short intervals one after another before the failed components could be physically repaired. In such cases, the use of path calculation algorithms that would be able to quickly determine the pairs of paths: the primary and the backup one for each request immediately after the failure of a given network element, could in fact reduce the task of protection against multiple failures to the problem of protection against subsequent single failures (involving only single backup paths instead of sets of k backup paths). This, in turn, would result in a significantly lower amount of link capacity needed for the installation of transmission paths. Despite the availability in the literature of a number of solutions concerning resilient routing, they are not aimed in parallel at reducing the calculation time.

The aim of this dissertation is to present the proposals of methods of ensuring the resilience of multi-hop (end-to-end) transmission with the use of the proposed transformation techniques aimed at reducing the time of path calculation, as well as enabling the network to serve a greater number of demands by reducing the ratio of network resource utilization in scenarios of protection against multiple failures and development of path calculation mechanisms aimed at reducing the blocking probability of demands. The verification of the characteristics of the methods described in this dissertation was possible owing to the use of the simulation research environment implemented for this purpose, and was aimed at proving the thesis of the dissertation stating that the use of network topology graph transformation techniques can enable the reduction of the time necessary to determine end-to-end resilient communication paths.

This dissertation covers five numbered chapters, ten annexes and three unnumbered parts (i.e. the list of symbols, the list of abbreviations and the list of references). Chapter 1, being an introduction, outlines the characteristics of optical transport networks and highlights the vulnerability of wide area network elements to failures. It also presents the purpose of this work and the most important assumptions behind the methods proposed in the dissertation. Chapter 2 describes the strategies available in the literature for ensuring transmission protection in optical networks and their classification based on the moment of setting up the backup paths, the scope of protection of backup paths, scenarios of failures of network elements and the method of reserving the network resources necessary for

the installation of backup paths. Chapter 2 also highlights the problems related to the deployment of these solutions in optical wide area networks and justifies the choice of the transmission protection strategy adopted in the remaining part of this dissertation.

Chapter 3 presents the original concept of path calculation aimed at reducing the probability of blocking the demands by striving to balance the load on network links. The solutions presented in Chapter 3 are inspired by the assumptions known from quantum mechanics that operate with the concept of a quantum – the smallest unit that can exist simultaneously in several locations in space with a certain probability. In particular, the method presented in Chapter 3.1 adapts the properties of the quantum in the proposal of a quantum metric of a link cost, taking into account the probability of potential conflicts concerning the assignment of network resources to communication paths (and thus analyzes a priori the degree of increase in the risk for a network link to become fully utilized), in the case a given link becomes utilized by the currently calculated communication path. The results of simulation studies show a significant advantage of the proposed solution (even up to 53.6%) over the reference solution in terms of the blocking probability of demands. Chapter 3.2 shows a modification of the method from Chapter 3.1 aimed at reducing the time of computation of quanta (i.e. routes represented in a probabilistic form). In fact, the modified method provides a trade-off between the value of the request blocking rate (here, up to a maximum of 8.24% better than the reference solution) and the computation time.

The main goal of Chapter 4 is to propose a method of fast calculation of communication paths, which could be used in the scenario of a sequence of single failures (occurring one after another in short intervals) to determine transmission paths after the occurrence of a given failure but before the next failure occurs. The proposed method, in order to reduce the time of path calculations, uses the original concept of the so-called “cell map” which is a result of a certain transformation of the structure of a network topology graph, in which the vertices represent not the network nodes but the base cycles of the network topology graph. Since for the network topology graph the number of base cycles is usually lower than the number of nodes of this network, the task of determining pairs of disjoint routes (i.e. the primary path and the backup path) while operating on the structure of the cell map takes less time than in the case of using the classical network topology graph. Indeed, it was proved by the simulation results, during which the

computation time was reduced by up to 73.54%, compared to the reference approach.

Chapter 5 contains a summary of the work along with the main conclusions and an indication of potential areas for further research. A set of annexes to this dissertation includes an extension of its selected aspects, such as proofs of the correctness of the proposed algorithms, analysis of the computational complexity of methods and the presentation of detailed simulation results and extensions of the proposed methods in selected contexts.

The obtained results prove the validity of the thesis of the dissertation and confirm the usefulness of the proposed transformation methods in obtaining a significant reduction in the time of calculation of communication paths as well as reducing the risk of blocking of demands.

Spis Treści

Spis symboli _____	17
Spis skrótów _____	23
1 Wstęp _____	25
2 Dostępne metody i tryby ochrony transmisji przed awariami w optycznych sieciach transportowych _____	33
2.1 Zastosowanie metod ochrony transmisji w optycznych sieciach transportowych _____	37
2.2 Wybór sposobu ochrony transmisji i modelu routingu _____	42
3 Zastosowanie metod transformacyjnych podczas doboru tras ukierunkowane na redukcję współczynnika blokowania żądań _____	51
3.1 Metoda redukcji prawdopodobieństwa blokowania żądań przy wyznaczaniu tras w sieciach o ograniczonych zasobach inspirowana teorią kwantową _____	52
3.1.1 Wprowadzenie _____	52
3.1.2 Podstawy nowej metody _____	57
3.1.3 Wyznaczenie kwantu (q) dla danego żądania _____	64
3.1.4 Wyznaczenie wektora prognozowanych konfliktów _____	67
3.1.5 Dobór proporcji wartości współczynników α oraz β _____	70
3.1.6 Założenia symulacyjne _____	76
3.1.7 Wyniki symulacji _____	80
3.1.8 Wnioski _____	83
3.2 Metoda redukcji czasu doboru tras ukierunkowana na zrównoważenie obciążenia w sieciach rozległych _____	84
3.2.1 Analiza możliwości modyfikacji algorytmu budowania kwantów ukierunkowanej na redukcję czasu obliczeń _____	85
3.2.2 Podstawy metody kwantowego doboru tras opartego na budowaniu kwantów ze zbioru k najkrótszych tras _____	88
3.2.3 Dobór współczynnika krzywizny funkcji odwrotnej proporcjonalności (γ) oraz liczby najkrótszych tras (k) _____	94
3.2.4 Wyniki badań symulacyjnych dla proponowanej metody _____	96
3.3 Wnioski _____	97
4 Zastosowanie metod transformacyjnych do opracowania metody niezawodnego doboru tras o istotnie ograniczonym czasie obliczeń _____	99
4.1 Znane metody doboru par węzłowo-rozłącznych tras _____	100
4.1.1 Algorytm Suurballe-Tarjana _____	100
4.1.2 Algorytm Bhandariego _____	103

4.1.3 Wybór podejścia referencyjnego _____	105
4.2 Metoda transformacyjna dla redukcji czasu wykrycia pary węzłowo-rozłącznych tras w sieciach teleinformatycznych _____	106
4.2.1 Podstawy nowej metody _____	106
4.2.2 Aktualizacja kosztów elementów mapy komórkowej ____	121
4.2.3 Wyniki symulacji _____	122
4.3 Wnioski _____	124
5 Podsumowanie _____	127
Aneks 1 – Udowodnienie poprawności działania algorytmu DFSm __	131
Aneks 2 – Złożoność algorytmiczna metody doboru tras kwantowych	137
Aneks 3 – Wyniki badań efektywności metody kwantowej _____	151
Aneks 4 – Dobór ilorazu a/β _____	159
Aneks 5 – Wyniki symulacji algorytmu kwantowego _____	167
Aneks 6 – Analiza złożoności algorytmicznej PPCB _____	179
Aneks 7 – Warunek w wierszu 14 PPCB _____	185
Aneks 8 – Modyfikacje PPCB _____	187
Aneks 9 – Budowanie kwantów w metodzie PPCB _____	191
Aneks 10 – Sposoby szybkiej aktualizacji mapy komórkowej po zmianie topologii sieci _____	195
Spis bibliografii _____	197

Spis symboli

A	–	Zbiór cykli bazowych, które mają obliczone przez PPCB cykle S_i
a	–	Awaria elementu sieci (węzła/łącza).
B	–	Zbiór cykli bazowych, które mają wykonane przez PPCB próby połączenia z sąsiednimi cyklami bazowymi
$b(e_i)$	–	Przepustowości łącza e_i .
C	–	Cykl bazowy w grafie G .
$c(d)$	–	Przepustowość wymagana dla transmisji informacji dla żądania d
$c(e_i)$	–	Metryka kosztu łącza e_i
$c(P)$	–	Koszt ścieżki w grafie G ; metryka kosztu trasy w sieci; liczba przeskoków (dla przypadku jednostkowych kosztów łączy)
$c_a(e_i)$	–	Składnik metryki kwantowej oparty na prognozowanych danych
$c_b(e_i)$	–	Składnik metryki kwantowej oparty na aktualnych danych
D	–	Zbiór żądań instalacji tras transmisji w sieci
d	–	Żądanie instalacji trasy
E	–	Zbiór krawędzi grafu topologii sieci;
E^M	–	Zbiór wspólnych krawędzi dla par cykli bazowych grafu G ; zbiór krawędzi mapy komórkowej G^M
$E^M(S)$	–	Zbiór wszystkich krawędzi grafu mapy komórkowej pomiędzy cyklami bazowymi zawartymi w cyklu S
E^θ	–	Zbiór krawędzi Mapy Motylkowej G^θ
$\overline{ E(P) }$	–	Średnia liczba przeskoków w trasach
e^M	–	Krawędź Mapy Komórkowej G^M

$e(v_i, v_j)$	–	Krawędź grafu G pomiędzy wierzchołkami v_i, v_j reprezentująca łącze sieciowe pomiędzy węzłami v_i, v_j
e_i	–	Krawędź grafu G reprezentująca łącze sieci
G	–	Graf topologii sieci
G^M	–	Struktura Mapy Komórkowej
G^O	–	Struktura Mapy Motylkowej
$g(e_i)$	–	Część przepustowości łącza e_i , która jest zarezerwowana dla obsługi żądań
H	–	Zbiór modeli ruchu sieciowego
h_i	–	Model ruchu sieciowego
l_{ij}	–	Łuk grafu skierowany od węzła v_i do węzła v_j
M	–	Macierz sąsiedztwa węzłów
$n(X)$	–	Liczba elementów pamięci wymaganych dla przechowywania elementu X
P	–	Ścieżka w grafie G reprezentująca trasę w sieci
$P(e_i)$	–	Ścieżka (trasa) w grafie G (sieci) zawierająca krawędź (łącze) e_i .
$P(v_i, v_j)$	–	Ścieżka (trasa) w grafie G (sieci) pomiędzy wierzchołkami (węzłami) v_i oraz v_j
$P(v_s, *)$	–	Ścieżka (trasa) w grafie G (sieci) pomiędzy zdefiniowanym wierzchołkiem (węzłem) źródłowym v_s i niezdefiniowanym wierzchołkiem (węzłem) docelowym
$P(*, v_t)$	–	Ścieżka (trasa) istniejąca w grafie G (sieci) pomiędzy zdefiniowanym wierzchołkiem (węzłem) docelowym v_t i niezdefiniowanym wierzchołkiem (węzłem) źródłowym
P_o	–	Ścieżka podstawowa w grafie G reprezentująca trasę podstawową
P_{opt}	–	Ścieżka optymalna pod względem jej kosztu w grafie G reprezentująca trasę optymalną
P'	–	Ścieżka alternatywna (zapasowa); trasa alternatywna (zapasowa)
Q	–	Zbiór wszystkich kwantów sieci

q	–	Kwant; trasa kwantowa.
R	–	Zbiór ścieżek w grafie G ; zbiór tras w sieci.
$R(v_i, v_j)$	–	Zbiór ścieżek w grafie G (tras w sieci) istniejących pomiędzy wierzchołkami (węzłami) v_i oraz v_j .
$R(e_i)$	–	Zbiór ścieżek w grafie G (tras w sieci) zawierających krawędź (łączy) e_i .
R^*	–	Zbiór rozłącznych ścieżek w grafie; zbiór rozłącznych tras w sieci.
$r(\rho_B)$	–	Przewaga proponowanej metody nad podejściem referencyjnym pod względem prawdopodobieństwa blokowania żądań
S	–	Cykl w grafie G ; zbiór cykli bazowych
t_1	–	Czas wykrycia awarii
t_2	–	Czas lokalizacji awarii (czas niezbędny w celu wykrycia awarii elementu, który uległ awarii)
t_3	–	Czas oczekiwania przez mechanizmy wykrycia awarii wyższych warstw sieci dotyczący zakończenia akcji naprawczych po awarii przez mechanizmy niższych warstw (istotne dla sieci warstwowych)
t_4	–	Czas dostarczenia informacji o awarii do węzła sieci odpowiedzialnego za odtwarzanie transmisji po awarii
t_5	–	Czas naprawienia transmisji. Najbardziej czasochłonna część, która wymaga zależnie od modelu: wykrycia zabezpieczającej trasy, instalacji zabezpieczającej trasy w sieci, przełączenia strumienia danych na trasę zabezpieczającą
t_6	–	Czas synchronizacji. Czas niezbędny w celu weryfikacji trasy zabezpieczającej i przekierowania transmisji do węzłów docelowych poprzez trasę zabezpieczającą
t_7	–	Czas normalizacji. Czas niezbędny do przeprowadzenia fizycznej naprawy uszkodzonego elementu sieci i powrotu do konfiguracji tras transmisji sprzed wystąpienia awarii
$t(a)$	–	Czas trwania awarii a
$t_i(p_0)$	–	Opóźnienie transmisji dla trasy podstawowej i -tego żądania.
$t_i(p')$	–	Opóźnienie transmisji dla trasy alternatywnej i -tego żądania.

t_{rt}	–	Czas przywrócenia transmisji po awarii elementu sieci
$\overline{t_r(P)}$	–	Średni czas potrzebny na wyznaczenie tras dla wszystkich żądań
\overline{U}	–	Średnia wartość współczynnika wykorzystania zasobów sieci
V	–	Zbiór wierzchołków grafu (węzłów topologii sieci)
$V(v_i)$	–	Zbiór wierzchołków grafu (węzłów topologii sieci) sąsiednich względem wierzchołka (węzła) v_i
V^M	–	Zbiór wierzchołków reprezentujących cykle bazowe grafu G
$V^{\mathcal{O}}$		Zbiór węzłów Mapy Motylkowej
V_s^M	–	Zbiory źródłowych cykli bazowych
V_t^M	–	Zbiory docelowych cykli bazowych
$V^M(S)$	–	Zbiór wszystkich cykli bazowych zawartych w cyklu S
v	–	Wierzchołek grafu topologii sieci; węzeł sieci; obiekt „węzeł” Mapy Komórkowej
v_s	–	Wierzchołek źródłowy w grafie G reprezentujący źródłowy węzeł transmisji
v_t	–	Wierzchołek docelowy w grafie G reprezentujący docelowy węzeł transmisji
v^M	–	Węzeł Mapy Komórkowej G^M
$v_s(d)$	–	Źródłowy węzeł żądania d
$v_t(d)$	–	Docelowy węzeł żądania d
W	–	Wektor prognozowanych konfliktów o zasoby przepustowości sieci pomiędzy żadaniami na instalację tras.
$w(e_i)$	–	Element wektora prognozowanych konfliktów W dla łącza e_i określający poziom prognozowanego konfliktu dla łącza e_i .
Z	–	Zbiór tras wykrytych algorytmem DFSm
α	–	Współczynnik prognozowanych konfliktów w sieci
β	–	Współczynnik wykorzystanych zasobów sieci
Φ	–	Ciąg numerów cykli

γ	–	Współczynnik krzywizny funkcji odwrotnej proporcjonalności stosunku prawdopodobieństw optymalności tras
λ	–	Długość fali stosowana w transmisji
Θ	–	Zbiór motylkowy – dwa cykle i trasa pomiędzy nimi
$p(X)$	–	Prawdopodobieństwo zdarzenia X
$\overline{\rho_B}$	–	Prawdopodobieństwo blokowania żądań instalacji tras
Ω	–	Obciążenie sieci
Ξ	–	Wektor binarnej reprezentacji cykli
ψ	–	Numer cyklu



Spis skrótów

AON	–	(ang. All-Optical Network) sieć z w pełni optyczną transmisją
BFS	–	Algorytm Breadth-first search
CZRS	–	Centrum zarządzania ruchem sieciowym
DFS	–	Algorytm Depth-first search
DFSm	–	Zmodyfikowany algorytmu Depth-first search
DWDM	–	(ang. Dense Wavelength Division Multiplexing) technika gęstego zwielokrotnienia falowego
OFDM	–	(ang. Orthogonal Frequency-Division Multiplexing) zwielokrotnianie z ortogonalnym podziałem częstotliwości
PPCB	–	Procedura Połączenia Cykli Bazowych
QoS	–	(ang. Quality of Service) jakość usług
RWA	–	(ang. Routing and Wavelength Assignment problem) problem doboru tras i przydziału długości fali
SDH	–	(ang. Synchronous Digital Hierarchy) technika Synchronicznej Hierarchii Systemów Cyfrowych
SDN	–	(ang. Software-Defined Networks) sieci programowalne
SWANET	–	(ang. Self-routed Wavelength-Addressable Network) sieć z własnym przekierowywaniem opartym na długościach fali
TDM	–	(ang. Time Division Multiplexing) technika zwielokrotnienia w dziedzinie czasu
VoIP	–	(ang. Voice-over-IP) rozmowa przez IP
WDM	–	(ang. Wavelength Division Multiplexing) technika zwielokrotnienia w dziedzinie długości fali



1 Wstęp

Od czasu powstania pierwszych sieci komputerowych jak ARPANET w 1969 r. [1] nieustannie rośnie znaczenie sieci teleinformatycznych w praktycznie wszystkich obszarach działalności ludzkiej. Sieci te odgrywają wiodącą rolę nie tylko w przypadku zastosowań, dla których zostały pierwotnie opracowane (czyli transmisji/dostępu do informacji tekstowej), ale również w przekazie informacji innego (tj. pierwotnie niezakładanego) typu, jak m.in. transmisji głosu, obrazów/wideo, obsługi ruchu strumieniowego. W początkowym etapie rozwoju sieci teleinformatycznych, dostęp do zasobów sieci Internet był powszechnie zapewniany poprzez połączenie telefoniczne wykorzystujące zasoby infrastruktury sieci telekomunikacyjnej. Obecne sieci teleinformatyczne przejęły nawet realizację klasycznych zadań sieci telekomunikacyjnych – np. klasycznych usług telefonicznych (transmisji mowy). Trudno wyobrazić sobie prowadzenie działalności gospodarczej, naukowej, finansowej, handlu itd. w XXI wieku bez nieograniczonego dostępu do globalnej infrastruktury teleinformatycznej.

Wzrost ilości danych transmitowanych w sieciach teleinformatycznych nie byłby możliwy bez równoległego rozwoju technologii transmisji, a w szczególności bez zwiększenia przepływności łączy sieci. W tym zakresie, około 25 lat temu przełomowym było zastosowanie techniki zwielokrotnienia z podziałem długości fal – WDM (ang. Wavelength Division Multiplexing) [2]-[4] dla transmisji wykorzystującej łącza światłowodowe. W odróżnieniu od wcześniej stosowanych metod (takich jak Synchronous Digital Hierarchy (SDH) [5]), główną zaletą WDM jest możliwość uzyskania transmisji całkowicie optycznej (ang. all-optical), tj. bez konwersji optyczno-elektrycznej (jak również elektryczno-optycznej) sygnału w węzłach tranzytowych. Kolejną zaletą WDM jest wspomniane zwielokrotnienie z podziałem długości fal umożliwiające w danym łączu jednoczesną transmisję na różnych kanałach (związanych z różnymi długościami fal), co istotnie zwiększa sumaryczną przepustowość łączy. Dalszy rozwój technologii doprowadził m.in. do opracowania techniki DWDM (ang. Dense Wavelength Division Multiplexing) [6], [7] umożliwiającej wykorzystanie 80 (lub więcej) kanałów (każdy o przepustowości np. 40 Gb/s) na jednym łączu optycznym. W niniejszej pracy technologia DWDM stanowi rozwiązanie podstawowe dla większości badań.

Ewolucji architektur sieci teleinformatycznych towarzyszy istotny wzrost udziału żądań o wysokich parametrach jakości usług (ang. Quality of Service – QoS) zdefiniowanych w kontekście przepustowości (ang. bandwidth), opóźnień transmisji (ang. delay), zmienności opóźnień (ang. jitter), oraz strat pakietów (ang. packet losses) [8], [9], często ponadto wrażliwych na przerwanie ciągłości transmisji. Strumieniowe aplikacje multimedialne zazwyczaj wymagają gwarantowanej przepustowości oraz ciągłości transmisji (niezbędnych dla stabilności ich pracy). Przerwanie transmisji na skutek awarii elementu sieci może być krytyczne np. dla strumieniowego ruchu Voice-over-IP (VoIP) czy wideokonferencji. W przypadku aplikacji finansowych przerwa w ciągłości dostępu do informacji (np. dotyczącej aktualnej sytuacji na rynkach kapitałowych) może spowodować ogromne straty finansowe. Przerwanie transmisji podczas operacji ratunkowych oraz wojskowych może doprowadzić nawet do zagrożenia zdrowia czy życia ludzi.

Niedostępność kluczowych węzłów sieci teleinformatycznej mającej zastosowanie przemysłowe może stać się powodem co najmniej nieprawidłowego działania a nawet katastrofy infrastruktury lokalnej/globalnej (produkcja niebezpiecznych elementów, elektrownie atomowe). Zgodne z [10], [11], w przypadku dużej części aplikacji wymagania dotyczące ciągłości transmisji (wpływające na zmienność opóźnienia transmisji) są bardziej rygorystyczne niż te związane z samym opóźnieniem transmisji od krańca do krańca (ang. end-to-end). Oznacza to, że dla metod odtwarzania transmisji po awarii, szybkość przywrócenia transmisji (wpływająca na redukcję zmienności opóźnień) ma przewagę nad doбором tras optymalnych pod względem opóźnienia transmisji.

Awarie elementów sieci powodują ich wykluczenie z udziału w transmisji danych do czasu fizycznego naprawienia awarii. W sytuacji gdy nie ma możliwości realizacji transmisji trasami alternatywnymi, awaria prowadzi nawet do przerwania transmisji aż do czasu naprawienia uszkodzonego elementu.

W przypadku możliwości wykorzystania alternatywnych tras, transmisja jest z kolei wstrzymana do momentu aktywacji odpowiednich tras zabezpieczających wykorzystujących elementy sieci, które nie uległy awarii. W tym przypadku często wykorzystane trasy zainstalowane zawczasu (ang. preplanned protection), tj. przed wystąpieniem awarii elementów sieci [12].

Sieci rozległe (projektowane obecnie w większości przy wykorzystaniu technologii optycznej), z uwagi na duży obszar pokrycia są podatne na awarie swoich elementów (statystycznie głównie łączy, ale również i węzłów [13]). Powodem tych awarii mogą być klęski żywiołowe (takie jak pożary lasów lub trzęsienia ziemi) [14], działania ludzkie zarówno nieumyślne (uszkodzenie łączy optycznych podczas prac budowniczych), jak i umyślne (ataki hackerskie lub fizyczne na elementy sieci) [15]. Zgodne z [16], [17], w optycznych sieciach transportowych średnio raz na dwa dni zdarzają się awarie, które wpływają na transmisję 30 tys. użytkowników końcowych. W pracy [16] został opisany przykład awarii łącza sieci teleinformatycznej obsługującego transmisję 12 mln strumieni użytkowników końcowych.

W celu ograniczenia wpływu awarii elementów sieci na ciągłość transmisji w sieciach teleinformatycznych, zaproponowano dotychczas w literaturze różnorodne modele i metody ochrony transmisji, których celem jest umożliwienie szybkiego przywrócenia transmisji poprzez wykorzystanie w tym celu nieuszkodzonych elementów sieci. Zgodne z [18], [19] zdolność sieci do przywrócenia ciągłości transmisji w czasie nie dłuższym niż odgórnie określony w obliczu wystąpienia awarii jej elementów, określona mianem *przeżywalności* (ang. *survivability*) jest jednym z najważniejszych aspektów zapewnienia niezawodności sieci teleinformatycznych (ang. *communication networks resilience*).

Ochrona transmisji przed awarią elementów sieci teleinformatycznej jest możliwa poprzez wykorzystanie *tras zapasowych* (ang. *backup paths*), tj. alternatywnych tras transmisji. Aby zapewnić należytą ochronę przed awarią łączy/węzłów, trasy zapasowe chroniące przed awarią łącza/węzła nie powinny mieć wspólnych łączy/węzłów tranzytowych z ochranianym fragmentem trasy podstawowej. W celu wyznaczenia tras podstawowych i zapasowych, stosowane są algorytmy znane z teorii grafów [20]. Algorytmy te operują na grafach, które są bezpośrednim odwzorowaniem topologii sieci oraz jej parametrów takich jak np. długości łączy.

W sieciach rozległych najczęściej spotykane są awarie pojedynczych łączy (z uwagi na ryzyko wystąpienia awarii proporcjonalne do ich długości). Jednakże, w obliczu coraz częstszych przypadków awarii wielokrotnych (tj. jednoczesnych awarii k elementów sieci), ochrona transmisji typowo wiąże się z koniecznością ustanowienia k tras zabezpieczających (wykorzystywanych głównie w krótkim

okresie pomiędzy wystąpieniem awarii a fizyczną naprawą uszkodzonych elementów sieci).

Warto zauważyć, że znaczna część scenariuszy klasyfikowanych jako awarie wielokrotne dotyczy w rzeczywistości sekwencji awarii pojedynczych, które wystąpiły w krótkich odstępach czasu zanim możliwa była fizyczna naprawa dotychczas uszkodzonych elementów. Zastosowanie algorytmów doboru tras, które byłyby w stanie odpowiednio szybko wyznaczyć pary tras: podstawową i zabezpieczającą dla każdego żądania zaraz po awarii danego elementu sieci mogłoby więc sprowadzić zadanie ochrony przed awarią wielokrotną do problemu ochrony przed kolejnymi awariami pojedynczymi. To z kolei implikowałoby wyraźne zmniejszenie zapotrzebowania na przepustowości łączy w celu instalacji tras zabezpieczających (gdyż, w celu ochrony danej trasy podstawowej przed sekwencją awarii elementów sieci, wystarczyłyby pojedyncze trasy zabezpieczające zamiast k tras zabezpieczających). W literaturze brakuje jednak rozwiązań z zakresu doboru tras niezawodnych jednocześnie ukierunkowanych na redukcję czasu obliczeń tras.

W innych dziedzinach nauki są znane i często stosowane podejścia transformacyjne wykorzystujące przekształcenia pewnej struktury pierwotnej do innej struktury, której wykorzystanie ma przewagę nad strukturą pierwotną. Przykładem przekształcenia jest np. transformacja Fouriera [21], mająca szereg różnorodnych zastosowań od przetwarzania sygnałów [22] do kryptoanalizy [23] i kryptografii [24]. W przetwarzaniu sygnałów taka transformacja pozwala na reprezentację sygnału oryginalnie zdefiniowanego w dziedzinie czasu w postaci dwóch odrębnych struktur: widma amplitudowego i widma fazowego. Reprezentacja ta pozwala z kolei na rozwiązanie problemów, które są trudne do obliczenia wprost dla pierwotnej reprezentacji. Przykładowo, trudne obliczeniowo jest obliczenie obrazu analogowego sygnału na wyjściu urządzenia elektronicznego mając informację o parametrach sygnału wejściowego. Transformacja Fouriera pozwala na reprezentację wejściowego sygnału jako pary (widmo amplitudowe, widmo fazowe) oraz obliczenie analogicznych charakterystyk widmowych dla urządzenia elektronicznego. Pozwala to z kolei na obliczenie widma amplitudowego oraz widma fazowego sygnału wyjściowego. Odwrócenie widma amplitudowego i widma fazowego pozwala więc uzyskać obraz wyjściowego sygnału. Dyskretna transformacja kosinusowa [25] jest stosowana w kompresji danych w takich

standardach jak: MPEG i JPEG. Standardy kompresji obrazów JPEG-2000 i ICER opierają się na dyskretnej transformacji falkowej [26], [27]. Z kolei, transformacja chirplet [28] znalazła swoje przeznaczenie w radiolokacji.

Metody transformacyjne nie były dotychczas popularne w obszarze algorytmów doboru tras. Wszystkie najważniejsze algorytmy doboru tras dostępne w literaturze działają przy wykorzystaniu struktury grafowej będącej bezpośrednią reprezentacją parametrów i topologii sieci oraz rozpatrują trasy jako ciągi węzłów sieci. Z tego powodu ich skalowalność jest często ograniczona.

Celem pracy jest przedstawienie propozycji autorskich metod zapewniania niezawodności transmisji wieloskokowej (od krańca do krańca) przy wykorzystaniu proponowanych technik transformacji ukierunkowanych na ograniczenie czasu wyznaczania tras, jak i na umożliwienie obsługi przez sieć większej liczby żądań dzięki redukcji zapotrzebowania na zasoby sieci w scenariuszach ochrony przed awarią wielokrotną oraz opracowaniu mechanizmów doboru tras ukierunkowanych na redukcję współczynnika blokowania żądań.

Weryfikacja charakterystyk metod proponowanych w niniejszej rozprawie, ma na celu wykazanie tezy rozprawy, która jest zdefiniowana następująco:

Zastosowanie technik transformacji grafu topologii sieci umożliwi redukcję czasu niezbędnego do wyznaczenia tras niezawodnych od krańca do krańca.

Jak zostało zauważone wcześniej w niniejszym rozdziale, w przypadku wykazania możliwości ograniczenia czasu niezbędnego do wyznaczenia tras, ochrona transmisji przed awarią wielokrotną w znacznej mierze mogłaby być uzyskana poprzez techniki ochrony przed sekwencją awarii pojedynczych. To z kolei istotnie ograniczałoby ilość zasobów niezbędnych do instalacji tras zabezpieczających i dawałoby możliwość obsługi większej liczby żądań, redukując tym samym stopień blokowania żądań. Ograniczenie czasu wyznaczania tras transmisji może więc także skutkować obniżeniem stopnia blokowania żądań.

Niniejsza rozprawa przedstawia propozycje autorskich metod doboru tras w dwóch obszarach zastosowań:

- 1) wykorzystania metod transformacji do opracowania autorskich metod doboru tras ukierunkowanych na zrównoważenie obciążenia

sieci poprzez zastosowanie w metryce kosztu łączy czynnika reprezentującego prognozowane konflikty o zasoby łączy. Prognozowanie konfliktów jest zrealizowane poprzez wykorzystanie dodatkowych struktur reprezentujących w postaci probabilistycznej przepustowość wymaganą przez dotychczas nieobsłużone żądania;

- 2) wykorzystania metod transformacji do opracowania autorskich metod wyznaczania tras transmisji zapewniających ochronę przed awarią węzłów/łączy sieci ze specjalnym naciskiem na redukcję czasu obliczeń. Redukcja czasu jest osiągnięta poprzez wykorzystanie autorskiej struktury (mapy komórkowej) zamiast grafu topologii sieci. Mapa komórkowa jest grafem, którego węzły reprezentują cykle w sieci, a krawędzie reprezentują łączy sieci wspólne dla dwóch cykli. Wykrycie pary rozłącznych tras odbywa się poprzez podział najkrótszego cyklu zawierającego źródłowy i docelowy wierzchołek. Cykl, z kolei, jest wykrywany poprzez połączenia węzłów mapy komórkowej.

Wyniki badań wskazują na przewagę takich metod nad algorytmami referencyjnymi, co ukazuje przydatność tychże metod w tej dziedzinie. W szczególności, rozdziały 3-4 niniejszej rozprawy pokazują, że wykorzystanie metod transformacji pozwala na budowanie algorytmów doboru tras zdolnych do redukcji zarówno współczynnika blokowania żądań w sieci, jak i czasu niezbędnego do wyznaczenia par rozłącznych tras w celu zapewnienia niezawodnej transmisji. Dotychczasowy brak ukierunkowania prac naukowców w tym zakresie, w szczególności w kontekście wykorzystania technik transformacji, sprawiał, że rozwój metod projektowania algorytmów routingu niezawodnego był istotnie hamowany.

Dalsza część rozprawy obejmuje cztery rozdziały numerowane. W szczególności, celem rozdziału 2 jest opis dostępnych technologii i metod ochrony transmisji w optycznych sieciach transportowych oraz uzasadnienie wyboru strategii ochrony transmisji przyjętej w dalszej części rozprawy. W rozdziale 3 są zaprezentowane dwa podejścia ukierunkowane na redukcję prawdopodobieństwa blokowania żądań w sieci szkieletowej poprzez dążenie do zrównoważenia obciążenia łączy sieci. Redukcja prawdopodobieństwa blokowania żądań jest ważna

w doborze tras niezawodnych, ponieważ instalacja zapasowych tras wymaga dodatkowych zasobów przepustowości sieci. W rozdziale 4 są ukazane nowe metody wykrycia pary węzłowo-rozłącznych tras w sieci, których celem jest redukcja czasu obliczeń tras. Rozdział 5 podsumowuje rozprawę uwypuklając m.in. główne wnioski oraz wskazując potencjalne obszary dalszej pracy badawczej.

Wyniki prac badawczych, w oparciu o które została przygotowana niniejsza rozprawa, zostały częściowo opublikowane poniższych pracach (1)-(5), tj. w czterech artykułach w czasopismach (w tym jednym w czasopiśmie *Networks* z listy JCR), oraz w materiałach jednego workshopu zagranicznego:

- (1) Mysliński K., Rak J.: “A New Quantum-Inspired Approach to Reduce the Blocking Probability of Demands in Resource-constrained Path Computation Scenarios”, *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne*, nr. 8-9, pp. 728-734, 2016.
- (2) Mysliński K., Rak J., Kuszner Ł.: “Network Graph Transformation Providing Fast Calculation of Paths for Resilient Routing”, in *Proc. 2016 8th International Workshop on Resilient Network Design and Modeling (RNDM)*, pp. 238-244, 2016, 13-15.09.2016, Halmstad, Szwecja.
- (3) Mysliński K.: “Metoda redukcji prawdopodobieństwa blokowania żądań w sieci szkieletowej oparta o mechanizm prognozowania konfliktów w zasobach przepustowości”, *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne*, vol. 8-9, pp. 715-718, 2017.
- (4) Mysliński K., Rak J.: “Metoda szybkiego wyznaczania par węzłowo-rozłącznych tras dla ochrony transmisji unicast”, *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne*, vol. 8-9, pp. 717-723, 2015.
- (5) Mysliński K., Rak J., Kuszner Ł.: “Toward Fast Calculation of Communication Paths for Resilient Routing”, *Networks*, Wiley, vol. 70, no. 4, pp. 308-326, 2017.

Prezentacje zawartości dwóch prac oznaczonych powyżej jako pozycje (1) oraz (4) zostały ponadto nagrodzone pierwszą i trzecią nagrodą dla młodych naukowców kolejno podczas konferencji KSTiT 2016 oraz KSTiT 2015.

Wyniki badań naukowych, które nie zostały dotychczas opublikowane są zawarte głównie w dziesięciu załącznikach prezentujących rozszerzenie wybranych aspektów, m.in. w zakresie dowodów poprawności proponowanych algorytmów, analizy złożoności obliczeniowej metod, prezentacji szczegółowych wyników symulacyjnych, jak i rozszerzeń autorskich metod w wybranych kontekstach.





2 Dostępne metody i tryby ochrony transmisji przed awariami w optycznych sieciach transportowych

Zgodnie m.in. z pracami [14], [19], [29], [30], istniejące metody ochrony transmisji przed awarią elementów sieci mogą być sklasyfikowane pod względem:

- chwili wyznaczania tras alternatywnych,
- zasięgu ochrony tras zabezpieczających,
- liczby elementów sieci, które mogą jednocześnie ulec awarii,
- rodzaju elementów sieci, które mogą ulec awarii,
- trybu ochrony transmisji,
- sposobu rezerwacji zasobów sieci na potrzeby instalacji tras zabezpieczających.

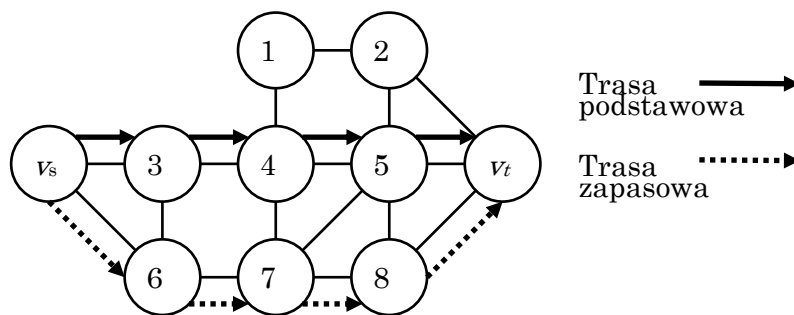
W szczególności, pod względem chwili wyznaczania tras alternatywnych metody ochrony transmisji dzielą się na:

- proaktywne (ang. proactive), w przypadku których instalacja tras alternatywnych ma miejsce wraz z instalacją tras podstawowych;
- reaktywne (ang. reactive) rozpoczynające procedurę określenia tras zapasowych dopiero po wystąpieniu awarii elementu sieci.

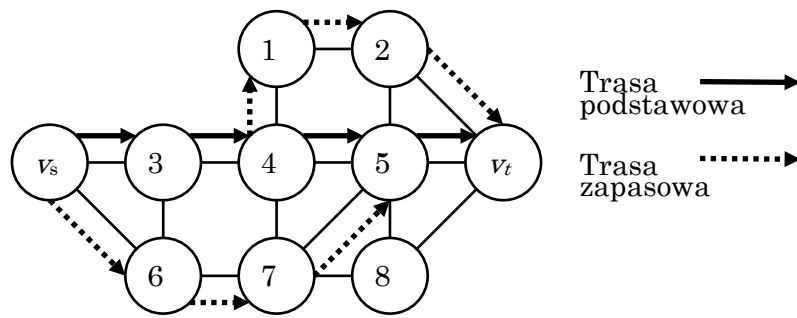
Mechanizmy proaktywne cechują się generalnie krótszą wartością czasu przełączenia transmisji na trasę zabezpieczającą. Odbywa się to jednakże kosztem zwiększonego zapotrzebowania na zasoby sieci niezbędnych do instalacji zawczasu tras zabezpieczających (używanych ponadto typowo jedynie w stosunkowo krótkim okresie pomiędzy wystąpieniem awarii a naprawą fizyczną uszkodzonego elementu). Metody reaktywne są z kolei znacznie bardziej efektywne pod względem zapotrzebowania na zasoby sieci, lecz nie gwarantują 100% ochrony po wystąpieniu awarii (trasy zabezpieczające mogą nie zostać wyznaczone po awarii z uwagi na chwilowy brak stosownych zasobów sieci w danym okresie).

Biorąc pod uwagę możliwe warianty zasięgu ochrony trasy podstawowej metody ochrony transmisji dzielimy na:

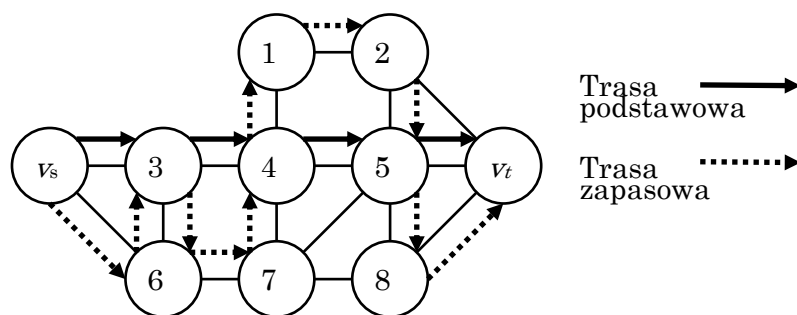
- ochronę trasy (ang. path protection), gdzie pojedyncza trasa zapasowa chroni całą trasę podstawową. W przykładzie zaprezentowanym na Rys. 2.1, trasa podstawowa $P_o=(v_s, 3, 4, 5, v_t)$ jest chroniona poprzez pojedynczą trasę zapasową $P'=(v_s, 6, 7, 8, v_t)$.
- ochronę obszarową (ang. segment protection), w przypadku której trasa podstawowa jest ochraniajana przez kilka tras alternatywnych, z których każda zapewnia ochronę pewnego fragmentu trasy podstawowej. Przykład ochrony obszarowego jest zaprezentowany na Rys. 2.2. Fragment trasy $P_{o1}=(v_s, 3, 4, 5)$ jest chroniony poprzez trasę zapasową $P_{1'}=(v_s, 6, 7, 5)$. Z kolei, część trasy $P_{o2}=(4, 5, v_t)$ jest chroniona poprzez trasę zapasową $P_{2'}=(4, 1, 2, v_t)$.
- ochronę lokalną. W tym przypadku trasy zapasowe chronią pojedyncze łącza trasy podstawowej (w przypadku ochrony przed awarią pojedynczego łącza) lub dwa sąsiednie łącza trasy podstawowej (dla wariantu ochrony przed awarią pojedynczego węzła). W przykładzie zaprezentowanym na Rys. 2.3 dotyczącym ochrony przed awarią pojedynczego łącza, łącza sieciowe (elementy trasy podstawowej) $e(v_s, 3)$, $e(3, 4)$, $e(4, 5)$, $e(5, v_t)$ są chronione (odpowiednio) poprzez trasy zapasowe $P_1'=(v_s, 6, 3)$, $P_2'=(3, 6, 7, 4)$, $P_3'=(4, 1, 2, 5)$, $P_4'=(5, 8, v_t)$.



Rys. 2.1 Przykład ochrony trasy podstawowej poprzez trasę zapasową



Rys. 2.2 Przykład ochrony obszarowej



Rys. 2.3 Przykład ochrony lokalnej (wariant dla ochrony przed awarią łącza)

Generalnie im mniejszy jest zasięg ochrony tras zabezpieczających, tym więcej zasobów sieci jest niezbędnych w celu instalacji tras alternatywnych (przykładowo, sumaryczna liczba łączy wszystkich tras zabezpieczających daną trasę podstawową w przypadku ochrony o zasięgu łącza (na Rys. 2.3 jest to dziesięć łączy) jest znacznie większa niż sumaryczna liczba łączy w przypadku pojedynczej trasy zabezpieczającej daną trasę podstawową dla wariantu ochrony o zasięgu ścieżki (na Rys. 2.1 są to cztery łącza).

Pod względem liczby elementów sieci, które mogą jednocześnie ulec awarii metody ochrony transmisji dzielą się na techniki zapewniające:

- ochronę przed awariami pojedynczymi (poprzez wyznaczenie jednej trasy zapasowej dla ochrony danej trasy/fragmentu trasy),
- ochronę przed jednoczesną awarią wielu (k) elementów sieci poprzez instalację k tras zapasowych dla ochrony danej trasy bądź każdego jej fragmentu.

Analizując rodzaje elementów sieci, które mogą ulec awarii, wyróżniamy metody:

- ochronę przed awarią łącza,
- ochronę przed awarią węzła.

Pod względem trybu ochrony transmisji, metody (stosowane wyłącznie z wariantem proaktywnym instalacji tras) dzielą się na:

- tryb ochrony ścieżki 1:1. W tym trybie trasy zapasowe wyznaczone zawczasu, lecz wykorzystywane wyłącznie po wystąpieniu awarii
- tryb ochrony ścieżki 1+1. W tym trybie wyznaczone zawczasu trasy zapasowe są wykorzystane na potrzeby transmisji równoległe z trasami podstawowymi, a po wystąpieniu awarii transmisja jest kontynuowana przy wykorzystaniu trasy/tras, które nie uległy awarii.

Biorąc pod uwagę możliwość współdzielenia zasobów (przepustowości łącza) tras zabezpieczających, metody (stosowane z założenia w przypadku ochrony proaktywnej) dzielą się na:

- metody ochrony dedykowanej, tj. bez współdzielenia zasobów (przepustowości łącza rezerwowane na potrzeby danej trasy zabezpieczającej nie są współdzielone z żadną inną trasą zabezpieczającą);
- metody ze współdzieleniem zasobów (trasy zapasowe chroniące rozłączne trasy podstawowe współdzielą zasoby pomiędzy sobą w celu podniesienia efektywności wykorzystania zasobów sieci. Trasy zapasowe są ustanawiane biorąc pod uwagę kryterium optymalności wspólnego wykorzystania zasobów. Trasy zabezpieczające P'_1 i P'_2 mogą wykorzystywać wspólne kanały transmisji na wspólnych łączach, jeśli ochraniające trasy podstawowe P_1 i P_2 nie zawierają wspólnych elementów tranzytowych (a więc kiedy nie ma ryzyka konieczności jednoczesnego aktywowania obu tras zabezpieczających na skutek jednoczesnej awarii obu tras podstawowych). Metody współdzielenia zasobów tras zabezpieczających pozwalają na redukcję poziomu zapotrzebowania na zasoby sieci w związku z

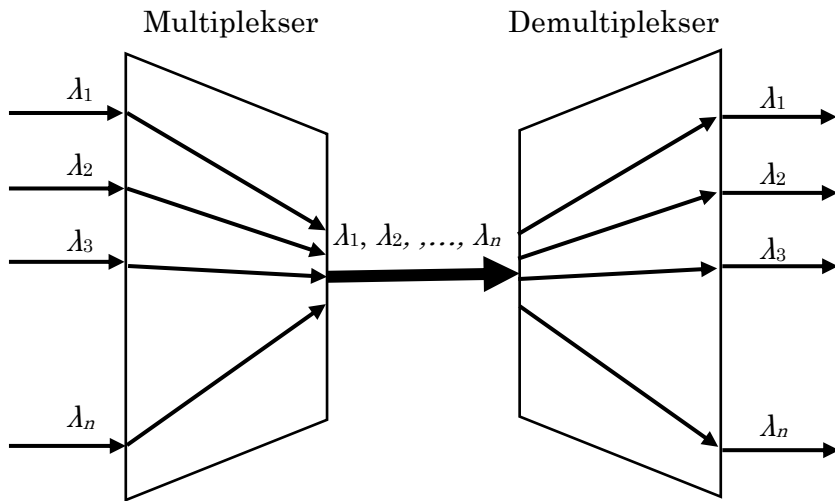
instalacją tras zabezpieczających. Jednakże może się to odbywać kosztem niekorzystnego wzrostu wartości innych parametrów wykrytych tras zabezpieczających (na przykład ich długości, co koreluje dodatnio ze wzrostem czasu niezbędnego do aktywacji trasy alternatywnej) [31].

Z uwagi na mnogość możliwych rozwiązań ukazanych w powyższej klasyfikacji, w podrozdziale 2.2 zawarty jest opis wraz z uzasadnieniem strategii ochrony przyjętej w niniejszej rozprawie.

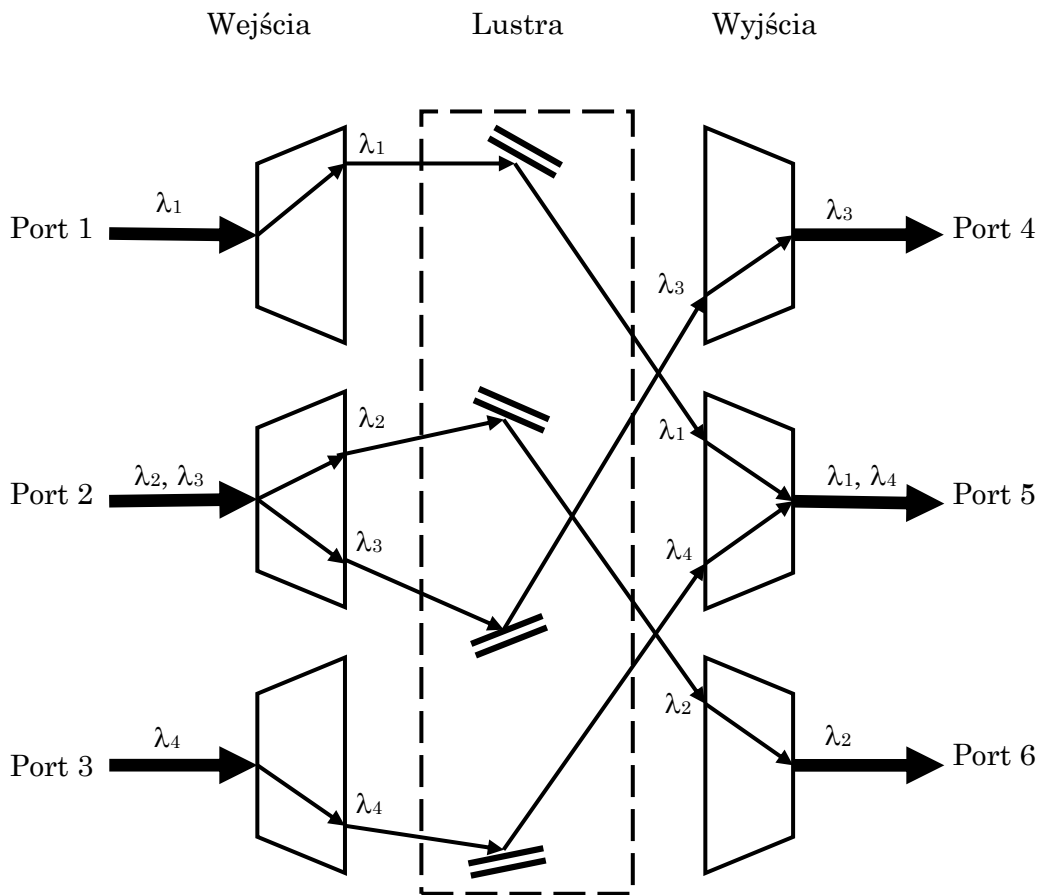
2.1 Zastosowanie metod ochrony transmisji w optycznych sieciach transportowych

Sieci SDH są standardem wykorzystującym zwielokrotnienie w dziedzinie czasu TDM (ang. Time Division Multiplexing) [32], które pozwala na obsługę kilku abonentów wykorzystując jeden kanał fizyczny. Osiągnięto to poprzez nadanie dla strumieni transmisji krótkich rozłącznych przedziałów czasu transmisji. Rozwiązanie SDH generalnie zostało opracowane dla transmisji sygnałów elektrycznych. Z kolei specyfikacja WDM została zaprojektowana dla sieci optycznych i pozwala pojedynczemu łączu na obsługę wielu strumieni transmisji poprzez przydzielenie im różnych długości fal. Wykorzystanie optycznych multiplexerów/demultiplexerów (Rys. 2.4) oraz przełączników optycznych pozwala na wyeliminowanie konwersji sygnału z postaci optycznej na elektryczną i odwrotnie w węzłach tranzytowych, co z kolei redukuje opóźnienie dostarczenia pakietów do węzłów docelowych.

Możliwość wyeliminowania konwersji sygnału pomiędzy domenami optyczną i elektryczną doprowadziła do rozwoju koncepcji AON (ang. All-Optical Networks) [33] dotyczącej wieloskokowej transmisji pakietów całkowicie w dziedzinie optycznej. Głównym elementem sieci AON są przełączniki optyczne, które pozwalają na przekierowanie transmisji wykorzystującej daną długość fali z dowolnego portu na inny dowolny port przełącznika (Rys. 2.5).



Rys. 2.4 Schemat agregacji i rozdziału fal w przypadku transmisji w sieci WDM



Rys. 2.5 Schemat przełącznika optycznego 3x3 wraz z przykładowym przełączeniem ruchu na wybranych długościach fal

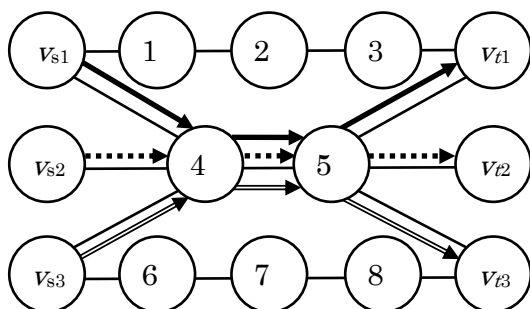
Warto zwrócić uwagę na to, że w kontekście powyższej architektury, problem doboru tras ulega rozszerzeniu do problemu RWA (ang. Routing and Wavelength Assignment problem) – tj. doboru tras i przydziału długości fali [34], [35]. W szczególności oznacza to, że w przypadku transmisji całkowicie optycznej, gdy węzły nie są wyposażone w konwertery długości fal, nie jest możliwa instalacja tras pomiędzy węzłem źródłowym v_{s1} a docelowym v_{t1} i pomiędzy węzłami v_{s2} a v_{t2} , jeśli trasy te mają wykorzystywać taką samą długość fali $\lambda(v_{s1}, v_{t1}) = \lambda(v_{s2}, v_{t2})$, a ich trasy zawierają wspólne łącza

W celu zapewnienia efektywnego doboru tras w sieciach AON, niezbędne jest więc rozwiązanie nie tylko problemu doboru tras, ale również zastosowanie w rozwiązaniu końcowym także poprawnego z punktu widzenia współistnienia w sieci wielu tras przydziału długości fali dla każdej trasy, który nie powoduje konfliktów między trasami (rozłącznego pod względem długości fal). Z uwagi na NP -zupełność problemu RWA [35], nie jest możliwe zagwarantowanie uzyskania rozwiązania optymalnego w sensie wartości funkcji kryterialnej dla topologii sieci rzeczywistych w rozsądnym (tj. wielomianowym) czasie. Problem ten może być ograniczony poprzez zwiększenie liczby dostępnych długości fal na łączach (np. jak ma to miejsce w przypadku technologii DWDM – ang. Dense Wavelength Division Multiplexing), ponieważ większa liczba dostępnych długości fal redukuje ryzyko wystąpienia wysycenia sieci (ang. network congestion), a proste metody heurystyczne umożliwiają uzyskanie rozwiązania akceptowalnego. Jednak, warto pamiętać, że głównym celem architektury DWDM jest zwiększenie informacyjnej przepustowości łącza optycznego, a nie ograniczenie problemu RWA.

Mimo tego, że sieci transportowe AON są sieciami o dużej przepustowości, w odniesieniu do problemu doboru tras one są traktowane jako sieci o ograniczonych zasobach. Powodem tego jest fakt, że w odróżnieniu od sieci z komutacją pakietów, w sieciach z komutacją kanałów (jakimi są sieci transportowe) instalacja trasy rezerwuje całe kanały lub jego część na łączach sieci na potrzeby transmisji. Przepustowość kanałów zarezerwowanych na potrzeby instalacji podstawowych tras transmisji nie może być współdzielona z innymi trasami podstawowymi, niezależnie od tego na ile efektywnie będą wykorzystane dane kanały. Wyczerpanie liczby dostępnych kanałów na łączach niezbędnych dla instalacji kolejnych tras może prowadzić do blokowania kolejnych żądań, zwłaszcza jeśli decyzje o doborze tras są podejmowane lokalnie.

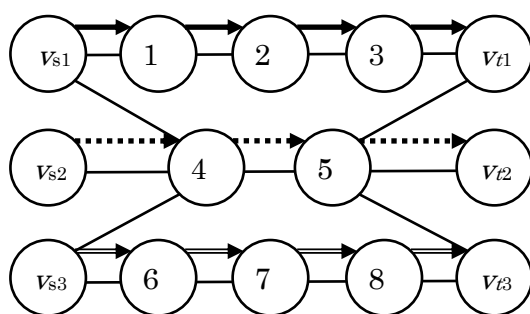
Powyższy problem może być ograniczony poprzez zastosowanie koncepcji tzw. routingu scentralizowanego (powszechnego w obecnych sieciach transportowych), w której centrum zarządzania siecią odbiera wszystkie żądania instalacji tras w sieci i które wyznacza wszystkie trasy. Rozwiązanie to pozwala na rozwiązanie problemu doboru tras dla całego zbioru żądań jednocześnie, dając tym samym znacznie lepsze rozwiązanie pod względem wartości funkcji kryterialnej niż analogiczne rozwiązanie problemu doboru tras dla każdego żądania oddzielnie.

Przykład rozwiązania problemu doboru tras dla każdego żądania oddzielnie jest ukazany na Rys. 2.6. Każda z wykrytych tras jest optymalna w sensie kosztu dla pojedynczego żądania i zawiera trzy łącza. Jednak sytuacja wygląda inaczej jeżeli założyć, że ta sama sieć jest siecią o ograniczonych zasobach, w której każde łącze ma przepustowość równą 1, a każde żądanie wymaga przepustowości równej 0.5 dla transmisji danych. W takim przypadku przepustowość łącza (4, 5) nie wystarczy dla transmisji danych dla wszystkich trzech żądań, co z kolei prowadzi do blokowania żądań i niemożliwości dostarczenia danych do węzłów docelowych. Zadanie określenia doboru tras optymalnego pod względem redukcji współczynnika blokowania żądań jest problemem *NP*-zupełnym [36]. Jednak mając kompleksową informację o charakterze planowanego ruchu w całej sieci, jest większa szansa znalezienia przez centrum zarządzania akceptowalnego rozkładu tras dla wszystkich żądań.



Rys. 2.6 Przykład zdecentralizowanego doboru tras w sieci o nieograniczonych zasobach

Na Rys. 2.7 zaprezentowany jest rozkład tras, który zapewnia transmisję pomiędzy węzłami źródłowym a docelowym dla każdego z analizowanych żądań, co zostało osiągnięte dzięki suboptymalnym trasom dla dwóch żądań.



Rys. 2.7 Przykład scentralizowanego doboru tras w sieci o ograniczonych zasobach

Scentralizowany dobór tras również pozwala na efektywne rozwiązanie problemu RWA w rozsądnym czasie za pomocą algorytmów heurystycznych. Dalszy rozwój sieci optycznych doprowadził do powstania elastycznych sieci optycznych EONs (ang. elastic optical networks) [37], [38]. W takich sieciach kanał transmisji może być podzielony na szereg podkanałów, każdy z których może być wykorzystany osobno lub razem z innymi podkanałami dla transmisji danych w ramach danego połączenia.

Dla podziału kanału na podkanały wykorzystana jest modulacja OFDM (ang. Orthogonal Frequency-Division Multiplexing) [39]. Pozwala to na elastyczne zarządzanie rezerwowaną przepustowością łączy sieci i przez to podnosi efektywność wykorzystania przepustowości sieci. Wyodrębnienie podkanałów faktycznie podnosi liczbę możliwych wariantów instalacji tras i ogranicza w pewnym stopniu zjawisko blokowania żądań w wyniku ograniczeń związanych z zadaniem RWA.

Cały czas trwają prace w kierunku zastosowania w praktyce całkowicie optycznej transformacji długości fali [40], [41] bez optyczno-elektrycznych i elektryczno-optycznych transformacji na węzłach sieci. Dostępność całego szeregu rozwiązań mówi o tym, że razem z wszechobecnym wdrożeniem AON będzie możliwe wdrożenie i metod całkowicie-optycznej transformacji, które pozwolą z kolei uniknąć problemu przydziału długości fali.

2.2 Wybór sposobu ochrony transmisji i modelu routingu

W niniejszej sekcji zostały przedstawione założenia konstrukcyjne dotyczące wszystkich rozwiązań przedstawionych w rozprawie. Jednym z głównych założeń jest próba zwrócenia uwagi na możliwość zastosowania metod transformacyjnych w celu rozwiązania problemu doboru tras – podejścia potencjalnie wartościowego zarówno w obszarze badań naukowych, jak i zastosowań komercyjnych np. przez producentów sprzętu sieciowego. Z uwagi na szybki postęp technologiczny, celem jest także zaproponowanie rozwiązań mających zastosowanie nie tylko dla architektur obecnych sieci, ale także w kontekście ich ewolucji.

Jak zostało opisane wcześniej, obecne sieci transportowe wykorzystują głównie scentralizowane mechanizmy doboru tras. Z roku na rok rośnie znaczenie koncepcji sieci programowalnych SDN (ang. Software-Defined Networking) [42]-[46] w sieciach z komutacją pakietów, będącej również przykładem architektury wykorzystującej zarządzanie scentralizowane. Scentralizowany dobór tras pozwala na efektywne ograniczenie problemu obciążenia sieci. Umożliwia także wykorzystanie zaproponowanych w niniejszej pracy algorytmów nie tylko w przypadku klasycznych architektur sieci transportowych (rozległych), które są sieciami komutacji kanałów, ale także w przypadku różnorodnych architektur wykorzystujących scentralizowane zarządzanie, w szczególności bazujących na koncepcji SDN.

Transmisja informacji w sieciach z komutacją kanałów nie polega na dostarczeniu osobnych pakietów od węzła docelowego do węzła źródłowego poprzez węzły transportowe, lecz jest transmisją ciągłego strumienia danych o ustalonej przepustowości. W literaturze jest dostępnych kilka odmian całkowicie optycznej transmisji pakietów w sieciach optycznych, jak np. SWANET (ang. Self-routed Wavelength-Addressable Network) [47], które nie są jednak stosowane przez operatorów. Ważnym warunkiem efektywnej komutacji pakietów w sieci jest bowiem możliwość umieszczenia transmitowanego pakietu w buforze w węzłach tranzytowych z możliwością dostępu do tego pakietu w dowolnej chwili. Głównym problemem wdrożenia optycznej komutacji pakietów w sieciach transportowych jest brak przydatnych do tego celu buforów optycznych.

Warto zwrócić uwagę, że w sieciach lokalnych rośnie rola zarządzania przepływami dla zapewnienia wymagań QoS. W takich przypadkach transmisja

danych traktowana jest nie jako komutacja osobnych pakietów, lecz komutacja ciągłego strumienia pakietów o zarezerwowanej przepustowości (co przybliża funkcjonowanie takich sieci do sieci z komutacją kanałów). Algorytmy doboru tras zaprojektowane dla sieci z komutacją kanałów mogą być więc (po modyfikacjach) zastosowane w sieciach z komutacją pakietów z zarządzaniem przepływami. W związku z powszechnym stosowaniem w praktyce metod komutacji kanałów, jak i widocznego rozwoju technologii zarządzania przepływami w sieciach lokalnych, w niniejszej pracy przyjęto model **komutacji kanałów**.

Kolejne założenie w niniejszej rozprawie dotyczy kwestii rozpatrywania problemu doboru długości fal wraz z problemem doboru tras (łącznie problemu RWA). Innymi słowy, istotne jest rozstrzygnięcie czy projektowane algorytmy doboru tras powinny także określać konfigurację dotyczącą przydziału długości fal. Generalnie, problem RWA w przyszłości nie straci istotnie na ważności w przypadku przewagi wdrożeń koncepcji AON nad wdrożeniem sprzętu sieciowego z optoelektryczną konwersją sygnału. Zdaniem autora niniejszej rozprawy, taki scenariusz jest bardzo prawdopodobny. Jak zostało opisane powyżej, dostępny jest cały szereg rozwiązań transmisji całkowicie w dziedzinie optycznej sygnału, tj. z całkowicie optyczną konwersją długości fal w węzłach tranzytowych. W przypadku wdrożenia w przyszłości architektur AON oferujących pełen zakres konwersji długości fal (tj. z dowolnej długości fali łącza wchodzącego na dowolną długość fali łącza wychodzącego), problem przydziału długości fali zniknie. Z tego punktu widzenia problem **przydziału długości fali nie jest rozpatrywany w dalszej pracy**.

W celu zapewnienia ciągłości transmisji w sieciach optycznych w sytuacji awarii elementów sieci (węzłów/łączy), ważnym parametrem jest czas odtworzenia transmisji po awarii t_{rt} (2.1) [12].

W przypadku ochrony proaktywnej, trasa zabezpieczająca jest wyznaczana zawczasu – w momencie ustanowienia połączenia wraz z trasą podstawową, co redukuje łączny czas przywrócenia transmisji po awarii z uwagi na brak konieczności poszukiwania i instalacji trasy zabezpieczającej po awarii w ramach czasu t_s . Proaktywne wyznaczenie tras gwarantuje także dostępność zasobów sieci (przepustowości łączy) dla trasy zabezpieczającej po awarii elementu sieci. Reaktywne metody ochrony transmisji nie są z kolei w stanie zagwarantować dostępności trasy zabezpieczającej po awarii np. w sytuacji dużego obciążenia sieci. Ten fakt spowodował wybór w niniejszej rozprawie koncepcji ochrony **proaktywnej**.

$$t_{rt} = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7 \quad (2.1)$$

gdzie:

- t_1 – czas wykrycia awarii
- t_2 – czas lokalizacji awarii (czas niezbędny w celu wykrycia awarii elementu, który uległ awarii)
- t_3 – czas oczekiwania przez mechanizmy wykrycia awarii wyższych warstw sieci na ukończenie pracy przez mechanizmy niższych warstw (istotne dla sieci warstwowych)
- t_4 – czas dostarczenia informacji o awarii do węzła sieci odpowiedzialnego za odtwarzanie transmisji po awarii
- t_5 – czas naprawienia transmisji. Najbardziej czasochłonna część, która wymaga zależnie od modelu: wykrycie zabezpieczającej trasy, instalację zabezpieczającej trasy w sieci, przełączenie strumienia danych na trasę zabezpieczającą
- t_6 – czas synchronizacji. Czas niezbędny dla weryfikacji trasy zabezpieczającej i dostarczania transmisji do węzłów docelowych poprzez trasę zabezpieczającą
- t_7 – czas normalizacji, czas niezbędny do przeprowadzenia fizycznej naprawy uszkodzonego elementu sieci i powrotu do ponownego uruchomienia tras transmisji przed wystąpieniem awarii

Zgodne z [13], awarie pojedynczych łączy stanowią około 70% awarii w sieciach transportowych optycznych. Kolejne 20% przypadków dotyczy awarii pojedynczych węzłów, a 10% stanowią wielokrotne awarie elementów sieci. Warto podkreślić, że modele ochrony przed awariami pojedynczych łączy zapewniają ochronę wyłącznie przed awarią łączy. Odmienne, metody, modele ochrony trasy podstawowej przed awarią jej węzłów tranzytowych zapewniają także ochronę przed jednoczesną awarią wszystkich łączy sąsiednich względem ochranianego węzła tranzytowego trasy podstawowej. W taki sposób model ochrony przed awarią pojedynczego węzła jest odpowiedni w ujęciu statystycznym w przypadku 90%

scenariuszy awarii. Ten fakt spowodował wybór w niniejszej rozprawie modelu **ochrony przed awarią pojedynczego węzła sieci**.

Rozpatrując zasięg ochrony przez pojedynczą trasę zabezpieczającą, model ochrony lokalnej (np. ochrona pojedynczego węzła/łącza trasy podstawowej) jest częściej stosowana w przypadku wariantów ochrony reaktywnej oraz w sieciach o zdecentralizowanym doborze tras. Pozwala to zredukować czas przywrócenia transmisji co najmniej o czas dostarczenia informacji o awarii do centrum zarządzania siecią (w ramach czasu t_4). W sieciach o zdecentralizowanym doborze tras, zadanie wyznaczenia i aktywacji trasy zabezpieczającej jest realizowane poprzez węzeł, który bezpośrednio wykrył awarię, co pozwala na redukcję czasu przekierowania transmisji do węzła docelowego poprzez trasę zapasową. W takim przypadku docelowym węzłem trasy zapasowej jest bowiem węzeł następnego przeskoku na trasie podstawowej (dla ochrony przed awarią pojedynczego łącza) lub węzeł następny na trasie podstawowej względem węzła następnego przeskoku (dla ochrony przed awarią pojedynczego węzła).

Jak opisano powyżej, reaktywne metody ochrony transmisji nie gwarantują dostępności przepustowości łączy dla instalacji zapasowej trasy, zwłaszcza w scenariuszach o wysokim obciążeniu sieci. Z tego powodu, do dalszych badań w niniejszej pracy wybrany został wariant **ochrony proaktywnej**.

Należy także pamiętać, że w przypadku wykorzystania metody proaktywnej ochrony lokalnej w celu zapewnienia gwarancji odtwarzania transmisji po awarii, niezbędna jest instalacja zapasowych tras dla każdego tranzytowego elementu trasy podstawowej, co z kolei nie jest efektywne w sieci o ograniczonych zasobach. Proaktywna ochrona obszarowa wymaga instalacji także więcej niż jednej trasy zapasowej dla ochrony kilku obszarów trasy podstawowej i dlatego jest mniej efektywna pod względem zapotrzebowania na zasoby sieci niż metoda ochrony o zasięgu całej trasy. Warto dodać, że ochrona obszarowa jest przydatna w doborze tras ze współdzieleniem zasobów, gdzie pozwala na zauważalną redukcję obciążenia sieci. Jednak jak uzasadniono poniżej, model współdzielenia zasobów nie jest rozpatrywany w dalszej części rozprawy z powodu wyboru trybu ochrony 1+1. Z powodu wspomnianych powyżej ograniczeń wariantów ochrony lokalnej i obszarowej, w niniejszej pracy jest wykorzystywany **model ochrony ścieżki**.

Zgodnie ze statystykami przedstawionymi powyżej, zastosowanie ochrony trasy podstawowej poprzez pojedynczą trasę zapasową chroniącą przed awarią

pojedynczego węzła (tj. trasę alternatywną węzłowo-rozłączną z ochranianą trasą podstawową) stanowi skuteczną ochronę przed 90% wszystkich awarii w sieci i wymaga przy tym zainstalowania wyłącznie pojedynczych tras zabezpieczających. W celu zapewnienia ochrony trasy podstawowej w sytuacji awarii wielokrotnych (stanowiących około 10% wszystkich awarii), ochrona trasy podstawowej często związana jest z koniecznością zainstalowania $k+1$ rozłącznych tras, gdzie k jest założoną odgórnie liczbą możliwych jednoczesnych awarii elementów sieci. Zastosowanie metod ochrony ścieżki przed wielokrotnymi awariami elementów podniosłoby przeżywalność sieci tylko o maksymalnie około 10% przy znacznym obciążeniu sieci (koszt instalacji każdej z k tras zabezpieczających jest większy od kosztu instalacji trasy podstawowej, gdyż trasy zabezpieczające mają przeważnie więcej przeskoków od trasy podstawowej), powodując podniesienie prawdopodobieństwa blokowania żądań.

Warto zwrócić uwagę na to, że często awarie wielu elementów sieci nie mają miejsca dokładnie w tym samym czasie i są traktowane jako wielokrotne tylko z uwagi na to, że czas przywrócenia transmisji dla danej strategii ochrony transmisji przewyższa czas pomiędzy kolejnymi awariami elementów sieci. Z tego powodu skrócenie czasu odtwarzania uszkodzonych połączeń po awarii poprzez zastosowanie bardziej efektywnych czasowo rozwiązań może sprawić, że dany scenariusz awarii wielokrotnej może być traktowany jako sekwencja awarii pojedynczych.

Spostrzeżenie to jest kluczowe w niniejszej rozprawie w celu wykazania, że możliwa jest redukcja prawdopodobieństwa wystąpienia wielokrotnych awarii w sieci poprzez redukcję czasu instalacji tras w sieci.

Określenie „awaria wielokrotna” stosuje się w literaturze [13] w odniesieniu do scenariusza awarii kilku elementów sieci, w którym awaria danego elementu sieci nie została w pełni obsłużona (np. poprzez aktywację odpowiednich tras zabezpieczających) przed wystąpieniem awarii kolejnego elementu sieci. Przykładowo w sieci rozległej, w której wyznaczane są jedynie trasy podstawowe, a aktualizacja tras w celu dopasowania ich do zmieniających się żądań ma miejsce raz w tygodniu, (np. co poniedziałek o godz. 9:00) w przypadku awarii a_1 , a_2 , a_3 charakteryzujących się czasami wystąpień kolejno: $t(a_1)$: wtorek godz. 11:10, $t(a_2)$: środa godz. 16:30, $t(a_3)$: środa godz. 16:32, awarie te w istocie stanowią awarię wielokrotną. Jednak, jeżeli ta sama sieć realizuje dobór tras codziennie, ten sam

scenariusz zawiera dwie awarie: awarię pojedynczego elementu we wtorek o godz. 11:10 i wielokrotną awarię we środę w przedziale czasu 16:30-16:32. Jeżeli sieć jest zdolna przeliczyć i aktualizować trasy co minutę, to w takim scenariuszu występują trzy osobne awarie pojedynczych elementów. W ogólnym przypadku możemy mówić, że dwie awarie pojedynczych elementów sieci a_1 i a_2 są awarią wielokrotną jeżeli czas odtworzenia transmisji t_{rt} po pierwszej z nich spełnia warunek (2.2):

$$t_{rt} > |t(a_2) - t(a_1)| \quad (2.2)$$

Z tego powodu redukcja czasu wyznaczenia i instalacji nowych tras (czas wykrycia trasy zapasowej t_s jest zawarty w t_{rt} (2.1)) ogranicza prawdopodobieństwo występowania awarii wielokrotnych. Po wystąpieniu pierwszej awarii a_1 i aktywowania odpowiednich tras zabezpieczających, kolejna awaria a_2 może prowadzić do dwóch scenariuszy:

- scenariusz 1, w którym awarii uległy tylko trasy podstawowe (zapasowe trasy nie uległy awarii). W takim przypadku użyte zostaną odpowiednie trasy zabezpieczające,
- scenariusz 2, w którym awarii uległy trasy zabezpieczające, które są aktualnie w użyciu z uwagi na ich aktywację w następstwie wcześniejszej awarii a_1 uszkadzającej ochraniające trasy podstawowe. Ten scenariusz jest w istocie fatalny dla ochrony ścieżki przed pojedynczymi awariami: z uwagi na uszkodzenie zarówno tras podstawowych podczas awarii a_1 oraz zapasowych podczas awarii a_2 , nie jest możliwe szybkie przywrócenie transmisji.

Jednak, scenariusz 2 nie wystąpi nigdy, jeżeli w przedziale czasu $|t(a_2) - t(a_1)|$ sieć jest zdolna:

- wykryć awarię a_1 i dostarczyć informację o awarii do centrum zarządzania siecią;
- wyznaczyć nowe pary rozłącznych tras (podstawowej i zabezpieczającej) dla każdego żądania w sieci;
- zainstalować nowe trasy w sieci.

W takim przypadku opisanym zależnością (2.3), w momencie występowania awarii $t(a_2)$ każda aktywna trasa podstawowa ma trasę zapasową i wystąpienie scenariusza 2 nie jest możliwe.

$$t_1 + t_2 + t_3 + t_4 + t_5 + t_6 < |t(a_2) - t(a_1)| \quad (2.3)$$

Dodatkowym celem przyświecającym opracowaniu zaprezentowanych w niniejszej rozprawie mechanizmów szybkiego wyznaczania oraz instalacji tras podstawowych i zabezpieczających, jest sprawienie aby poprzez krótki czas wyznaczania tras i odtwarzania połączeń można było znaczną część awarii wielokrotnych traktować jako sekwencję awarii pojedynczych. To z kolei sprawiłoby, że klasyczny schemat ochrony trasy podstawowej poprzez pojedynczą trasę zabezpieczającą byłby wystarczający do zapewnienia ochrony w przypadku sekwencyjnych awarii pojedynczych, które, z uwagi na krótki odstęp pomiędzy nimi, są obecnie często traktowane jako awarie wielokrotne. Powyższe stwierdzenie dodatkowo uzasadnia przyjęcie w niniejszej rozprawie modelu ochrony przed awarią pojedynczego elementu sieci.

Zastosowanie wyłącznie jednej trasy zabezpieczającej do ochrony trasy podstawowej zmniejsza zapotrzebowanie na zasoby sieci redukując tym samym prawdopodobieństwo blokowania kolejnych żądań i zwiększając pojemność informacyjną sieci przy niezmiennej przepustowości. Pod pojęciem „*pojemności informacyjnej sieci*” rozumiemy w niniejszej pracy sumę przepustowości żądań które sieć jest w stanie obsłużyć. Nie jest to wartość stała dla danej struktury sieci lecz zależy również od przyjętego schematu doboru tras. W scenariuszu zaprezentowanym na Rys. 2.6 pojemność informacyjna sieci jest równa 1, ponieważ w sieci tej mogą być zainstalowane trasy tylko dla dwóch żądań o wymaganej przepustowości 0.5 każda (realizacja jednego żądania jest wstrzymana z powodu braku zasobów sieci). Jednak dla scenariusza ukazanego na rysunku Rys. 2.7 (z wykorzystaniem routingu scentralizowanego oraz ukierunkowanego na zrównoważenie obciążenia sieci) pojemność informacyjna jest równa 1.5, ponieważ możliwa jest instalacja tras dla wszystkich trzech żądań o wymaganej przepustowości 0.5 każda.

Jak zostało ukazane we wcześniejszych częściach niniejszej rozprawy, w przypadku modelu ochrony ścieżki, w literaturze występują jej dwa główne

warianty określane mianem modeli 1+1 i 1:1 [12]. Tryb ochrony 1:1 odnosi się do klasycznego wariantu ochrony ścieżki, w którym trasa zabezpieczająca jest instalowana wraz z trasą podstawową, lecz jest wykorzystywana wyłącznie po wystąpieniu awarii. Odmiennie, w trybie 1+1 trasa zabezpieczająca jest wykorzystywana równoległe z trasą podstawową. W sytuacji braku awarii, węzeł źródłowy transmituje dane równoległe dwiema ścieżkami, a węzeł docelowy decyduje, z której trasy odbierać dane. Po wystąpieniu awarii węzeł docelowy odbiera dane z tej trasy, która nie uległa uszkodzeniu, a więc ciągłość transmisji nie zostaje przerwana nawet na chwilę.

Z uwagi na to, że tryb ochrony 1+1 zapewnia minimalny czas naprawienia transmisji po awarii (równy różnicy opóźnień transmisji informacji wzdłuż trasy zabezpieczającej i podstawowej), w niniejszej rozprawie zakładane jest wykorzystanie **trybu 1+1**. Jako że w przypadku trybu ochrony 1+1, nie jest możliwe współdzielenie przepustowości łączy pomiędzy trasami zapasowymi, w niniejszej pracy stosowany jest **model bez współdzielenia zasobów**.

Podsumowując, w niniejszej pracy jest wybrany model proaktywnej ochrony ścieżki przed awarią pojedynczego węzła w sieci z komutacji kanałów o scentralizowanym doborze tras bez współdzielenia przepustowości łączy. W pracy jest przyjęty wariant ochrony 1+1, a problem przydziału długości fali nie jest wzięty pod uwagę, a jedynie jego podproblem dotyczący doboru tras. Kolejne założenie dotyczy sposobu redukcji obciążenia sieci, która w niniejszej pracy jest osiągana poprzez zastosowanie dla każdej trasy podstawowej tylko jednej trasy zapasowej chroniącej całą ścieżkę podstawową przed awarią pojedynczego elementu. Jedną z pożądanych cech rozwiązań proponowanych w niniejszej rozprawie jest redukcja czasu obliczenia tras, aby m.in. ograniczyć ryzyko sklasyfikowania sekwencji awarii pojedynczych jako awarii wielokrotnej. Tryb ochrony 1+1 gwarantuje minimalny czas naprawienia transmisji równoważny różnicy opóźnień transmisji pomiędzy trasą zapasową i podstawową.

W dalszej części pracy, w rozdziale 3 są zaprezentowane dwa podejścia ukierunkowane na redukcję prawdopodobieństwa blokowania żądań w sieci szkieletowej poprzez dążenie do zrównoważenia obciążenia.



3 Zastosowanie metod transformacyjnych podczas doboru tras ukierunkowane na redukcję współczynnika blokowania żądań

Instalacja zapasowych tras w sieci dla ochrony przed awarią elementów trasy podstawowej wymaga rezerwacji dodatkowych przepustowości na łączach wykorzystywanych przez trasy zapasowe. Prowadzi to do zwiększenia stopnia wykorzystania zasobów sieci, co z kolei skutkuje wzrostem wartości współczynnika blokowania żądań (ang. blocking probability). Z tego powodu, redukcja współczynnika blokowania żądań stanowi istotny problem rozpatrywany w niniejszym rozdziale. Celem rozdziału jest prezentacja i weryfikacja charakterystyk autorskiej metody doboru tras ukierunkowanej na zrównoważenie obciążenia sieci.

Wyniki badań dotyczących redukcji współczynnika blokowania żądań dla doboru tras ukierunkowanego na zrównoważenie obciążenia sieci zostały w niniejszym rozdziale ukazane w odniesieniu do wyłącznie tras pojedynczych/podstawowych bez obliczenia tras zapasowych. Rozszerzenie proponowanych metod do przypadku niezawodnego doboru tras jest opisane w Aneksie 9.

Podrozdział 3.1 prezentuje metodę inspirowaną mechaniką kwantową. Proponowane podejście wymaga uprzedniego wyznaczenia wszystkich możliwych tras pomiędzy każdą parą węzłów dla podanej topologii sieci. Mimo, że metoda ta może być zastosowana dla wykrycia zawczasu wszystkich tras w sieci (wszystkie trasy pomiędzy każdą parą węzłów można obliczyć przed uruchomieniem proponowanego poniżej algorytmu), powstaje problem dotyczący aktualności tras w przypadku awarii. Uszkodzenie elementu sieci powoduje niedostępność obliczonych tras zawierających ten element, co z kolei powoduje nieaktualność przeprowadzonych wcześniej obliczeń i ogranicza efektywność metody w przypadku

awarii elementów sieci. Badania ukierunkowane na rozwiązanie tego problemu zostały opisane w podrozdziale 3.2.

Podrozdział 3.2 prezentuje wyniki własnych badań ukierunkowanych na redukcję czasu obliczeń dla koncepcji opisanej w podrozdziale 3.1. W szczególności, podejście przedstawione w podrozdziale 3.2 nie wymaga obliczeń wszystkich tras pomiędzy każdą parą węzłów w sieci lecz wyznacza ograniczoną liczbę k tras. Podsumowanie wyników badań możliwości zastosowania metod transformacyjnych dla redukcji współczynnika blokowania żądań w trakcie doboru tras zamyka niniejszy rozdział i zostało ukazane w podrozdziale 3.3.

3.1 Metoda redukcji prawdopodobieństwa blokowania żądań przy wyznaczaniu tras w sieciach o ograniczonych zasobach inspirowana teorią kwantową

W niniejszym podrozdziale przedstawiona jest autorska metoda doboru tras ukierunkowana na zrównoważenie obciążenia sieci, a tym samym także na redukcję współczynnika blokowania żądań ustanowienia tras transmisji w sieciach szkieletowych oraz na zwiększenie pojemności informacyjnej sieci przy niezmięnionej przepustowości. Opisane podejście zostało opisane w autorskiej pracy [48] i opiera się na mechanizmie przewidywania dostępności przepustowości łączy sieci i jest zainspirowane teorią kwantową. Wyniki symulacji przeprowadzonych dla trzech topologii sieci wskazują na znaczną redukcję (nawet dwukrotną) prawdopodobieństwa blokowania żądań w porównaniu do referencyjnego podejścia [49].

3.1.1 Wprowadzenie

W celu przeprowadzenia doboru tras w sieciach teleinformatycznych często są wykorzystane metody wykorzystujące teorię grafów pozwalającą modelować topologię sieci w postaci grafu $G=(V, E)$, w którym węzły sieciowe są reprezentowane poprzez zbiór wierzchołków grafu $V=\{v_1, v_2, v_3, \dots, v_{|V|}\}$, a łącza sieci poprzez krawędzie grafu $E=\{e_1, e_2, e_3, \dots, e_{|E|}\}$. Aby wyznaczyć trasę o najmniejszym możliwym koszcie między zadaną parą węzłów krańcowych, stosowane są algorytmy grafowe znajdowania ścieżki o najmniejszym koszcie pomiędzy dwoma wierzchołkami grafu. Ścieżka w grafie jest ciągiem wierzchołków grafu $P=(v_s, \dots, v_t)$, w którym każdy następny wierzchołek jest połączony ze swoim

poprzednikiem poprzez krawędź grafu. Pierwszy w ciągu wierzchołek v_s i ostatni wierzchołek v_t są wierzchołkami źródłowym i docelowym żądania, a pozostałe wierzchołki są określane jako wierzchołki tranzytowe. Sumaryczny koszt wszystkich krawędzi wykorzystanych dla połączenia wierzchołków zawartych w P jest kosztem ścieżki $\mathcal{C}(P)$. Algorytm Dijkstry [50] jest często stosowany dla wyznaczenia tras od wierzchołka źródłowego v_s do wszystkich pozostałych wierzchołków grafu lub, po modyfikacji, do uzyskania pojedynczej trasy do danego wierzchołka docelowego v_t .

Wyznaczenie w sposób sekwencyjny tras dla kolejnych żądań d_i w sieciach o ograniczonych zasobach nie prowadzi do uzyskania rozwiązania optymalnego w sensie globalnym [51] (Rys. 2.6). W celu realizacji efektywnego (w sensie czasu obliczeń) wyznaczenia tras w sieciach szkieletowych powszechnie stosuje się wariant scentralizowanego doboru tras, tj. przeprowadzenia wszystkich obliczeń przez jeden węzeł sieci (zwany w niniejszej rozprawie centrum zarządzania ruchem sieciowym – CZRS). W ogólnym przypadku, dla każdego żądania unicastowego d_i określona jest para węzłów: źródłowy $v_s(d_i)$ i docelowy $v_t(d_i)$ oraz wymagana przepustowość transmisji $\mathcal{C}(d_i)$. W sieciach rozległych charakteryzujących się małymi zmianami łącznego natężenia ruchu, CZRS zawczasu odbiera informację o wszystkich żądaniach D i ustanawia trasy sekwencyjnie dla kolejnych żądań, aktualizując wartość metryki kosztu łączy $\mathcal{C}(e_i)$ po wykryciu każdej trasy. Wzór (3.1) prezentuje znaną z literatury (i referencyjną w niniejszym rozdziale) postać metryki $\mathcal{C}(e_i)$ ukierunkowaną na równoważenie obciążenia łączy w sieci [49].

$$c(e_i) = \frac{g(e_i) + \varepsilon}{b(e_i)} \quad (3.1)$$

gdzie:

- $g(e_i)$ – część przepustowości łącza e_i zarezerwowana dotychczas w celu obsługi żądań transmisji,
- $b(e_i)$ – nominalna przepustowość łącza e_i .

Zgodnie z wzorem (3.1), po wyznaczeniu trasy obsługi następnego żądania, wartość metryki kosztu łącza ulega zwiększeniu proporcjonalnie do zmniejszania się dostępnej przepustowości łącza. Składnik ε (bardzo mała liczba) ma na celu

zapewnienie niezerowej wartości metryki kosztu łączy o całkowitej dostępnej przepustowości (aby umożliwić zastosowanie algorytmów doboru tras wymagających niezerowych wartości kosztu łączy). Zastosowanie wzoru (3.1) w doborze tras sprawia, że wyznaczone trasy przeważnie omijają łączy, których stopień dostępności zasobów zbliża się do zera, co z kolei prowadzi do zmniejszenia liczby żądań odrzuconych z uwagi na niedostępność zasobów sieci.

Powyższy sposób obliczenia metryki kosztu łączy opiera się wyłącznie na aktualnej informacji o zasobach zarezerwowanych dla dotychczas obsługiwanych połączeń i niestety nie bierze pod uwagę żądań, których trasy nie zostały jeszcze wyznaczone, co może zwiększyć ryzyko odrzucenia kolejnych żądań (z powodu niedostępności zasobów sieci). W celu efektywnego ograniczenia skali tego problemu, w bieżącym rozdziale jest przedstawiony mechanizm przewidywania dostępności zasobów łączy, którego zadaniem jest pozostawienie części zasobów łączy dla ich późniejszego wykorzystania w odniesieniu do jeszcze nieobsłużonych żądań.

Metoda prezentowana w niniejszym rozdziale jest inspirowana mechaniką kwantową [52]-[55]. W szczególności, mechanika kwantowa operuje pojęciem kwantu jako minimalnej możliwej części jakiegokolwiek jednostki fizycznej. Drugim ważnym założeniem mechaniki kwantowej jest stwierdzenie, że kwant q może istnieć jednocześnie w kilku lokalizacjach przestrzeni z określonym prawdopodobieństwem [52], [53]. Wykorzystanie tego założenia w prezentowanej metodzie pozwala przeprowadzić operacje w odniesieniu do elementów (tutaj tras transmisji) o nieokreślonej w danej chwili ostatecznej postaci.

W odniesieniu do zadania alokacji zasobów, kwant może jednocześnie (z pewnym prawdopodobieństwem) wykorzystywać część zasobów wielu elementów w przestrzeni. Ilość zasobów wykorzystanych przez kwant q w przestrzeni (x, y) jest równoważna prawdopodobieństwu lokalizacji kwantu q w określonym miejscu przestrzeni (x, y) . Przykład rozkładu prawdopodobieństw wykorzystania zasobów przestrzeni przez kwant q_1 jest ukazany w Tabeli 3.1.

Jeżeli w przestrzeni istnieje wiele kwantów (pozycje kwantów q_2 , q_3 i q_4 są prezentowane w Tabelach 3.2-3.4), możliwe jest obliczenie sumarycznej macierzy, której wartości elementów ukazują sumaryczny stopień istotności danego zasobu dla wszystkich kwantów (Tabela 3.5).

Współrzędne kwantu q_1		x				
		1	2	3	4	5
y	1	0.010	0.020	0.040	0.020	0.010
	2	0.025	0.050	0.100	0.050	0.025
	3	0.030	0.060	0.120	0.060	0.030
	4	0.025	0.050	0.100	0.050	0.025
	5	0.010	0.020	0.040	0.020	0.010

Tabela 3.1 Wykorzystanie zasobów przestrzeni przez kwant q_1

Współrzędne kwantu q_2		x				
		1	2	3	4	5
Y	1	0.010	0.10	0.40	0.10	0.010
	2	0.010	0.010	0.10	0.010	0.010
	3	0.010	0.010	0.010	0.010	0.010
	4	0.000	0.000	0.000	0.000	0.000
	5	0.000	0.000	0.000	0.000	0.000

Tabela 3.2 Wykorzystanie zasobów przestrzeni przez kwant q_2

Współrzędne kwantu q_3		x				
		1	2	3	4	5
y	1	0.010	0.010	0.010	0.010	0.010
	2	0.010	0.045	0.050	0.045	0.010
	3	0.010	0.050	0.46	0.050	0.010
	4	0.010	0.045	0.050	0.045	0.010
	5	0.010	0.010	0.010	0.010	0.010

Tabela 3.3 Wykorzystanie zasobów przestrzeni przez kwant q_3

Współrzędne kwantu q_4		x				
		1	2	3	4	5
y	1	0.020	0.100	0.020	0.020	0.010
	2	0.100	0.300	0.100	0.020	0.010
	3	0.020	0.100	0.020	0.020	0.010
	4	0.020	0.020	0.020	0.020	0.010
	5	0.010	0.010	0.010	0.010	0.000

Tabela 3.4 Wykorzystanie zasobów przestrzeni przez kwant q_4

Współrzędne kwantów		x				
		1	2	3	4	5
y	1	0.050	0.230	0.470	0.150	0.04
	2	0.145	0.495	0.350	0.215	0.055
	3	0.070	0.220	0.610	0.140	0.060
	4	0.055	0.115	0.180	0.115	0.045
	5	0.030	0.040	0.060	0.040	0.020

Tabela 3.5 Sumaryczna macierz wykorzystania zasobów przestrzeni

Informacja o prawdopodobieństwie występowania kwantu w danej lokalizacji zasobów wraz z sumaryczną macierzą prawdopodobieństw wykorzystania zasobów przestrzeni pozwalają ocenić poziom konfliktów pomiędzy kwantami. Im większa jest wartość elementu macierzy sumarycznej, tym większe prawdopodobieństwo konfliktu pomiędzy kwantami w ubieganiu się o konkretną lokalizację zasobu w przestrzeni. Mimo, iż w literaturze z obszaru fizyki brak jest zastosowania analogicznego podejścia, proponowana w niniejszym rozdziale koncepcja wydaje się perspektywiczna w odniesieniu do doboru tras ukierunkowanego na zrównoważenie obciążenia w sieciach teleinformatycznych.

Odwzorowując powyższe założenia na zadanie doboru tras możemy rozpocząć realizację żądania instalacji trasy pomiędzy węzłami sieci od określenia kwantu, wykorzystującego łącza sieci dla transmisji danych z określonymi prawdopodobieństwami. W przypadku kiedy trasa nie jest jeszcze wyznaczona, „trasa kwantowa” wykorzystuje wszystkie łącza w sieci z różnym prawdopodobieństwem. Pozwala to reprezentować trasę kwantową pomiędzy dwoma węzłami jako wektor kosztów wszystkich łączy sieci, w której wartości elementów odpowiadają prawdopodobieństwom wykorzystania łączy na potrzeby instalacji trasy. Taka reprezentacja daje możliwość agregacji informacji o prognozowanych konfliktach dotyczących przepustowości łączy względem zbioru żądań (poprzez zbudowanie sumarycznej macierzy potencjalnego wykorzystania łączy). To z kolei pozwala opracować metrykę kosztu łączy sieci wobec prognozowanych konfliktów, która przyczyniałaby się do redukcji prawdopodobieństwa blokowania żądań w sposób opisany poniżej.

3.1.2 Podstawy nowej metody

Prezentowane podejście bazuje na zastosowaniu zamiast metryki kosztu łączy (3.1) autorskiej metryki wyrażonej wzorem (3.2), określonej w niniejszej pracy mianem „metryki kwantowej”.

$$c(e_i) = \left(\frac{g(e_i) + c(d_j)}{b(e_i)} \right) \beta + w(e_i) \alpha \quad (3.2)$$

gdzie:

- $c(d_j)$ – przepustowość wymagana dla żądania d_j
- $b(e_i)$ – nominalna przepustowość łączy e_i
- $g(e_i)$ – sumaryczna przepustowość łączy e_i zarezerwowana na potrzeby instalacji wyznaczonych dotychczas tras
- $w(e_i)$ – stopień prognozowanych konfliktów na łączy e_i . (wartości wektora W są obliczane w sposób opisany w sekcji 3.1.5)
- β – waga stopnia zajętości łączy w całkowitym koszcie łączy
- α – waga miary prognozowanych konfliktów dotyczących alokacji zasobów na łączy e_i dla kolejnych żądań w całkowitym koszcie łączy

Współczynniki α i β powiązane są pomiędzy sobą zależnością (3.3).

$$\alpha + \beta = 1 \quad (3.3)$$

W proponowanym wzorze (3.2), koszt łączy $c(e_i)$ zależy nie tylko od zarezerwowanej przepustowości łączy $g(e_i)$ ale również od przepustowości wymaganej dla instalacji trasy $c(d_j)$. Podejście to ma dwie zasadnicze zalety. Pierwsza zaleta polega na tym, że składnik ten wskazuje nie na stopień alokacji zasobów na łączy przed, lecz po zainstalowaniu na tym łączy wyznaczanej trasy. Podejście to pozwala więc na redukcję ryzyka wyczerpania dostępnej przepustowości łączy sieci po zainstalowaniu danej trasy. Drugą zaletą jest to, że takie rozwiązanie pozwala efektywnie funkcjonować w sieciach o różnych przepustowościach łączy. Przykładowo, założmy że mamy dwa łączy e_1 oraz e_2 , których nominalne przepustowości wynoszą odpowiednio $b(e_1) = 10$ kanałów,

$b(e_2) = 80$ kanałów, a przepustowości przydzielone istniejącym trasom na tych łączach wynoszą odpowiednio $g(e_1) = 4$ kanały i $g(e_2) = 40$ kanałów. W celu instalacji kolejnej trasy wymagana jest np. przepustowość $c(d) = 4$ kanały. Ze względu na znikomą wartość ε , metryka (3.1) określa koszt rozważanych łączy jako $c(e_1) = 4/10 = 0.4$, $c(e_2) = 40/80 = 0.5$. Jednak oczywistym jest, że drugie łącze e_2 mające 40 niewykorzystanych kanałów jest odpowiedniejsze dla instalacji trasy o przepustowości 4 kanałów niż łącze e_1 o liczbie wolnych kanałów wynoszącej 6, gdyż instalacja trasy na łączu e_2 prowadzi do znacznie mniejszego procentowego spadku dostępnych zasobów łącza e_2 niż w przypadku instalacji trasy na łączu e_1 . Zastąpienie w metryce (3.2) składnika ε poprzez przepustowość bieżącego żądania $c(d_i)$ sprawia, że metryka ta zwraca wartości $c(e_1) = (4+4)/10 = 0.8$, $c(e_2) = (40+4)/80 = 0.55$ właściwie skorelowane z dostępną przepustowością łączy, co skutkuje wyborem drugiego łącza (e_2).

Ważnym elementem metryki kwantowej jest również poziom prognozowanego konfliktu $w(e_i)$, który jest tym większy, im większe jest prawdopodobieństwo rywalizacji o zasoby (przepustowości) łącza e_i (sposób obliczenia $w(e_i)$ jest opisany w sekcji 3.1.4). W przypadku wysokich wartości α , koszt metryki łącza zależy generalnie od prognozowanych konfliktów (tj. od wartości $w(e_i)$), a algorytm doboru tras w znacznym stopniu ignoruje informację o stopniu zajętości przepustowości łączy. Wysokie wartości β powodują z kolei ograniczenie roli informacji o możliwych przewidywanych konfliktach. Dyskusja właściwego doboru balansu pomiędzy współczynnikami α i β jest zawarta w sekcji 3.1.5.

Ważnym elementem prezentowanej koncepcji jest kwant q , zdefiniowany w niniejszej pracy jako trójka $(v_1(q), v_2(q), W(q))$, gdzie $v_1(q), v_2(q)$ są parą węzłów: źródłowym i docelowym transmisji, a $W(q)$ jest wektorem (o długości równej liczbie łączy sieci), którego każdy element $w_i(q)$ reprezentuje prawdopodobieństwo użycia łącza e_i dla instalacji trasy pomiędzy węzłami $v_1(q)$ a $v_2(q)$. Wartość elementu $w_i(q)$ jest obliczona za pomocą wzoru (3.4).

Przykładowo, Rys. 3.1 prezentuje wartości wektora $W(q)$ dla każdego łącza sieci w odniesieniu do tras możliwych do ustanowienia pomiędzy węzłami v_a i v_b . Są tylko dwie możliwe trasy pomiędzy węzłami v_a i v_b : $P_1 = (v_a, 2, 1, 3, v_b)$ oraz $P_2 = (v_a, 2, 4, 3, v_b)$. Łącza $(v_a, 2)$ oraz $(3, v_b)$ są zawarte w dwóch trasach (z dwóch możliwych), dlatego wartości ich współczynników $w(q)$ są równe 1. Łącza $(2, 1)$,

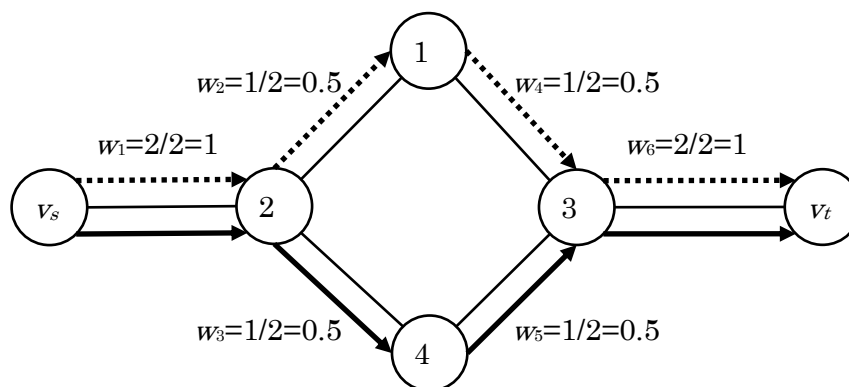
(1, 3), (2, 4) i (4, 3) są z kolei zawarte tylko w jednej z możliwych dwóch tras, toteż w ich przypadku wartości $w(q)$ wynoszą 0.5.

$$w_i(q) = \frac{|R(e_i) \cap R(v_1(q), v_2(q))|}{|R(v_1(q), v_2(q))|} \quad (3.4)$$

gdzie:

$R(e_i)$ – zbiór wszystkich tras w sieci zawierających łącze e_i

$R(v_1(q), v_2(q))$ – zbiór wszystkich tras w sieci istniejących pomiędzy węzłami $v_1(q)$ a $v_2(q)$.



Rys. 3.1 Przykład wartości elementów $W(q)$

W celu wykrycia wszystkich tras w sieci pomiędzy parą węzłów jest możliwe zastosowanie algorytmu Depth-first search (DFS) [56] zmodyfikowanego w następujący sposób:

- Zmodyfikowany DFS nie koloruje wykrytych wierzchołków, lecz dodaje ich do pośredniego wektora P (Krok 1 w poniższym pseudokodzie), który ukazuje aktualną wykrytą trasę.
- Zmodyfikowany DFS dla każdego węzła sąsiedniego uruchamia iteracyjnie rekurencję DFS z lokalną kopią wektora P (Krok 2, wiersze: 6, 7).

Zmodyfikowany algorytm DFS: $DFS_m(M, v_s, v_t, Z, P)$

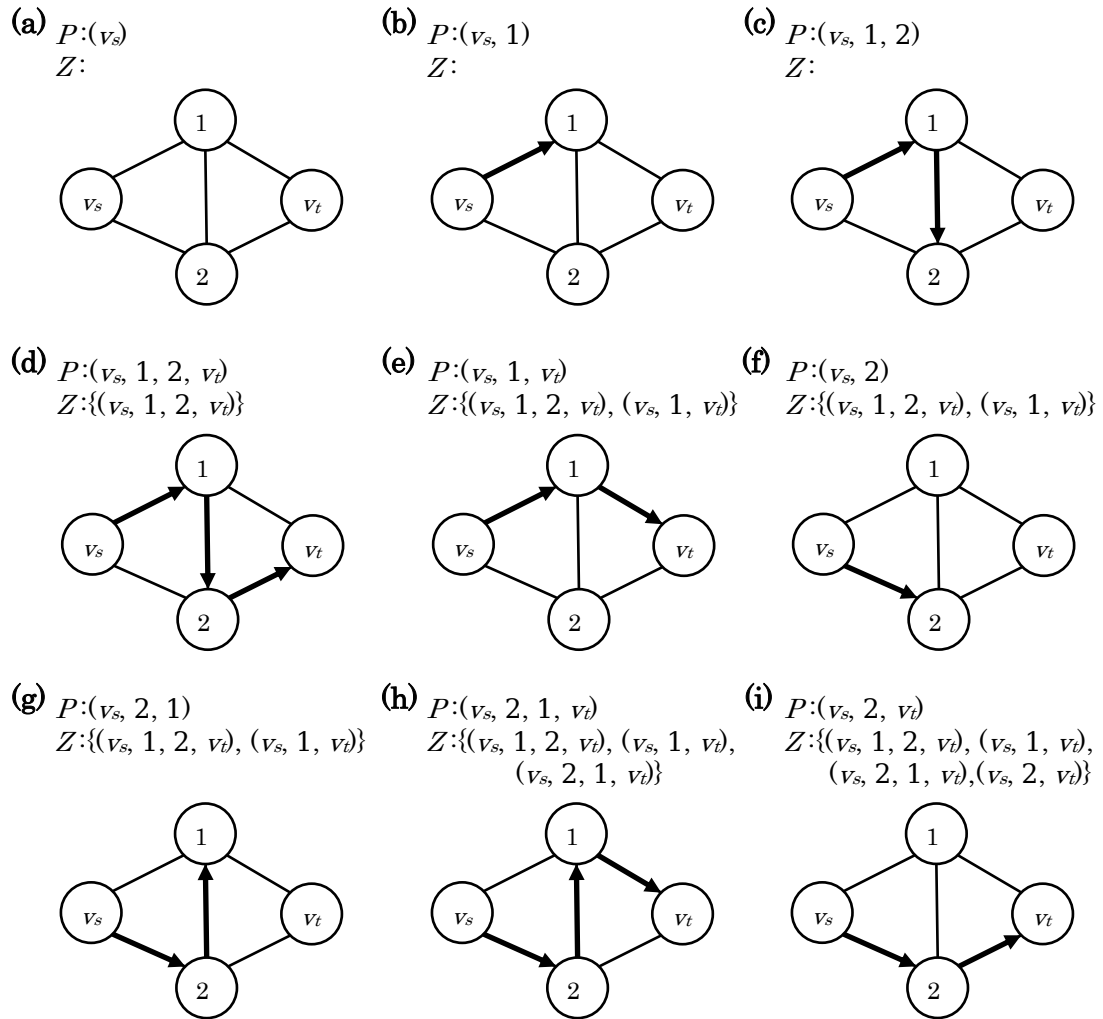
Parametry wejściowe:	
M	- Macierz sąsiedztwa grafu topologii sieci
v_s	- Wierzchołek źródłowy
v_t	- Wierzchołek docelowy
Z	- Zbiór wykrytych tras (jest pusty w pierwszej iteracji)
P	- Wykryta trasa (jest pusta w pierwszej iteracji)
	Krok 1:
1	Dodaj v_s do P ;
	Krok 2:
2	Jeżeli: $v_s=v_t$:
3	Dodaj P do Z ;
	W przeciwnym razie:
4	Dla każdego wierzchołka v_i sąsiedniego względem wierzchołka v_s :
5	Jeżeli $v_i \notin P$:
6	Utwórz nową kopię P_1 wektora P ;
7	Uruchom $DFS_m(M, v_s, v_t, Z, P_1)$;
8	Koniec;

W pierwszej iteracji algorytm dodaje źródłowy wierzchołek v_s do trasy P jako jej pierwszy wierzchołek (wiersz 1). W Kroku 2 algorytm sprawdza czy wierzchołek źródłowy nie jest wierzchołkiem docelowym. W pierwszej iteracji warunek (wiersz 2) nigdy nie będzie spełniony. W wierszu 4 algorytm operując na macierzy sąsiedztwa wykrywa wszystkie sąsiednie wierzchołki v_i dla węzła źródłowego i dla każdego z nich uruchamia rekurencyjnie algorytm DFS_m przekazując sąsiedni węzeł jako źródłowy i nową kopię trasy, w której jest zawarty v_i jako pierwszy wierzchołek. Każda kopia DFS_m doda przekazany jej sąsiedni wierzchołek do trasy jako węzeł następnego przeskoku do swojej lokalnej kopii trasy (wiersz 1) i powtarza tę samą procedurę dla wszystkich węzłów sąsiednich niezawartych w trasie. Warunek w wierszu 5 blokuje możliwość dodania do trasy węzłów powtórnie. Kiedy DFS_m osiąga wierzchołek docelowy ($v_s=v_t$), algorytm dodaje go do trasy i po sprawdzaniu warunku w wierszu 2 dodaje wykrytą trasę do zbioru Z . Tym samym bieżąca rekurencja kończy swoje działanie. Po zakończeniu działania wszystkich rekurencyjnych wywołań (również pierwszej rekurencji, która jest uruchomiona na wierzchołku docelowym), zbiór Z zawiera wszystkie możliwe trasy pomiędzy daną parą węzłów sieci.

Przykładowo, dla topologii sieci prezentowanej na Rys. 3.2(a) i kwantu $v_1(q)=v_s$, $v_2(q)=v_t$, po uruchomieniu $DFS_m(M, v_s, v_t, Z, P)$ algorytm doda v_s do wektora P jako pierwszy węzeł. Węzeł v_s nie jest węzłem docelowym v_t , dlatego algorytm wybiera pierwszego sąsiada węzła v_s , którym jest węzeł 1 i tworzy kopię



P wektora P . Rekurencyjne uruchamia $DFS_m(M, 1, v_t, Z, (v_s))$ z lokalną kopią P zamiast P oraz węzłem 1 zamiast v_s .



Rysunek 3.2 Przykład działania algorytmu DFS_m

Rekurencyjny $DFS_m(M, 1, v_t, Z, (v_s))$ dodaje węzeł 1 na koniec swojego wektora $P=(v_s, 1)$ (Rys. 3.2(b)). Algorytm sprawdza, że węzeł 1 nie jest docelowym v_t (wiersz 2) i wybiera pierwszego sąsiada węzła 1. Pierwszym sąsiadem jest węzeł v_s i algorytm wybiera następnego sąsiada z powodu zawartości v_s w wektorze $P=(v_s, 1)$ (wiersz 5). Następnym sąsiadem węzła 1 jest węzeł 2, który nie jest zawarty w wektorze P . DFS_m tworzy kopię wektora $P=(v_s, 1)$ i inicjuje rekurencyjnie nowe wywołanie $DFS_m(M, 2, v_t, Z, (v_s, 1))$.

Rekurencyjne wywołanie algorytmu dodaje węzeł 2 na koniec swojej lokalnej kopii wektora $P=(v_s, 1, 2)$ (Rys. 3.2(c)). Algorytm sprawdza, że węzeł 2 nie jest

węzłem docelowym (wiersz 2) i wybiera pierwszego sąsiada wężła 2. Pierwszym sąsiadem jest wężel v_s , który jest zawarty w $P=(v_s, 1)$. Drugim sąsiadem jest wężel, który również jest zawarty w wektorze P . Trzecim sąsiadem jest wężel v_t , dla którego algorytm rekurencyjnie uruchamia swoją kopię $DFS_m(M, v_t, v_t, Z, (v_s, 1, 2))$.

Uruchomienie algorytmu DFS_m z parametrami $DFS_m(M, v_t, v_t, Z, (v_s, 1, 2))$ dodaje wężel v_t na koniec swojej lokalnej kopii wektora $P=(v_s, 1, 2, v_t)$. Algorytm ustala, że wężel v_t jest wężłem docelowym i dodaje wektor $(v_s, 1, 2, v_t)$ do zbioru wykrytych tras Z , który jest wspólny dla wszystkich wywołań DFS_m (Rys. 3.2(d)). Na tym etapie wywołanie $DFS_m(M, v_t, v_t, Z, (v_s, 1, 2))$ się zakończy, a sterowanie zostanie przekazane do $DFS_m(M, 2, v_t, Z, (v_s, 1))$.

$DFS_m(M, 2, v_t, Z, (v_s, 1))$ nie wykrywa nowych sąsiadów wężła 2 i przekazuje sterowanie do wywołania $DFS_m(M, 1, v_t, Z, (v_s))$. $DFS_m(M, 1, v_t, Z, (v_s))$ przechodzi do następnego sąsiada wężła 1, którym jest wężel v_t . Algorytm zakłada nową kopię wektora $P=(v_s)$ i rekurencyjnie uruchamia nową wersję $DFS_m(M, v_t, v_t, Z, (v_s, 1))$. $DFS_m(M, v_t, v_t, Z, (v_s, 1))$ dodaje v_t do wektora $P=(v_s, 1, v_t)$. Algorytm wykrywa, że v_t jest wężłem docelowym i dodaje wykrytą trasę (wektor P) do wspólnego zbioru wykrytych tras (Rys. 3.2(e)). Zatem przekazuje sterowanie do wywołania $DFS_m(M, 1, v_t, Z, (v_s))$.

$DFS_m(M, 1, v_t, Z, (v_s))$ nie wykrywa nowych sąsiadów i przekazuje sterowanie do wywołania $DFS_m(M, v_s, v_t, Z, P)$, które wybiera następnego sąsiada wężła v_s i uruchamia swoją kopię $DFS_m(M, 2, v_t, Z, (v_s))$. $DFS_m(M, 2, v_t, Z, (v_s))$ dodaje wężel na koniec wektora $P=(v_s, 2)$ (Rys. 3.2(f)) i wybiera sąsiada wężła 2. Wężel v_s jest zawarty w $P=(v_s, 2)$, więc algorytm wybiera kolejnego sąsiada: jest nim wężel 1. Algorytm zakłada kopię wektora P i rekurencyjnie uruchamia $DFS_m(M, 1, v_t, Z, (v_s, 2))$.

Wywołanie $DFS_m(M, 1, v_t, Z, (v_s, 2))$ dodaje na koniec wektora P wężel 1 (Rys. 3.2(g)) i wybiera sąsiada, dla którego rekurencyjnie uruchamia swoją kopię $DFS_m(M, v_t, v_t, Z, (v_s, 2))$. $DFS_m(M, v_t, v_t, Z, (v_s, 2))$ dodaje v_t na koniec wektora $P=(v_s, 2, 1, v_t)$. Algorytm wykrywa, że wężel v_t jest wężłem docelowym i dodaje trasę P do zbioru wspólnych tras Z (Rys. 3.2(h)). Tym samym, wywołanie $DFS_m(M, v_t, v_t, Z, (v_s, 2))$ kończy swoje działanie i przekazuje sterowanie do $DFS_m(M, 1, v_t, Z, (v_s, 2))$. Z uwagi na to, że $DFS_m(M, 1, v_t, Z, (v_s, 2))$ przebadal

wszystkie węzły sąsiednie (węzły v_s i 2 są zawarte w P , a dla v_t już została uruchomiona kopia DFSm), sterowanie zostaje przekazane do $DFSm(M, 2, v_t, Z, (v_s))$. Wywołanie $DFSm(M, 2, v_t, Z, (v_s))$ wybiera nowego sąsiada węzła 2, którym jest węzeł v_t i rekurencyjnie uruchamia $DFSm(M, v_t, v_t, Z, (v_s, 2))$. $DFSm(M, v_t, v_t, Z, (v_s, 2))$ dodaje v_t na koniec wektora $P=(v_s, 2, v_t)$. Wykrywa, że węzeł v_t jest docelowym i dodaje wektor $(v_s, 2, v_t)$ do zbioru wykrytych tras Z (Rys. 3.2(i)). Tym samym $DFSm(M, v_t, v_t, Z, (v_s, 2))$ kończy swoje działanie i przekazuje sterowanie do $DFSm(M, 2, v_t, Z, (v_s))$.

$DFSm(M, 2, v_t, Z, (v_s))$ z kolei przekazuje sterowanie do $DFSm(M, v_s, v_t, Z, P)$. Algorytm nie wykrywa innych sąsiadów węzła v_s i kończy swoje działanie. Po zakończeniu działania algorytmu, w zbiorze Z są zawarte wszystkie możliwe trasy pomiędzy v_t a v_s : $\{(v_s, 1, 2, v_t), (v_s, 1, v_t), (v_s, 2, 1, v_t), (v_s, 2, v_t)\}$. Formalne udowodnienie poprawności algorytmu DFSm jest zaprezentowane w Aneksie 1. Złożoność algorytmiczna DFSm jest z kolei poddana analizie w Aneksie 2.

Dysponując informacją o wszystkich możliwych trasach pomiędzy parą węzłów, możliwe jest następnie obliczenie ile razy każde łącze e_i jest zawarte w trasach ze zbioru Z (jako $|R(e_i) \cap R(v_1(q), v_2(q))|$). Dlatego warto jest zbudować wektor pośredni X , którego każdy element x_i odpowiada liczbie występowania łącza e_i w trasach. Definiując wektor X z elementami o początkowej zerowej wartości, możliwe jest przebadanie każdej trasy ze zbioru Z , zwiększając o 1 wartości x_i które odpowiadają zawartym w nich łączom. Przekształcając wzór (3.4) możemy obliczyć wartość każdego elementu $w_i(q)$ wektora $W(q)$ na podstawie wzoru (3.5), tj. równą ilorazowi liczby tras ze zbioru Z zawierających łącze e_i do całkowitej liczby tras zbioru Z . Iloraz ten odpowiada również prawdopodobieństwu wykorzystania łącza e_i na potrzeby instalacji trasy pomiędzy parą węzłów $v_1(q)$ oraz $v_2(q)$.

$$w_i(q) = \frac{x_i}{|Z|} \quad (3.5)$$

W następnej sekcji jest zaprezentowany sposób wykorzystania tras kwantowych w celu zbudowania wektora prognozowanych konfliktów dotyczących przepustowości łącza sieci.



3.1.3 Wyznaczenie kwantu (q) dla danego żądania

W niniejszej sekcji prezentowany jest algorytm budowania kwantu, który obejmuje pięć głównych kroków. Algorytm pobiera jako dane wejściowe macierz sąsiedztwa M , parę węzłów krańcowych żądania: v_s, v_t , a zwraca w wyniku swojego działania kwant q .

Algorytm budowania kwantu

Parametry wejściowe:	
M	- Macierz sąsiedztwa
v_s	- Wierzchołek źródłowy
v_t	- Wierzchołek docelowy
Parametry wyjściowe:	
q	- kwant
	Krok 1:
1	Utwórz zbiory pośredniczące $P=\emptyset; Z=\emptyset$;
2	Na podstawie M zdefiniuj wektor wszystkich łączy E ;
3	Wyzeruj wartości każdego elementu wektora E ;
	Krok 2:
4	Uruchom algorytm $DFS_m(M, v_s, v_t, Z, P)$
	Krok 3:
5	Dla każdej trasy P w zbiorze Z :
6	Dla każdego łącza e_i zawartego w P :
7	Inkrementuj wartość elementu wektora E :
	$e_i := e_i + 1$;
8	Krok 4:
9	Dla każdego elementu e_i wektora E :
10	Oblicz nową wartość:
	$e_i := e_i \setminus Z$;
	Krok 5:
11	Utwórz kwant q ;
12	Ustaw: $v_1(q) := v_s; v_2(q) := v_t; W(q) := E$;
13	Zwróć q ;
14	Koniec

W Kroku 1 algorytm definiuje pusty wektor tras P i pusty zbiór wykrytych tras Z (wiersz 1), które są niezbędne do uruchomienia algorytmu DFS_m w Kroku 2. Algorytm transformuje następnie macierz sąsiedztwa M w wektor łączy sieci E (wiersz 2) i ustawia wartość każdego elementu tego wektora jako równą zero (wiersz 3). Wartość elementu e_i wektora E oznacza liczbę tras zawierających

łącze e_i . Na początku działania algorytmu ona jest równa 0, gdyż na tym etapie nie zostało jeszcze przeprowadzone wykrycie tras pomiędzy węzłami v_s oraz v_t . W Kroku 2 algorytm uruchamia algorytm DFSm w celu wykrycia wszystkich istniejących tras w sieci pomiędzy węzłami v_s oraz v_t . W wyniku realizacji Kroku 2 wszystkie trasy wykryte poprzez DFSm są umieszczane w zbiorze Z . W Kroku 3 algorytm analizuje każdą trasę w zbiorze Z inkrementując wartości e_i wektora łącz E jeżeli łącze e_i jest zawarte w badanej trasie (wiersz 7). W wyniku realizacji tego kroku, każdy element e_i wektora E uzyskuje wartość równą liczbie tras w Z zawierających łącze e_i . Mówiąc inaczej $e_i = |R(e_i) \cap R(v_1(q), v_2(q))|$. W Kroku 4 algorytm normalizuje wartości dla każdego elementu wektora łącz e_i , dzieląc je wartości przez liczbę wszystkich tras w zbiorze Z (wiersz 10). W ten sposób uzyskuje wartość $e_i = |R(e_i) \cap R(v_1(q), v_2(q))| / |R(v_1(q), v_2(q))|$, która jest równa wartości $w_i(q)$ we wzorach (3.4) i (3.5). Następnie, w Kroku 5, algorytm definiuje kwant q jako strukturę trzech elementów: dwóch węzłów $v_1(q)$, $v_2(q)$ i wektora prawdopodobieństwa użycia łącz $W(q)$ (wiersz 11) i ustawia wartości odpowiadających im elementów (wiersz 12). Finalnie, algorytm zwraca wyznaczony kwant q (wiersz 13) i kończy swoje działanie.

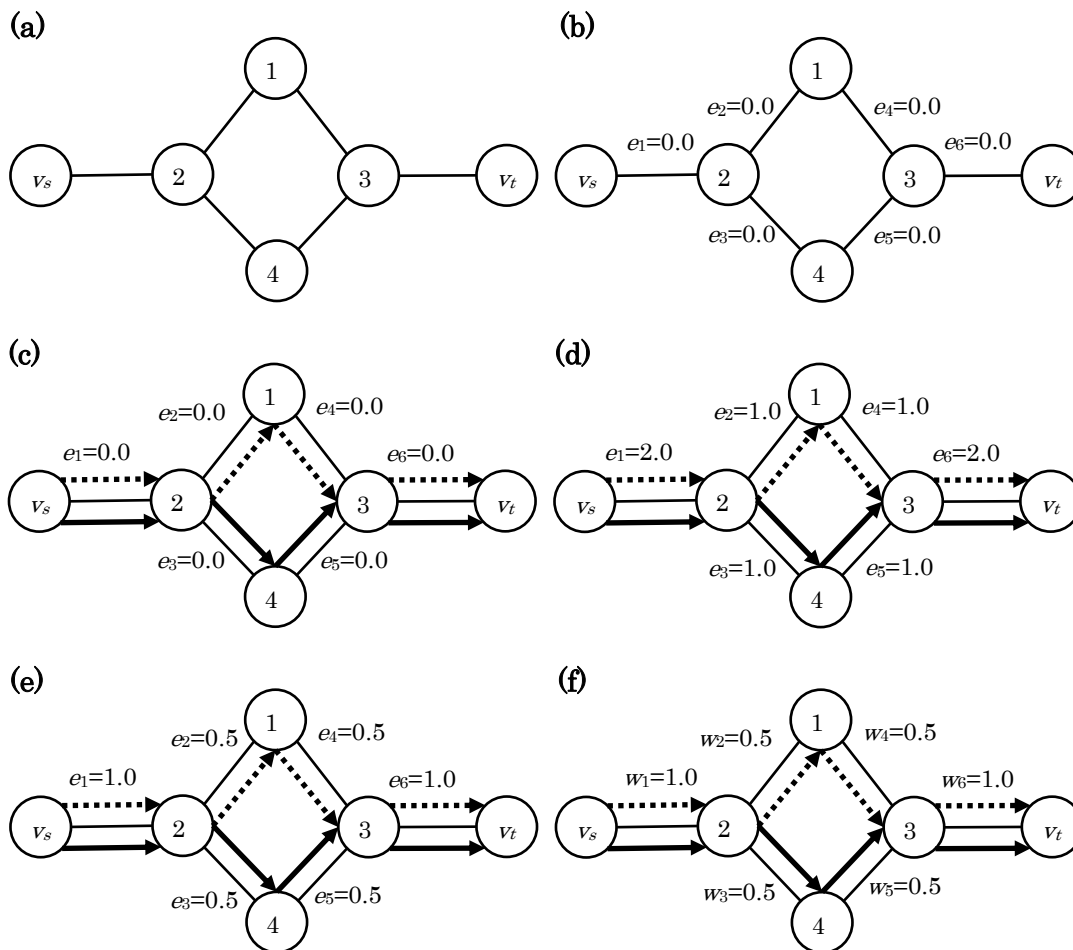
Przykład

Załóżmy, że algorytm budowania kwantu jest uruchomiony dla macierzy sąsiedztwa sieci zaprezentowanej na Rys. 3.3(a) i węzłów v_s , v_t tej sieci. W Kroku 1 algorytm buduje pusty wektor tras $P = \emptyset$ oraz pusty zbiór wykrytych tras $Z = \emptyset$. Definiuje także na podstawie macierzy sąsiedztwa sieci wektor łącz E . Przypisuje każdemu elementowi e_i wartość zero (wynik tej operacji jest zaprezentowany na Rys. 3.3(b)).

W Kroku 2 algorytm uruchamia algorytm DFSm, który ma na celu wykrycie wszystkich możliwych tras w sieci pomiędzy węzłami v_s i v_t oraz umieszczenie ich w zbiorze wykrytych tras $Z = \{(v_s, 2, 1, 3, v_t), (v_s, 2, 4, 3, v_t)\}$. Wykryte trasy są przedstawione na Rys. 3.3(c).

W Kroku 3 algorytm analizuje każdą trasę ze zbioru Z i inkrementuje wartości odpowiednich elementów wektora łącz E jeżeli trasa zawiera dane łącze. Trasa $(v_s, 2, 1, 3, v_t)$ zawiera łącza $e(v_s, 2)$, $e(2, 1)$, $e(1, 3)$, $e(3, v_t)$, które odpowiadają elementom e_1 , e_2 , e_4 , e_6 . Trasa $(v_s, 2, 4, 3, v_t)$ zawiera łącza $e(v_s, 2)$, $e(2, 4)$, $e(4, 3)$, $e(3, v_t)$, które odpowiadają elementom e_1 , e_3 , e_5 , e_6 . W takim razie

wartości elementów e_1, e_6 zostaną inkrementowane dwa razy, a wartości e_2, e_3, e_4, e_5 zostaną inkrementowane jeden raz. Wynik Kroku 3 jest zaprezentowany na Rys. 3.3(d).



Rys. 3.3 Przykład działania algorytmu budowania kwantu

W Kroku 4 algorytm wylicza znormalizowane zgodnie ze wzorem (3.5) wartości elementów wektora łącz, dzieląc ich wartości przez liczbę wykrytych tras $|Z| = |\{(v_s, 2, 1, 3, v_t), (v_s, 2, 4, 3, v_t)\}| = 2$. Te znormalizowane wartości są ukazane na Rys. 3.3(e). W Kroku 5 algorytm definiuje kwant q wraz z wektorem prawdopodobieństw stosowania łącz $\mathcal{W}(q)$ w oparciu o wyliczone wartości wektora łącz E (Rys. 3.3(f)). Algorytm zwraca kwant q i kończy działanie.

Mając informację o topologii sieci możliwe jest zbudowanie wszystkich istniejących kwantów w sieci, których liczba jest równa liczbie istniejących par węzłów w sieci (wzór (3.6)).

$$|Q| = \frac{|V|(|V| - 1)}{2} \quad (3.6)$$

3.1.4 Wyznaczenie wektora prognozowanych konfliktów

W wyniku użycia algorytmu budowania kwantu za pomocą algorytmu DFSm możliwe jest dokonanie transformacji topologii sieci do innej struktury będącej zbiorem Q kwantów dla każdej pary węzłów sieci. Po wyznaczeniu zbioru Q wszystkich kwantów w sieci (po obliczeniu kwantu dla każdej pary węzłów w sieci) możliwe jest obliczenie wektora prognozowanych konfliktów W dla zbioru żądań instalacji tras w sieci D przy wykorzystaniu nowej struktury Q . Wektor ten z kolei daje możliwość dokonania doboru tras na podstawie metryki kosztów łącza z wykorzystaniem informacji o przewidywanych konfliktach o przepustowości łącza sieci wzór (3.2).

Algorytm obliczenia wektora prognozowanych konfliktów jest zaprezentowany poniżej i obejmuje pięć głównych kroków. Parametrami wejściowymi algorytmu są macierz sąsiedztwa M , zbiór D wszystkich żądań instalacji tras w sieci oraz obliczony wcześniej zbiór Q wszystkich kwantów w sieci. Jako wynik swojego działania algorytm zwraca wektor prognozowanych konfliktów.

W Kroku 1 algorytm definiuje na podstawie macierzy sąsiedztwa M wektor prognozowanych konfliktów W i ustawia początkowe wartości wszystkich elementów wektora W równe zero.

Krok 2 ma charakter iteracyjny i jest wykonywany sekwencyjnie dla każdego żądania oddzielnie. W wierszu 4 dla każdego żądania d_i algorytm identyfikuje w zbiorze wszystkich kwantów Q odpowiadający mu kwant q_j . W ogólności, kwant q_j dotyczy żądania d_i jeżeli para węzłów $v_s(d_i)$, $v_d(d_i)$, pomiędzy którą ma być zainstalowana trasa, odpowiada parze węzłów $v_1(q_j)$ oraz $v_2(q_j)$ kwantu. Po odnalezieniu odpowiedniego kwantu q_j algorytm zwiększa wartości wektora prognozowanych konfliktów W o odpowiednie wartości wektora $W(q_j)$ przemnożone przez przepustowość wymaganą dla instalacji trasy $\alpha(d_i)$ (wiersze: 5, 6). Przemnożenie wartości elementów wektora $W(q_j)$ przez

wymaganą przepustowość $c(d_i)$ jest uzasadnione tym, że im większa jest wymagana przepustowość dla instalacji trasy, tym większe jest prawdopodobieństwo konfliktu spowodowanego przez taką trasę.

Algorytm obliczenia wektora prognozowanych konfliktów W

Parametry wejściowe:	
M	- Macierz sąsiedztwa
D	- Zbiór żądań instalacji tras w sieci
Q	- Zbiór wszystkich kwantów w sieci
Parametry wyjściowe:	
W	- Wektor prognozowanych konfliktów
	Krok 1:
1	Na podstawie macierzy M zdefiniuj wektor prognozowanych konfliktów W ;
2	Wyzeruj wartości każdego elementu wektora W ;
	Krok 2:
3	Dla każdego żądania d_i w zbiorze D :
4	Znajdź w zbiorze kwantów Q kwant q_j odpowiadający d_i : ($v_3(d_i)=v_1(q_j)$ oraz $v_1(d_i)=v_2(q_j)$ lub $v_3(d_i)=v_2(q_j)$ oraz $v_1(d_i)=v_1(q_j)$)
5	Dla każdego elementu w_k wektora W :
6	Zwiększ wartość elementu: $w_k := w_k + w_k(q_j) c(d_i)$;
	Krok 3:
7	Znajdź wśród wszystkich elementów wektora W element o minimalnej wartości w_{\min} oraz element o maksymalnej wartości w_{\max} ;
	Krok 4:
8	Dla każdego elementu w_k wektora W :
9	Oblicz nową wartość: $w_k := (w_k - w_{\min}) / w_{\max}$;
	Krok 5:
10	Zwróć: W ;
11	Koniec

Wynikiem działania Kroku 2 jest wektor W , którego wartość każdego elementu jest tym większa im większe jest prawdopodobieństwo konfliktu o zasoby łącza któremu odpowiada ten element. Warto zwrócić uwagę, że nie wszystkie kwanty będą miały wpływ na wartości elementów wektora W . Na przykład, jeżeli w zbiorze żądań nie istnieje żądanie instalacji trasy pomiędzy węzłami v_i i v_j to kwant dotyczący pary węzłów v_i i v_j nie będzie wpływał na wartości wektora W . Z powodu nieobecności w zbiorze D żądania instalacji trasy pomiędzy węzłami v_i i v_j taki kwant nie będzie użyty dla inkrementacji wartości wektora W .



W Kroku 3 algorytm wykrywa elementy wektora W o najmniejszej wartości w_{\min} i o maksymalnej wartości w_{\max} . Jest to niezbędne w celu normalizacji wartości wektora W przeprowadzanej w Kroku 4.

W Kroku 4 algorytm modyfikuje wartości wektora W . Ważny jest fakt, że proponowana metryka kosztu (3.2) zawiera dwie składowe: (1) element dotyczący zarezerwowanych zasobów: $(g(e_i)+c(d_i))/b(e_i)$ który przyjmuje wartości z przedziału $\langle 0; 1 \rangle$ oraz (2) czynnik prognozowanych konfliktów: $w(e_i)$, który odpowiada elementowi wektora prognozowanych konfliktów W . Elementy wektora W bez normalizacji przyjmowałyby wartości z przedziału $\langle 0; \infty \rangle$. Powodowałyby to dużą przewagę czynnika prognozowanych konfliktów i prowadziłyby do błędów podczas doboru tras.

W trakcie prowadzonych badań zostały przebadane następujące sposoby normalizacji (wzór (3.7)) i modyfikacji (wzory (3.8), (3.9)) wartości elementów wektora W :

$$w_i = \frac{w'_i}{w_{\max}} \quad (3.7)$$

gdzie:

w'_i – wartość wektora prognozowanych konfliktów przed modyfikacją.

$$w_i = \frac{w'_i - w_{\min}}{w_{\max} - w_{\min}} \quad (3.8)$$

$$w_i = \frac{w'_i - w_{\min}}{w_{\max}} \quad (3.9)$$

Logicznym jest fakt, że przy małych odchyleniach wartości elementów wektora W (kiedy w_{\max} jest bliskie w_{\min}) przydatność prognozowanych konfliktów maleje. W przypadku zwykłej normalizacji (wzór (3.7)) wartości elementów wektora W zbliżają się bowiem do 1, co zwiększa nadmiernie w metryce (3.2) rolę prognozowanych konfliktów nad częścią dotyczącą zarezerwowanych zasobów. W przypadku modyfikacji (wzór (3.8)) elementy wektora W przyjmują wartości z przedziału $[0; 1]$ co powoduje zbyt dużą zmianę wartości metryki kosztu łączącej przy



małej różnicy w prognozowanych konfliktach. Jedynie modyfikacja określona wzorem (3.9) powoduje redukcję wartości elementów wektora W dla małych zmian wartości prognozowanych konfliktów, co z kolei pozwala „automatycznie” zredukować wpływ prognozowanych konfliktów na wartości metryki kosztu łączy sieci (3.2) bez dodatkowego zarządzania współczynnikami dotyczącymi prognozowanych konfliktów a i wykorzystanych zasobów β .

W Kroku 5 algorytm zwraca wektor prognozowanych konfliktów i kończy działanie.

Przy wykorzystaniu sposobu opisanego powyżej, możliwe jest obliczenie wektora prognozowanych konfliktów który zależy od:

- topologii sieci,
- zbioru żądań instalacji tras i przepustowości żądanych dla tych tras.

Pozwala to na skorzystanie z wzoru (3.2) w celu obliczenia wartości metryki kosztu łączy podczas wyznaczania tras. Redukcja współczynnika blokowań żądań instalacji tras z powodu zastosowania proponowanego mechanizmu prognozowania konfliktów została potwierdzona poprzez badania symulacyjne opisane w dalszej części podrozdziału w sekcji 3.1.7.

3.1.5 Dobór proporcji wartości współczynników a oraz β

Jak widać ze wzoru (3.2), proponowana metryka kwantowa bazuje częściowo na aktualnych danych dotyczących dostępności zasobów łączy sieci oraz zapotrzebowań żądań (wyodrębnione poniżej z wzoru (3.2) jako wyrażenie (3.10)) oraz wykorzystuje częściowo rezultaty prognozowania (drugi składnik wzoru (3.2) wyodrębniony poniżej jako wyrażenie (3.11)):

$$c_{\beta}(e_i) = \left(\frac{g(e_i) + c(d_j)}{b(e_i)} \right) \beta \quad (3.10)$$

$$c_{\alpha}(e_i) = w(e_i)\alpha \quad (3.11)$$

Dobór współczynników α i β bezpośrednio wpływa więc na charakterystykę efektywnościową proponowanej metody. Zbyt wysokie wartości współczynnika α powodują ograniczenie przez algorytm znaczenia danych o aktualnie zarezerwowanych zasobach. Z kolei, za wysokie wartości współczynnika β skutkuje ograniczeniem roli mechanizmu prognozowania konfliktów.

Oznaczmy poprzez $r(\rho_B)$ przewagę (pod względem prawdopodobieństwa blokowania żądań) proponowanej metody nad podejściem referencyjnym wykorzystującym metrykę (3.1):

$$r(\rho_B) = \rho_B(\text{Podejście referencyjne}) - \rho_B(\text{Proponowane podejście}) \quad (3.12)$$

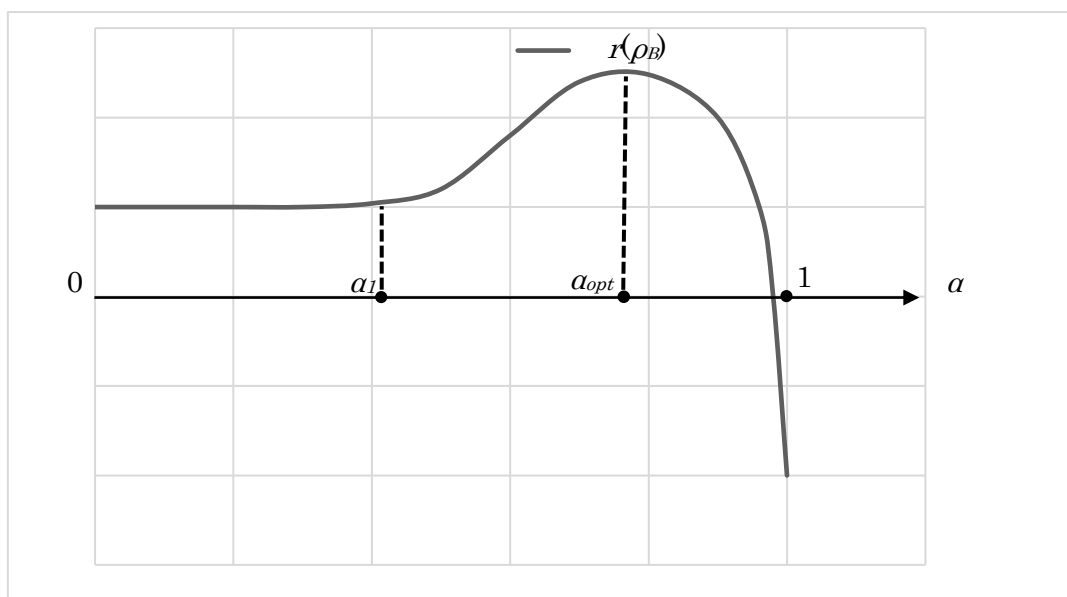
gdzie:

$\rho_B(\text{podejście } X)$ – prawdopodobieństwo blokowania żądań uzyskane dla podejścia X

Rys. 3.4 przedstawia ogólną hipotetyczną zależność wartości $r(\rho_B)$ od ilorazu α/β , która, jak pokazują dalsze części rozprawy, znajduje także poparcie w wynikach badań symulacyjnych.

Mając na uwadze związek pomiędzy α i β (wzór (3.3)) możemy określić relację pomiędzy α i β posługując się wyłącznie wartością współczynnika α . Logicznym jest, że dla $\alpha=0$ i $\beta=1$, mechanizm prognozowania konfliktów nie ma wpływu na wynik obliczenia metryki kwantowej. Wynik działania algorytmu zależy wtedy wyłącznie od składnika formuły dotyczącego aktualnych danych $c_\beta(e_i)$. Z opisanych w sekcji 3.1.2 powodów wzór (3.2) dla $\beta=1$, $\alpha=0$ sam w sobie ma przewagę nad podejściem referencyjnym (3.1). W przypadku $\alpha=0$, wartość $r(\rho_B)$ jest nieujemna.

W początkowym przedziale $\langle 0; \alpha_1 \rangle$ wzrost wartości współczynnika α nie ma wyraźnego wpływu na efektywność algorytmu, ponieważ przy małych wartościach α , wpływ części prognozowanych danych $c_\alpha(e_i)$ na wartość metryki kwantowej nie jest duży. Efekt czynnika prognozowania przy małych wartościach α nie jest wystarczający, aby znacząco zmienić charakterystykę działania algorytmu doboru tras.



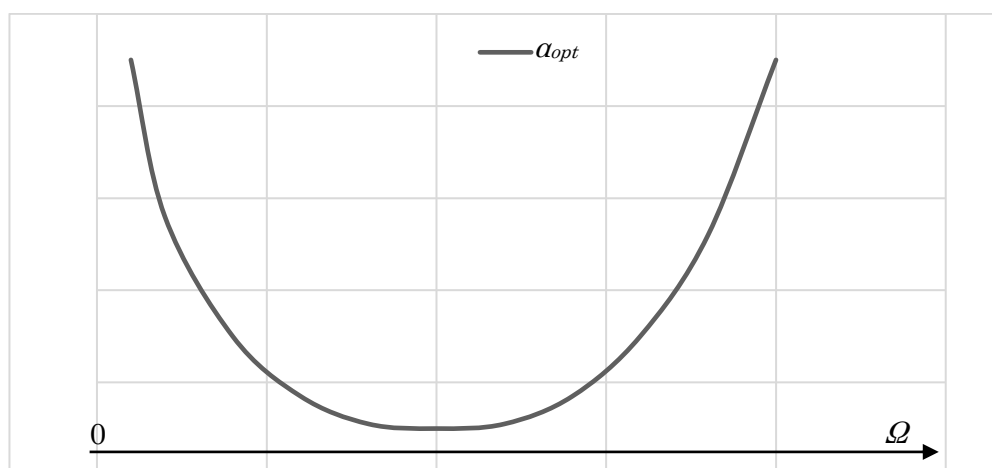
Rysunek 3.4 Hipotetyczny oczekiwany wpływ wartości współczynnika α na przewagę nowej metody nad referencyjnym podejściem

W przypadku wartości α powyżej wartości progowej α_1 , wpływ części prognozowanych danych $c_a(e_i)$ na efektywność działania algorytmu zaczyna wzrastać. Ta tendencja utrzymuje się do określonej optymalnej wartości α_{opt} , stanowiącej optymalny balans pomiędzy aktualną a prognozowaną informacją dla wyznaczania wartości metryki kosztu łącza.

Dalszy wzrost wartości α powoduje redukcję wpływu informacji o zarezerwowanych zasobach na instalację tras w sieci. Ma to z kolei negatywny wpływ na efektywność proponowanej metody. Przy wysokich wartościach α algorytm Dijkstry ma tendencję do niemal całkowitego pomijania informacji o wyczerpaniu zasobów łącza. Skutkiem tego efektywność proponowanej metody przy wysokich wartościach α będzie niższa od efektywności metody referencyjnej.

Wartości α_1 i α_{opt} nie są naturalnie stałe dla dowolnej sieci lecz zależą od obciążenia sieci Ω . Hipotetyczna zależność wartości α_{opt} od obciążenia dowolnej sieci Ω jest prezentowana na Rys 3.5. W przypadku niskich wartości obciążenia Ω , dobór tras ukierunkowany na zrównoważenie obciążenia nie jest za bardzo przydatny. Z tego wynika, że efektywność takich rozwiązań sama w sobie nie jest wysoka dla scenariuszy niskiego obciążenia sieci. Niskie obciążenie sieci wiąże się również z niewielkim prawdopodobieństwem występowania konfliktów związanych z rywalizacją o zasoby łącza. Istotny wpływ części prognozowanych danych $c_a(e_i)$ na

efektywność działania algorytmu doboru tras jest więc oczekiwany dopiero przy wysokich wartościach a (przy sztucznym przesunięciu ilorazu a/b w stronę a). Wzrost \mathcal{Q} powoduje z kolei zwiększenie znaczenia mechanizmu doboru tras ukierunkowanego na zrównoważenie obciążenia sieci (metod typowo bardziej zaawansowanych), a rola czynnika $c_a(e_i)$ zanika. Jednocześnie, wzrost wartości \mathcal{Q} skutkuje przyspieszeniem wyczerpania zasobów przepustowości łącz sieci podczas działania algorytmu doboru tras. Z tego powodu, przy wzroście obciążenia sieci oczekiwane jest, aby waga czynnika dotyczącego aktualnych danych o zarezerwowanych zasobach wzrastała szybciej niż waga informacji o prognozowanych konfliktach. Powyższe dwa efekty sprawiają, że wartość współczynnika a_{opt} maleje w pierwszej (lewej) połowie wykresu na Rys. 3.5.



Rys. 3.5 Hipotetyczny oczekiwany wpływ obciążenia sieci na optymalną wartość współczynnika a

Dalszy wzrost obciążenia powoduje zwiększenie prawdopodobieństwa występowania konfliktów o zasoby przepustowości, co z kolei przyczynia się do wzrostu wagi składnika dotyczącego przewidywanych konfliktów o zasoby. W przypadku stosowania algorytmu Dijkstry do wyznaczania tras, efektywne rozwiązanie problemu konfliktów przy wysokim obciążeniu sieci nie jest możliwe bez mechanizmu przewidywania tych konfliktów. To z kolei podnosi wagę informacji o prognozowanych konfliktach wyznaczonej w wyniku działania algorytmu budowania prognozowanych konfliktów przed uruchomieniem

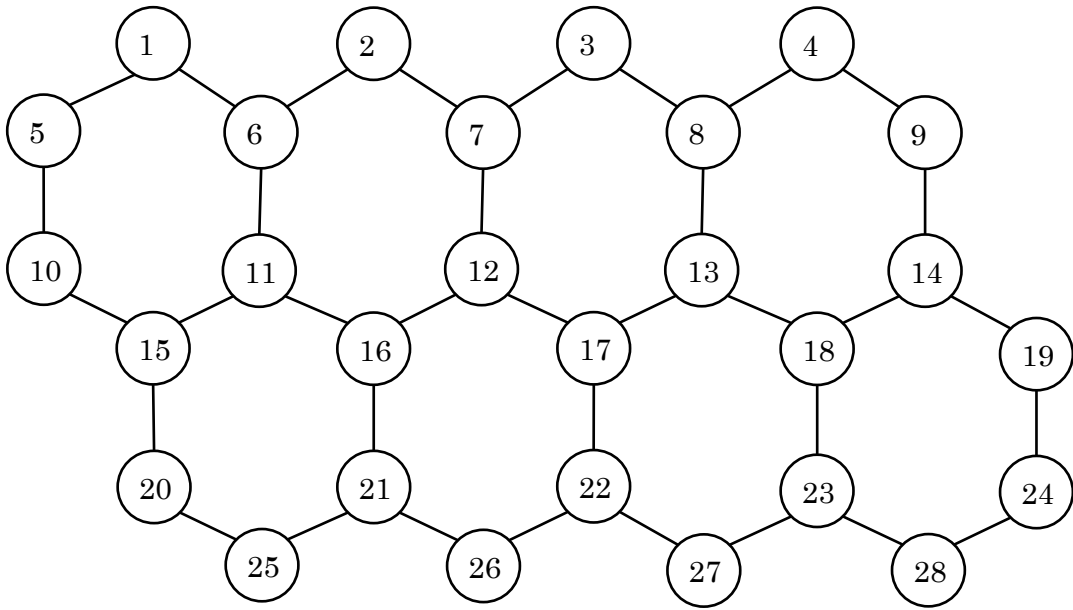
algorytmu doboru tras. Powoduje to wzrost wartości α_{opt} (przesunięcie funkcji α_{opt} w górę) w prawej połowie wykresu na Rys. 3.5.

Z tych samych powodów, przy wyznaczaniu tras dla kolejnych żądań (i przy postępującej alokacji zasobów sieci dla wyznaczonych tras), oczekiwana jest analogiczna reakcja parametrów α_1 na zmianę obciążenia sieci (najpierw spadek, a następnie wzrost).

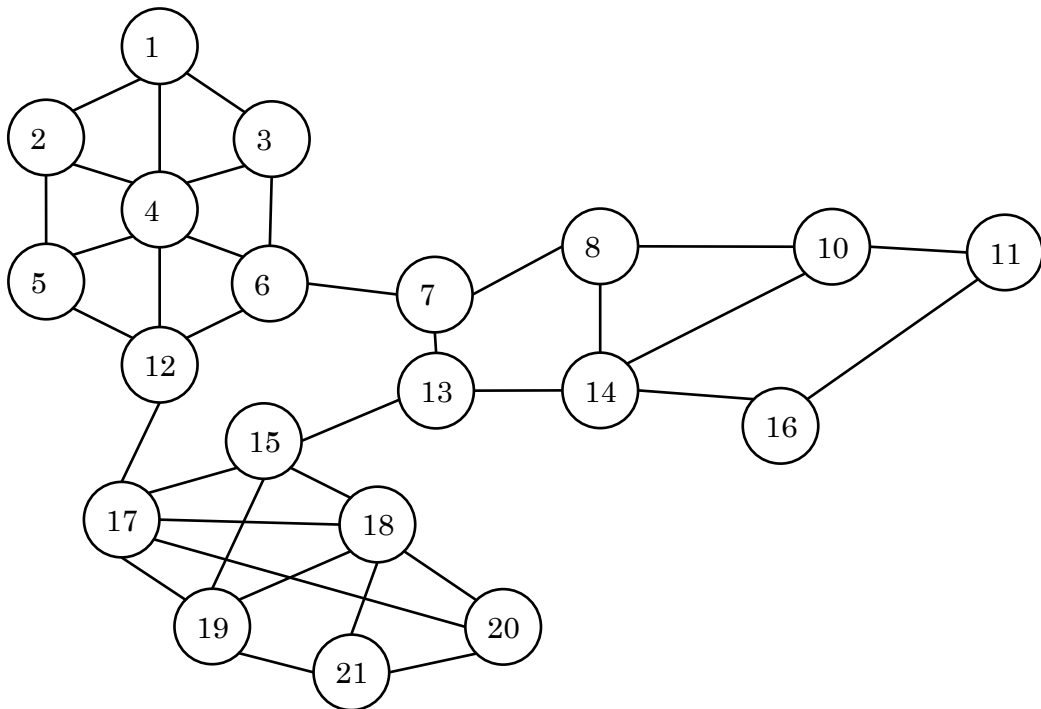
Powyżej opisane hipotetyczne zależności są prawidłowe dla sieci o regularnej topologii, tj. sieci o względnie jednakowych stopniach wszystkich węzłów, jak np. w przypadku topologii zaprezentowanej na Rys. 3.6. Warto zaznaczyć, że w przypadku rzeczywistych sieci rzadko spotykana jest topologia regularna (zazwyczaj topologie sieci teleinformatycznych są nieregularne, co ukazuje np. Rys. 3.7. Nieregularność topologii sprawia z kolei, że każda regularna część sieci ma swoje wartości α_1 i α_{opt} .

Z tego powodu hipotetyczna oczekiwana zależność wartości $r(\rho_B)$ od ilorazu α/β dla rzeczywistych sieci nie jest zawsze zgodna z Rys. 3.4, lecz wynika z sumy charakterystyk poszczególnych regularnych obszarów sieci z różnymi wartościami α_1 i α_{opt} . Skutkiem tego, wykres zaprezentowany na Rys. 3.4 może posiadać w przypadku topologii nieregularnych wiele ekstremów (np. jak na przykładowym Rys. 3.8), które mogą być niewidoczne przy małych obciążeniach sieci, ale ze wzrostem obciążenia ich obecność może być bardziej widoczna zarówno pod względem ich liczby, jak i wartości. Powodem tego jest obecność w niejednorodnych sieciach tak zwanych „wąskich gardeł” – łączy sieciowych których wyczerpanie dostępnych zasobów prowadzi do segmentacji sieci (do izolacji części sieci) – rozpadu sieci na odseparowane części, z których każda ma swoją topologię i w wyniku tego swoje optymalne parametry α_1 i α_{opt} .

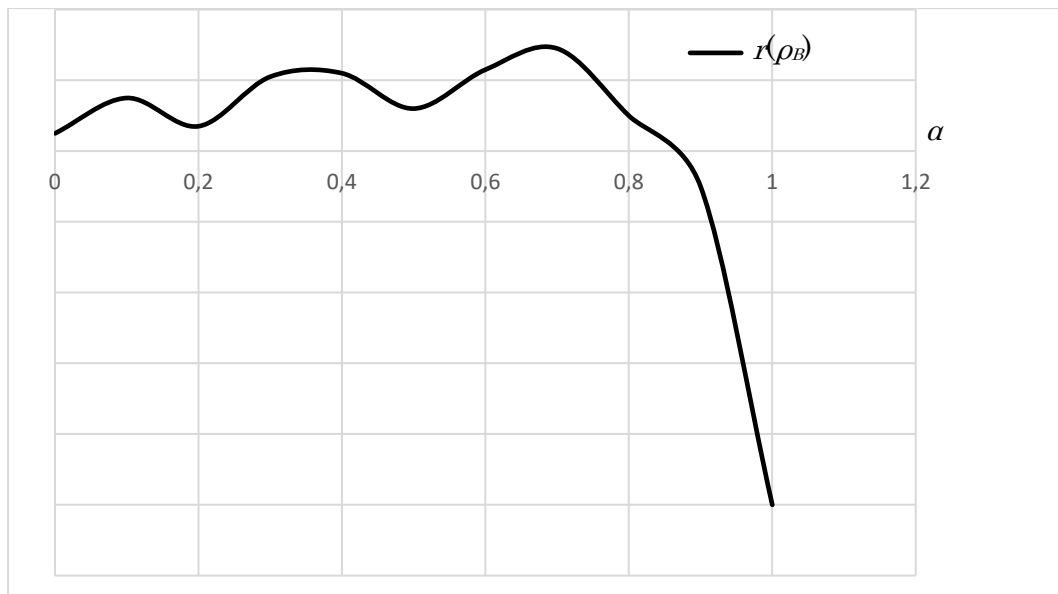
Na Rys. 3.7, wąskimi gardłami są łącza (6, 7), (13, 15), (12, 17).



Rysunek 3.6 Przykład regularnej topologii sieci



Rys. 3.7 Przykład nieregularnej topologii sieci

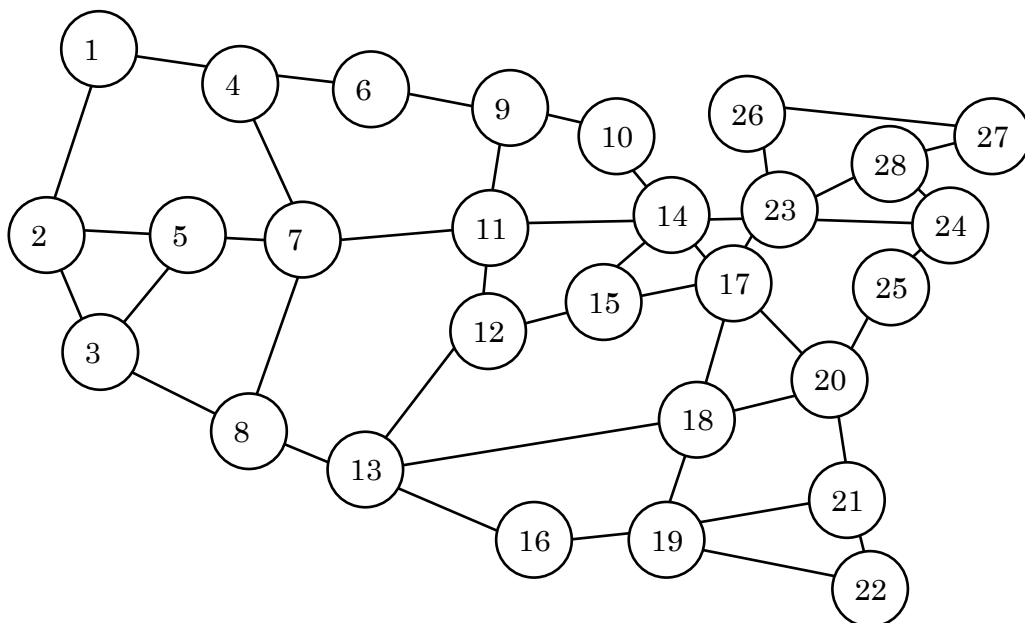


Rys. 3.8 Hipotetyczny oczekiwany wpływ wartości współczynnika α na przewagę nowej metody nad referencyjnym podejściem (dla sieci o topologii nieregularnej)

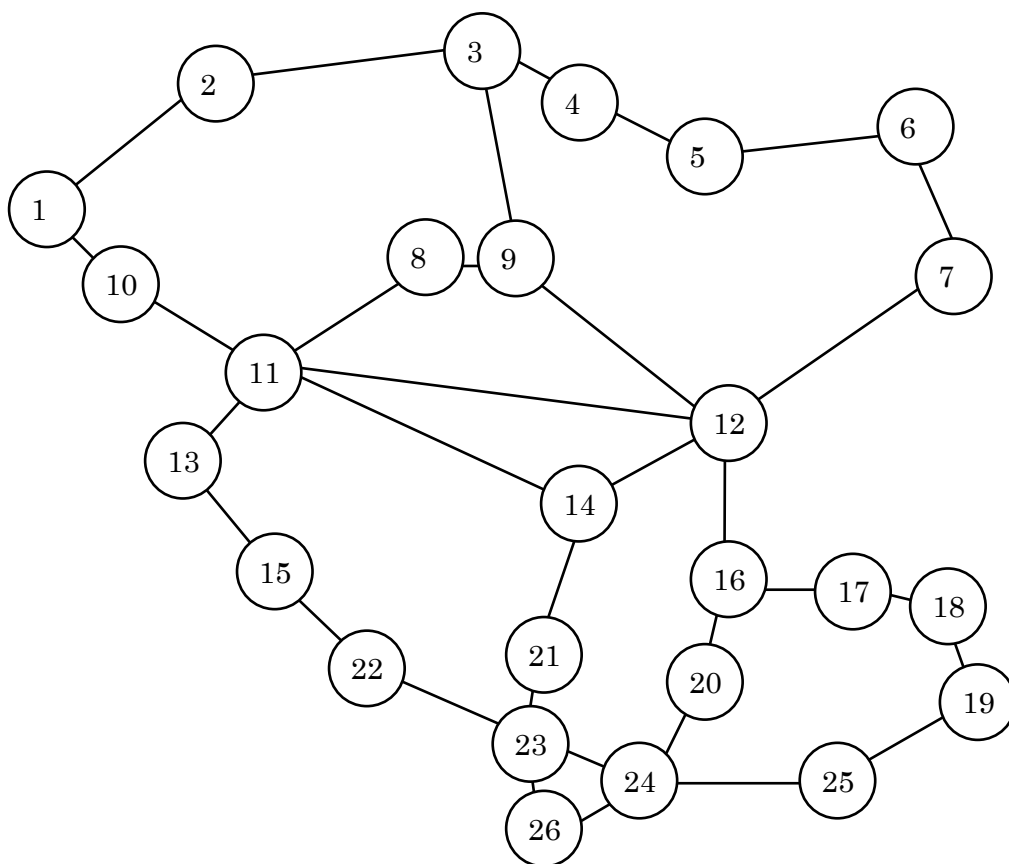
3.1.6 Założenia symulacyjne

W celu analizy wpływu współczynnika α/β na właściwości rozwiązania, zostały w niniejszej sekcji ukazane wyniki badań przeprowadzonych dla trzech sieci:

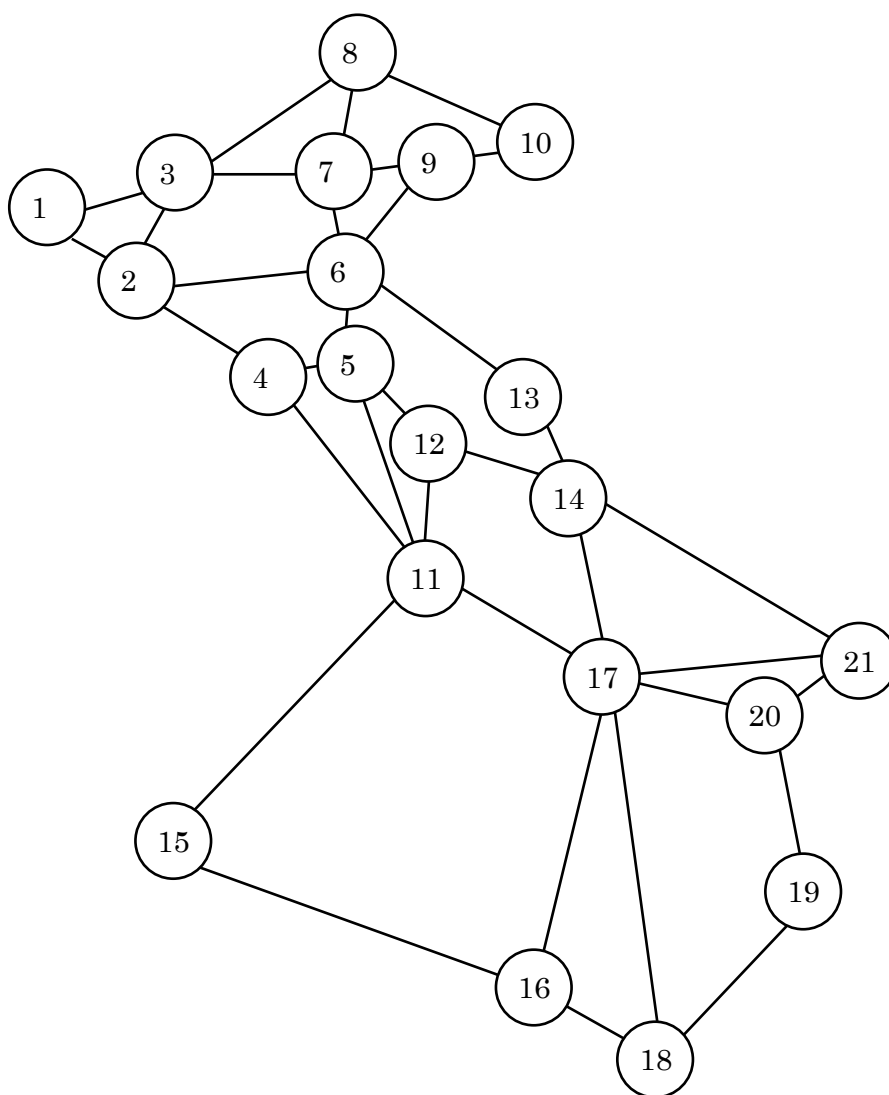
- Sieci szkieletowej 1 o topologii zaprezentowanej na Rys. 3.9 [57] (w dalszej pracy zwanej „siecią USA”).
- Polskiej sieci szkieletowej Pionier, której topologia jest zaprezentowana na Rys. 3.10 [57] (w dalszej pracy zwanej „siecią Pionier”).
- Sieci szkieletowej 2 o topologii ukazanej na Rys. 3.10 [57] (w dalszej pracy zwanej „siecią Włoską”).



Rysunek 3.9 Topologia sieci Stanów Zjednoczonych



Rys. 3.10 Topologia sieci Pioneer



Rys. 3.11 Topologia sieci Włoskiej

Powodem wyboru sieci Pionier jest z kolei jej kształt charakterystyczny dla wielu sieci powstałych w wyniku ewolucji sieci o topologii pierścieniowej (będących połączeniem kilku pierścieni). Powyższe trzy topologie są rozpatrywane także we wszystkich badaniach symulacyjnych prezentowanych w dalszej części niniejszej rozprawy.

Powyższe trzy topologie nie zostały wybrane w sposób przypadkowy. Sieć USA i sieć Włoska są często wykorzystywane w analogicznych badaniach w literaturze. Sieć USA charakteryzuje się ponadto wysoką regularnością topologii. Odwrotnie, sieć Włoska ma nieregularną topologię.

Dla analizy współczynnika α/β zostały wykorzystane cztery modelu ruchu sieciowego prezentowane w Tabeli 3.6 skutkujące różnymi poziomami obciążenia sieci.

Model	Przedziały przepustowości łączy wymaganej dla obsługi żądań $c(d)$ wyrażone liczbą kanałów łączy	Rozkład prawdopodobieństwa wymaganej przepustowości na instalacji trasy
A0_4	[1;4] kanały	Równomierny
A0_5	[1;5] kanały	Równomierny
A0_6	[1;6] kanały	Równomierny
A0_12	[1;12] kanały	Równomierny

Tabela 3.6 Modele ruchu sieciowego

Powyższe modele wykorzystane zostały w celu generacji zbiorów żądań (każdy zbiór zawierał wszystkie możliwe pary węzłów sieciowych), każde i -te żądanie o wymaganej przepustowości $c(d_i)$. Warto zwrócić uwagę, że model A0_12 został dodatkowo przebadany pod kątem reakcji sieci na zbyt wysokie obciążenie ponad granice możliwości sieci wynikające z przepustowości jej łączy.

W ramach badań wstępnych nad doбором wartości α/β zostało przeprowadzonych 200 (losowo wygenerowanych zbiorów żądań na instalację trasy zgodnie z odpowiedniemu modelu ruchu sieciowego) symulacji dla każdego z 12 przypadków (trzy topologie sieci, cztery modele ruchu). Założona liczba dostępnych kanałów na każdym łączy każdej sieci wynosiła 80. Wyniki uzyskane podczas symulacji (Tabela 3.7) potwierdzają wnioski o wpływie ilorazu α/β na efektywność metody oraz na zależność optymalnych wartości współczynnika α od stopnia obciążenia sieci.

Zaobserwowane wyniki badań potwierdzają, że:

1. Wartość α_{opt} nie jest stała dla wszystkich sieci i jest różna dla różnych topologii sieci.
2. Wartość α_{opt} jest także różna dla różnych modeli ruchu sieciowego.

Optymalna wartość α_{opt} powinna więc być dobierana indywidualnie dla każdej topologii sieci i modelu ruchu sieciowego.



Proponowana w niniejszej pracy metoda doboru ilorazu a/b (optymalnej wartości a_{opt}) bazuje na dwuetapowej symulacji. W tym celu niezbędne jest także wygenerowane na początku N_D zbiorów zadań dla danej topologii sieci i określenie modelu ruchu sieciowego (np. na podstawie analizy statystyki ruchu sieciowego).

Pierwszy etap symulacji dotyczy doboru tras przy wykorzystaniu proponowanego w niniejszym rozdziale metryku kosztu łącza (3.2) dla każdego wygenerowanego zbioru zadań i dla każdej wartości a_i w przedziale $[0; 1]$ z krokiem dyskretyzacji $a_{\Delta 1}$ oraz obliczenia średniej wartości prawdopodobieństwa blokowania zadań (oraz przedziału ufności przy poziomie ufności 0.95) dla każdej a_i . Następnie ma miejsce obliczanie średniej wartości prawdopodobieństwa blokowania zadań (oraz przedziału ufności przy poziomie ufności 0.95) dla każdej a_i przy wykorzystaniu metryki kosztu (3.1). Kolejnym krokiem etapu 1 jest obliczenie przewagi proponowanej metody nad metodą referencyjną względem prawdopodobieństwa blokowania zadań $r(\rho_B)$ dla każdej a_i . Finalnie, w oparciu o zależność $r(\rho_B)$ od a , wybierany jest przedział $[a_1; a_2]$, w którym wartość $r(\rho_B)$ jest najwyższa.

Drugi etap symulacji polega na doborze tras przy zastosowaniu proponowanego w niniejszym rozdziale metryku kosztu łącza (3.2) dla każdego wygenerowanego zbioru zadań i dla każdej wartości a_i w przedziale $[a_1; a_2]$ z krokiem dyskretyzacji $a_{\Delta 2}$, gdzie $a_{\Delta 2} \ll a_{\Delta 1}$. Kolejno ma miejsce obliczenie średniej wartości prawdopodobieństwa blokowania zadań (oraz wyznaczenie przedziału ufności) dla każdej wartości a_i korzystając z metryku kosztu (3.1) oraz obliczenie przewagi proponowanej metody nad metodą referencyjną względem prawdopodobieństwa blokowania zadań $r(\rho_B)$ dla każdej a_i . Finalnie, wybierana jest ta wartość a_{opt} , dla której wartość $r(\rho_B)$ jest maksymalna.

3.1.7 Wyniki symulacji

Przydatność zaprezentowanej w niniejszym podrozdziale metody doboru tras została poddana analizie podczas badań symulacyjnych przeprowadzonych dla trzech topologii sieci opisanych w sekcji 3.1.6:

- sieci USA (Rys. 3.9);
- sieci Pionier (Rys. 3.10);
- sieci Włoskiej (Rys. 3.11).

Dla każdej z powyższych topologii sieci, liczba dostępnych kanałów na każdym łączu sieci wynosiła 80. Dla każdej topologii sieci zostały przebadane cztery modele ruchu sieciowego zaprezentowane w Tabeli 3.6.

Dobór ilorazu α/β został przeprowadzony na podstawie 200 zbiorów żądań ($N_D = 200$) wygenerowanych dla każdej pary (Topologia Sieci, Model Ruchu Sieciowego). Krok dyskretyzacji $\alpha_{\Delta 1}$ dla pierwszego etapu symulacji został wybrany jako $\alpha_{\Delta 1} = 0.05$, a dla drugiego etapu symulacji jako $\alpha_{\Delta 2} = 0.005$. Wykresy uzyskane w trakcie drugiego etapu symulacji po określeniu ilorazu α/β są zaprezentowane w Aneksie 4.

Po dokonaniu wyboru ilorazu α/β dla każdej pary (Topologia Sieci, Model Ruchu Sieciowego), został przeprowadzony 1000 symulacji mających na celu uzyskanie precyzyjnej informacji o przewadze proponowanej metody nad podejściem referencyjnym.

Analizie podlegały następujące parametry dla każdej z metod:

- prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$,
- długość przedziału ufności dla $\overline{\rho_B}$ przy poziomie ufności 0.95 ($I_{0.95}[\overline{\rho_B}]$),
- średni czas potrzebny na wyznaczenie tras $\overline{t_r(P)}$ dla wszystkich żądań w zbiorze,
- długość przedziału ufności dla $\overline{t_r(P)}$ przy poziomie ufności 0.95 ($I_{0.95}[\overline{t_r(P)}]$),
- średnia wartość współczynnika wykorzystania zasobów sieci \overline{U} (obliczona zgodnie ze wzorem (3.13)),
- długość przedziału ufności dla \overline{U} przy poziomie ufności 0.95 ($I_{0.95}[\overline{U}]$),
- średnia liczba przeskoków w trasach $\overline{|E(P)|}$,
- długość przedziału ufności o poziomie ufności 0.95 ($I_{0.95}[\overline{|E(P)|}]$).

$$U = \frac{\sum_{i=1}^{|E|} c(e_i)}{\sum_{i=1}^{|E|} b(e_i)} \quad (3.13)$$

Szczegółowe wyniki symulacji dla sieci są zaprezentowane w Aneksie 5. Znormalizowane wartości przewagi proponowanego podejścia nad podejściem referencyjnym są zaprezentowane w Tabeli 3.7.

W każdym przypadku (dla każdej topologii sieci i dla każdego modelu ruchu sieciowego) proponowane podejście ma przewagę nad podejściem referencyjnym (nawet o 53.6% dla sieci USA i modelu ruchu A0_4) pod względem redukcji prawdopodobieństwa blokowania żądań.

Sieć	$\overline{\rho_B}$ (%)	$\overline{t_r(P)}$ (%)	\overline{U} (%)	$\overline{ E(P) }$ (%)
Model A0_4				
USA	53.6	-17.5	5.6	6.0
Włoska	21.0	-16.1	3.0	3.0
Pionier	27.0	-20.5	1.7	4.0
Model A0_5				
USA	18.3	-16.4	4.8	6.0
Włoska	7.2	-17.6	2.3	1.9
Pionier	1.9	-23.8	1.9	3.0
Model A0_6				
USA	14.6	-19.4	0.8	3.8
Włoska	4.6	-22.2	2.0	1.5
Pionier	0.2	-16.3	6.0	6.3
Model A0_12				
USA	17.2	-34.9	-32.8	6.7
Włoska	4.0	-18.6	-1.2	3.1
Pionier	0.4	-28.7	11.7	13.5

Tabela 3.7 Znormalizowane wartości przewagi proponowanego podejścia nad podejściem referencyjnym

Jak jest widoczne w Tabeli 3.7, proponowana metoda wyznacza trasy krótsze niż trasy, które zostały wykryte przez metodę referencyjną. W najlepszym przypadku (sieć Pionier i model ruchu A0_12), średnia długość wykrytych poprzez nową metodę ścieżek jest o 13% mniejsza w porównaniu do ścieżek wykrytych za pomocą metryki (3.1). W najgorszym przypadku (sieć Włoska i model ruchu A0_6)

proponowana metoda ma przewagę 1.5%. Redukcja długości tras powoduje również przewagę proponowanej metody pod względem wykorzystanych zasobów przepustowości sieci. Oprócz dwóch przypadków opisanych poniżej, proponowana metoda wymaga rezerwacji mniejszej ilości zasobów niż metoda referencyjna.

Warto zwrócić uwagę, że ujemna wartość dla znormalizowanej przewagi we współczynniku wykorzystania zasobów dla przypadków sieci USA i Włoskiej dla modelu A0_12 nie ukazuje przewagi podejścia referencyjnego. Odwrotnie, proponowana metoda ma lepsze wyniki również w tych dwóch przypadkach. Większe wartości współczynnika wykorzystania zasobów dla proponowanej metody są spowodowane tym, że proponowana metoda umożliwia zainstalowanie większej liczby tras (wynika to z rezultatów dotyczących prawdopodobieństwa blokowania żądań).

Jedyną wadą proponowanej metody jest czas obliczeń. Przewagi w odniesieniu do redukcji prawdopodobieństwa blokowania żądań, współczynnika wykorzystanych zasobów i długości tras są bowiem osiągane przy wzroście czasu obliczeń.

Najlepsze wyniki proponowana metoda uzyskuje dla sieci jednorodnych o niewysokim obciążeniu. Najmniejsze zyski notuje z kolei dla sieci opartych o struktury pierścieniowe. Jednakże ważnym jest fakt, że we wszystkich przypadkach nowa metoda demonstruje wyniki lepsze od metody referencyjnej.

3.1.8 Wnioski

Widoczna jest analogia proponowanej metody z prezentowanym we wprowadzaniu do niniejszej pracy przykładem zastosowania transformacji Fouriera do obliczenia wyjściowego sygnału. Widmo amplitudowe sygnału które reprezentuje sygnał w postaci rozkładu natężeń wszystkich możliwych składowych harmonicznych o różnej wartości podobne jest do zbioru żądań, który prezentuje model ruchu sieci w postaci rozkładu wszystkich możliwych par w sieci o różnej wymaganej przepustowości. Widmo amplitudowe systemu, która jest miarą odpowiedzi systemu na sygnał o niedorodnej częstotliwości jest podobna do wektora prognozowanych konfliktów, który możemy traktować jako miarą odpowiedzi sieci na żądania instalacji tras. W celu uzyskania sygnału wyjściowego są wykorzystane operacje systemu nad widmem amplitudowym sygnału wejściowego i widmem

amplitudowym systemu. Analogicznie, w proponowanej metodzie są wykorzystane operacje nad zbiorem żądań i wektorem prognozowanych konfliktów.

Teoretyczne wyniki analizy zostały udokumentowane zgodnymi wynikami symulacji, które wskazują na przydatność metod transformacyjnych do rozwiązania zadań doboru tras ukierunkowanego na redukcję współczynnika blokowania żądań.

Wyniki badań symulacyjnych wskazują na przewagę metryki kosztu łącza zaprojektowanej za pomocą metod transformacyjnych nad podejściem referencyjnym dla wszystkich badanych scenariuszy i topologii sieci. W najgorszym przypadku przewaga ta jest równa 0.2%, a w najlepszym osiąga ona 53.6%. Średnia arytmetyczna przewaga wynosi 14.2%.

Mimo tego, że zaprezentowana metoda zaprojektowana została dla doboru pojedynczych tras (tj. bez stowarzyszonych rozłącznych tras zabezpieczających), niniejsze badania zapoczątkowały dalsze prace nad opracowaniem metody redukcji współczynnika blokowania żądań w przypadku instalacji par rozłącznych tras w sieci, których wyniki zostały opisane w Aneksie 9.

3.2 Metoda redukcji czasu doboru tras ukierunkowana na zrównoważenie obciążenia w sieciach rozległych

Zaprezentowana w poprzednim podrozdziale metoda doboru tras niezawodnych ma istotną wadę dotyczącą czasu obliczenia zbioru wszystkich kwantów. Wyznaczenie zbioru wszystkich kwantów nie stanowi problemu w przypadku stabilnych sieci o topologiach niezmiennych w długim czasie. Jednakże, po awarii elementu sieci przydatność wektora prognozowanych konfliktów spada. Powodem tego jest fakt, że wektor prognozowanych konfliktów W został obliczony zawczasu na podstawie kwantów sieci dla topologii sieci, w której wszystkie jej elementy były sprawne. Analogicznie, kwanty zostały wyznaczone na podstawie zbioru wszystkich tras także dla topologii sieci sprzed momentu wystąpienia awarii. Po awarii wiele ze zidentyfikowanych uprzednio tras jest niedostępnych, a mimo to prawdopodobieństwa wykorzystania łączy są obliczone mając na uwadze możliwość instalacji tras poprzez te uszkodzone w wyniku awarii elementy. To z kolei powoduje wykorzystanie nieaktualnych po awarii wartości dla budowania kwantów, co prowadzi do błędów w wektorze prognozowanych konfliktów.

Obliczenie nowego wektora prognozowanych konfliktów po wystąpieniu awarii wymaga dodatkowego czasu i nie pozwala na wyznaczenie nowych tras w rozsądnym czasie.

Niniejszy podrozdział pracy przedstawia rezultaty badań ukierunkowanych na zmodyfikowanie zaprezentowanego w podrozdziale 3.1 podejścia, aby uzyskać istotne ograniczenie czasu obliczeń tras. Ma to na celu m.in. umożliwienie jego zastosowania do szybkiego wyznaczenia tras w sytuacji po wystąpieniu awarii. W sekcji 3.2.1 zostały opisane badania ukierunkowane na osiągnięcie tych celów. W sekcjach 3.2.2-3.2.4 została opisana metoda redukcji czasu obliczeń wektora prognozowanych konfliktów oryginalnie opublikowana w artykule [58]. W szczególności, w sekcji 3.2.2 przedstawione są podstawy autorskiej koncepcji szybkiego wyznaczania wektora prognozowanych konfliktów. Nowa metoda prognozowania konfliktów i nowa metryka kosztu łącza zostały przedstawione w sekcji 3.2.3, a dobór optymalnych wartości parametrów dla tego algorytmu jest omówiony w sekcji 3.2.3. Wyniki symulacji zostały opisane w sekcji 3.2.4. Podsumowanie zmodyfikowanej koncepcji jest zawarte w podrozdziale 3.3.

3.2.1 Analiza możliwości modyfikacji algorytmu budowania kwantów ukierunkowanej na redukcję czasu obliczeń

Pierwszy analizowany kierunek badań nad możliwymi modyfikacjami sposobu obliczenia kwantów dotyczył uwzględnienia podczas wyznaczania prawdopodobieństwa wykorzystania łącza także możliwości wystąpienia awarii. Innymi słowy, zbiór wszystkich kwantów Q powinien być wynikiem operacji nie nad jednym modelem (modelem obliczenia prawdopodobieństwa wykorzystania łącza e_i podczas instalacji trasy), a nad dwoma modelami (modelem obliczenia prawdopodobieństwa wykorzystania łącza e_i podczas instalacji trasy i modelem obliczenia wpływu awarii elementów sieci na wykorzystanie łącza e_i). Inne podobne rozwiązanie mogłoby uwzględniać niezerowe prawdopodobieństwo awarii elementów podczas konstrukcji wektora prognozowanych konfliktów. Hipotetycznie, oba kierunki badań są możliwe do zrealizowania poprzez wykorzystanie zaawansowanych metod probabilistycznych i opracowanie nowych. Jednakże, jak ukazuje dalsza część niniejszego podrozdziału, powyższe cele są trudne do osiągnięcia. W praktyce nie udało się uzyskać żadnych wyników umożliwiających osiągnięcie powyżej opisanych celów (powody tego są opisane

poniżej). Co więcej, poniżej wykazana została niemożność uzyskania zadawalających wyników.

Zbiór wszystkich kwantów Q jest unikalny dla danej sieci, której topologia jest reprezentowana poprzez graf G . Możemy traktować zbiór kwantów Q jako rezultat zwracany dla grafu G przez hipotetyczną funkcję $f(G)$:

$$Q = f(G) \quad (3.14)$$

Z kolei, element/elementy sieci, które uległy awarii możemy określić poprzez graf $G_a = (V_a, E_a)$, w którym zbiór V_a reprezentuje wszystkie węzły sieci, które uległy awarii, podczas gdy zbiór E_a reprezentuje wszystkie łącza sieci, które uległy awarii. Po wystąpieniu awarii mamy więc do czynienia z zastąpieniem pierwotnej topologii sieci opisaną grafem G przez strukturę G' określoną w następujący sposób:

$$G' = G \setminus G_a \quad (3.15)$$

Analogicznie, zbiór wszystkich kwantów Q' dla sieci po awarii jej elementów możemy wyznaczyć jako:

$$Q' = f(G \setminus G_a) \quad (3.16)$$

W celu zagwarantowania możliwości obsłużenia wszystkich zadań, których węzły źródłowe/docelowe nie uległy awarii, zbiór wszystkich kwantów dla topologii sieci po awarii powinien być efektywny w stopniu podobnym jak dla pierwotnej topologii sieci. Innymi słowy, różnica w zawartości zbiorów Q i Q' wyznaczonych dla sieci pierwotnej i dla sieci po awarii powinna być minimalna i w najlepszym przypadku powinna być bliska 0.

$$\Delta Q = [Q - Q'] \rightarrow 0 \quad (3.17)$$

Uwzględniając formuły (3.14) i (3.16), możemy zapisać wyrażenie (3.17) w następującej postaci:

$$\Delta Q = [f(G) - f(G \setminus G_a)] \rightarrow 0; \forall G_a \neq \emptyset \quad (3.18)$$

W celu spełnienia założenia opisanego wzorem (3.18), należy dobrać taką funkcję $f(G)$, której wynik nie zależy od grafu topologii sieci G . To z kolei ogranicza możliwość wykorzystania wektorów prognozowanych konfliktów, ponieważ wektor prognozowanych konfliktów jest budowany na podstawie zbioru wszystkich kwantów, które odwzorowują specyfikę i unikalność rozkładu prawdopodobieństwa występowania konfliktów dotyczących przepustowości łączy charakterystycznego dla sieci o danej topologii reprezentowanej przez graf G . Dlatego też powyższy kierunek jest skazany na niepowodzenie. Jak wynika ze (3.18), rozważania w tym zakresie nie są możliwe do przeprowadzenia bez uprzedniego wyeliminowania korelacji pomiędzy zbiorem wszystkich kwantów a topologią sieci. Co z kolei, redukuje przydatność zbioru wszystkich kwantów i efektywność tych metod. Analogicznie nieskuteczne byłoby przeprowadzenie badań ukierunkowanych na wyznaczenie zbiorów kwantów dla wszystkich możliwych topologii sieci G' (dotyczących wszystkich możliwych scenariuszy awarii) pochodzących od pierwotnej sieci G , a nie jedynie pojedynczego zbioru kwantów wyłącznie dla pierwotnej sieci G .

Opisane powyżej strategie zostały poddane w niniejszej sekcji aby ukazać brak perspektyw prowadzenia badań w związanych z nimi kierunkach. Jednocześnie, w dalszej części podrozdziału przedyskutowane zostaną kierunki badań warte uwagi.

Perspektywicznym wydaje się natomiast podejście do budowania kwantów nie na podstawie wszystkich możliwych tras pomiędzy każdą parą węzłów sieci, lecz na podstawie wszystkich możliwych zbiorów k -rozłącznych tras pomiędzy każdą parą węzłów sieci. Podstawy hybrydowej metody opartej na algorytmie kwantowym i na algorytmie doboru par rozłącznych tras (algorytm jest prezentowany w rozdziale 4) zostały opisane w Aneksie 9.

Kolejny perspektywiczny kierunek badań dotyczy opracowania metod redukcji czasu obliczeń kwantów. W ramach badań udało się opracować efektywną metodę redukcji czasu obliczenia kwantów. W dalszej części niniejszego podrozdziału zostały opisane szczegóły nowego podejścia budowania kwantów na podstawie ograniczonego zbioru k najkrótszych tras zamiast budowania kwantów ze zbioru wszystkich tras.

3.2.2 Podstawy metody kwantowego doboru tras opartego na budowaniu kwantów ze zbioru k najkrótszych tras

Złożoność metody kwantowego doboru tras istotnie ogranicza jej zastosowanie w przypadku sieci o topologii zmiennej w krótkim czasie, jak i w przypadku konieczności szybkiego wyznaczenia tras po awarii elementów sieci. W niniejszej części pracy przedstawiona jest udoskonalona wersja metody kwantowego doboru tras ukierunkowana na redukcję czasu obliczeń. W szczególności, udoskonalony został sposób wyznaczenia prawdopodobieństwa wykorzystania zasobów łącza na potrzeby instalacji trasy.

Załóżmy ogólny przypadek istnienia hipotetycznej sieci o topologii G . Centrum Zarządzania Ruchem Sieciowym (CZRS) otrzymało zbiór żądań D instalacji tras w tej sieci. Załóżmy, że CZRS wyznaczył zestaw tras $P(D)^*$ optymalny pod względem współczynnika blokowania żądań. Niech q jest kwantem, który został określony na podstawie zbioru wszystkich tras $P(q)$ pomiędzy węzłami $v_1(q)$ i $v_2(q)$. Zbiór wszystkich tras zawiera trasy o różnej liczbie przeskoków (3.19).

$$c(P_i) \neq \text{const}, i = 1, 2, \dots, |P(q)|; P_i \in P(q) \quad (3.19)$$

Niech $p(P_i \in P(D)^*)$ oznacza prawdopodobieństwo tego, że trasa P_i ze zbioru wszystkich tras $P(q)$ będzie wykorzystana przez CZRS dla ustanowienia połączenia pomiędzy węzłami $v_1(q)$ a $v_2(q)$. Mówiąc inaczej, $p(P_i \in P(D)^*)$ jest prawdopodobieństwem tego, że trasa P_i należy do zbioru tras $P(D)^*$ optymalnego pod względem współczynnika blokowania żądań. Jak wynika z sekcji 3.1.2 i w szczególności ze wzoru (3.4), prawdopodobieństwo $p(P_i \in P(D)^*)$ jest stałe dla wszystkich tras w zbiorze $P(q)$. Podstawowym wyróżnikiem nowej metody jest założenie, że wartość ta nie jest stała dla wszystkich tras w $P(q)$ i zależy od liczby przeskoków w każdej trasie:

$$\begin{cases} p(P_i \in P(D)^*) = p(P_j \in P(D)^*) \leftrightarrow c(P_i) = c(P_j); P_i, P_j \in P(q) \\ p(P_i \in P(D)^*) < p(P_j \in P(D)^*) \leftrightarrow c(P_i) < c(P_j); P_i, P_j \in P(q) \end{cases} \quad (3.20)$$

Powyższe podejście jest lepsze od opisanego w podrozdziale 3.1, gdyż logiczne jest, że prawdopodobieństwo tego, że trasa o jednym przeskoku zostanie

wykorzystana do realizacji żądania jest większe niż prawdopodobieństwo tego, że wykorzystana zostanie trasa posiadająca 30 przeskoków. Z tego wynika, że prawdopodobieństwo zdarzenia «trasa P_i jest optymalna» $p(P_i \in P(D)^*)$ zależy od stosunku liczby przeskoków trasy $c(P_i)$ do liczby przeskoków we wszystkich pozostałych trasach zbioru tras $P(q)$.

Z uwagi na przeznaczenie projektowanej metody dla sieci rzeczywistych, istotny jest aspekt czasu obliczeń. Z tego powodu, w celu redukcji czasu obliczeń w niniejszej pracy proponowany jest uproszczony sposób określenia wartości prawdopodobieństwa $p(P_i \in P(D)^*)$. Stosunek prawdopodobieństw tego, że trasa P_i jest optymalna do stosunku tego, że trasa P_j jest optymalna jest określony jako:

$$\frac{p(P_i \in P(D)^*)}{p(P_j \in P(D)^*)} = \left(\frac{c(P_j)}{c(P_i)} \right)^\gamma \quad (3.21)$$

gdzie:

γ – współczynnik krzywizny funkcji odwrotnej proporcjonalności (mechanizm doboru jest opisany w sekcji 3.2.3).

Pamiętając o tym, że suma prawdopodobieństw dla wszystkich tras w zbiorze $P(q)$ ma być równa 1 (wzór (3.22)), możemy przekształcić wzór (3.21) do postaci (3.23) w celu wyznaczenia prawdopodobieństwa tego, że trasa P_i jest optymalna.

$$\sum_{i=1}^{|P(q)|} p(P_i \in P(D)^*) = 1 \quad (3.22)$$

$$p(P_i \in P(D)^*) = \frac{(c(P_1)/c(P_i))^\gamma}{\sum_{j=1}^{|P(q)|} (c(P_1)/c(P_j))^\gamma} \quad (3.23)$$

gdzie:

P_1 – najkrótsza trasa we zbiorze wszystkich tras.

Następną istotną kwestią jest założenie, iż prawdopodobieństwo, że trasa jest optymalna maleje wraz ze wzrostem liczby jej przeskoków. Z tego powodu, w trakcie obliczenia kwantu można nie brać pod uwagę tras o dużej liczbie przeskoków. Pozwala to stosować w obliczeniach k najkrótszych tras, a nie wszystkich tras pomiędzy daną parą węzłów. Mechanizm doboru liczby k jest zaprezentowany w sekcji 3.2.3. W ramach niniejszej pracy nie jest wzięty pod uwagę problem wyznaczania k najkrótszych tras (opublikowanych i dostępnych jest wiele algorytmów rozwiązania tego problemu, na przykład w pracach [59]-[61]). Przegląd i analiza algorytmów wyznaczania tras jest poza zakresem niniejszej pracy.

Dalsza część niniejszego podrozdziału wyraźnie ukazuje, że przekształcenie zadania wykrycia wszystkich tras do postaci zadania wykrycia jedynie k najkrótszych tras pozwala istotnie przyspieszyć działanie algorytmu budowania kwantów.

Mając obliczone dla każdej z k najkrótszych tras pomiędzy węzłami $v_1(q_j)$ a $v_2(q_j)$ prawdopodobieństwo tego, że ta trasa jest optymalną (3.23), możemy policzyć $w_i(q_j)$, tj. prawdopodobieństwo zastosowania łącza e_i w trasie pomiędzy węzłami $v_1(q_j)$ a $v_2(q_j)$:

$$w_i(q_j) = \sum_{x=1}^k p(P_x \in P(D)^*); \forall P_x(d_j) | e_i \in P_x \quad (3.24)$$

Mając obliczony wektor $W(q_j)$ dla każdego kwantu q_j , możemy obliczyć wartości wektora prognozowanych konfliktów W jako:

$$w(e_i) = \sum_{j=1}^{|D|} w_i(q_j) c(d_j); \quad (3.25)$$

gdzie:

$w_i(q_j)$ – prawdopodobieństwo wykorzystania łącza e_i przez kwant q_j dla instalacji trasy

Następną cechą odróżniającą niniejszą metodę od opisanej w poprzednim rozdziale jest założenie, że wektor prognozowanych konfliktów jest zmienny w czasie. W celu uzasadnienia tego założenia, poniżej zostaną poddane analizie kolejno trzy przypadki:

- określenie formuły metryki kosztu łączy dla pierwszego żądania w zbiorze D ,
- określenie postaci metryki kosztu łączy dla drugiego żądania,
- określenie postaci metryki kosztu łączy dla ostatniego żądania w zbiorze.

W przypadku, kiedy CZRS otrzymuje zbiór żądań D i przystępuje do określenia trasy dla pierwszego żądania, zgodnie z wcześniejszymi rozważaniami, metryka kosztu łączy ma postać:

$$c(e_i) = \frac{c(d_j) + w(e_i)}{b(e_i)} \quad (3.26)$$

Powyższa metryka kosztu łączy jest ilorazem sumy wymaganej przepustowości dla instalacji trasy $c(d_j)$ i odpowiedniej wartości wektora prognozowanych konfliktów $w(e_i)$ do przepustowości łącza $b(e_i)$. Uwzględnienie wartości elementu wektora $w(e_i)$ pozwala podnieść w metryce kosztu łącza znaczenie poziomu prognozowanych konfliktów o zasoby (przepustowości) łączy i ominąć łącza o wysokim prawdopodobieństwie tychże konfliktów. Składnik dotyczący wymaganej przepustowości $c(d_j)$ pozwala z kolei przewidzieć, jak zmieni się wartość metryki łącza $c(e_i)$ w przypadku instalacji trasy z wykorzystaniem tego łącza. Warto zwrócić uwagę, że wymagana przepustowość $c(d_j)$ występuje w tej metryce dwa razy. Pierwszy raz w postaci jawnej, a drugi raz w postaci probabilistycznej jako część wartości elementu wektora prognozowanych konfliktów $w(e_i)$ (wzór (3.25)).

Po wybraniu łącza e_i przy wyznaczaniu danej trasy dostępna przepustowość zmaleje o $c(d_j)$. Warto zwrócić uwagę, że składnik $w(q_j)c(d_j)$ metryki kosztu łącza stanowi dodatkowe podniesienie jej wartości, które może wprowadzić dodatkowe błędy. Dlatego obliczając wartość metryki kosztu łącza, warto wyeliminować ten

wpływ poprzez przeliczenie wartości wektora prognozowanych konfliktów w następujący sposób:

$$w^j(e_i) = w(e_i) - w_i(q_j) c(d_j); \quad (3.27)$$

gdzie:

$w^j(e_i)$ – wartość elementu wektora prognozowanych konfliktów $w(e_i)$ dla żądania d_j .

Metryka kosztu łącza dla pierwszego żądania przekształca się wtedy do postaci:

$$c(e_i) = \frac{c(d_1) + w^1(e_i)}{b(e_i)} \quad (3.28)$$

Trasa realizacji drugiego żądania ma być wyznaczona w sytuacji, gdy określone przepustowości łącza zostały już zarezerwowane dla trasy obsługi pierwszego żądania. Z tego powodu, warto to uwzględnić w definicji wartości metryki kosztu łącza użytej do wyznaczenia trasy obsługi drugiego żądania zgodnie z formułą (3.29).

$$c(e_i) = \frac{g(e_i) + c(d_2) + w^2(e_i)}{b(e_i)} \quad (3.29)$$

gdzie $w^2(e_i)$ możemy policzyć jako:

$$w^2(e_i) = w^1(e_i) - w_i(q_j) c(d_j); \quad (3.30)$$

W taki sposób wpływ wykrytej trasy dla żądania d_1 jest wzięty pod uwagę w wartości $g(e_i)$. Wartość $c(d_2)$ bierze pod uwagę wymaganą przepustowość dla instalacji trasy dla żądania d_2 . Składniki $g(e_i)$ i $c(d_2)$ razem podnoszą wartość metryki kosztu łącza, których zasoby (przepustowości) są bliskie do wyczerpania z

powodu instalacji tras dla d_1 i d_2 . Wartość $w^2(e_i)$ podnosi wartość metryki kosztu dla tych łączy, dla których prawdopodobieństwo wystąpienia konfliktów o zasoby łączy jest wysokie. $w^2(e_i)$ nie zawiera kwantów dla żądań d_1 i d_2 , ponieważ ich wpływ już nie ma charakteru probabilistycznego i one są prezentowane w postaci jawnej.

Dla ostatniego żądania $d_{|D|}$ w ogóle nie mają znaczenia prognozowane konflikty. Dla ostatniego żądania metryka kosztu łączy przyjmuje następującą postać:

$$c(e_i) = \frac{g(e_i) + c(d_{|D|})}{b(e_i)} \quad (3.31)$$

Analizując wszystkie trzy przypadki, możemy na nowo zdefiniować metrykę kosztu łączy dla dowolnego żądania d_j ze zbioru D :

$$c(e_i) = \frac{g(e_i) + c(d_j) + w^j(e_i)}{b(e_i)} \quad (3.32)$$

gdzie:

$w^j(e_i)$ - może być policzona za pomocą formuły (3.33).

$$w^j(e_i) = w(e_i) - \sum_{x=1}^j w_i(q_x) c(d_x) \quad (3.33)$$

Elementy wektora W bez normalizacji przyjmują wartości z przedziału $(0; \infty)$. W odróżnieniu od metody prezentowanej w podrozdziale 3.1, elementy wektora W przyjmują duże wartości tylko na początku podczas wykrywania tras dla pierwszych żądań ze zbioru D i maleją podczas wykrywania trasy dla każdego kolejnego żądania. Z tego powodu wpływ wektora W na metrykę kosztu łączy jest wysoki w sytuacji gdy nie ma innej informacji niż prognoza dotycząca konfliktów. Wpływ ten spada z kolei do zera, kiedy algorytm dysponuje informacją rzeczywistą o zarezerwowanych zasobach. Pozwala to uniknąć normalizacji i/lub modyfikacji wartości elementów wektora W (wzory (3.7)–(3.9)), ponieważ, w odróżnieniu od

poprzedniej metody, czynnik prognozowanych konfliktów nie ma dużego wpływu na końcu działania metody. Tuż po uruchomieniu algorytmu (tj. w momencie, kiedy nie występują jeszcze zainstalowane trasy), duży wpływ wektora prognozowanych konfliktów nie doprowadzi do błędów, lecz jest jedynym/najważniejszym źródłem informacji o możliwych konfliktach.

W powyższy sposób metryka kosztu łącza zmienia się dla każdego nowego żądania. Faktycznie, przed obliczeniem pierwszego żądania wszystkie zasoby łącza wymagane dla instalacji tras są w postaci probabilistycznej/kwantowej (są reprezentowane przez $w(e_i)$). Jednak podczas wykrywania i po wykryciu trasy zasoby wymagane dla jej instalacji przekształcane są do postaci jawnej (są reprezentowane przez $g(e_i)$ i $c(d_i)$). W taki sposób, metryka kosztu (3.32) pozwala algorytmowi ominąć łącza, których zasoby są na wyczerpaniu z powodu rezerwacji przepustowości dla wcześniej rozpatrzonych żądań. W tym samym czasie pozwala to ominąć łącza, dla których prawdopodobieństwo wyczerpania zasobów jest wysokie biorąc od uwagę tylko te żądania, które nie zostały obsłużone.

3.2.3 Dobór współczynnika krzywizny funkcji odwrotnej proporcjonalności (γ) oraz liczby najkrótszych tras (k)

W celu określenia optymalnej liczby tras k dla budowania kwantu oraz współczynnika krzywizny funkcji odwrotnej proporcjonalności γ ze wzoru (3.21) zostały przeprowadzone badania dla wybranych sieci (USA, Włoskiej, Pionier) oraz trzech modeli ruchu sieciowego ukazanych w Tabeli 3.8.

Model	Przedziały wymaganej przepustowości dla instalacji trasy $c(d)$	Rozkład prawdopodobieństwa wymaganej przepustowości dla instalacji trasy
A0_2	[1;2] kanały	Równomierny
A0_4	[1;4] kanały	Równomierny
A0_8	[1;5] kanały	Równomierny

Tabela 3.8 Modele ruchu sieciowego

Oczekiwany jest wzrost liczby k wraz ze wzrostem obciążenia sieci, ponieważ im większe jest obciążenie sieci, tym dokładniejsze powinny być prognozy dotyczące konfliktów o zasoby łącza. Liczba k faktycznie jest wskaźnikiem tej dokładności podczas budowania kwantów. Z kolei wartość γ nie ulega analizie i jest określana w sposób empiryczny.



W ramach 7 200 symulacji zostały przebadane wartości γ od 1 do 10, wartości k od 1 do 80, 3 modele ruchu sieciowego oraz 3 topologie sieci. Wyniki symulacji potwierdzają, że wykorzystanie dziesięciu najkrótszych tras jest wystarczające dla budowania kwantów dla wybranych topologii sieci i modeli ruchu. Z drugiej strony badania ukazują, że optymalna uniwersalna para parametrów (γ, k) nie istnieje. Zależy ona od modelu ruchu sieciowego oraz od topologii sieci. Najlepsze zestawy parametrów (γ, k) dla różnych topologii sieci i modelu ruchu są zaprezentowane w Tabeli 3.9:

Model ruchu sieciowego	Sieć USA	Sieć Włoska	Sieć Pionier
A0_2	$\gamma = 1, k = 4$	-	$\gamma = 1, k = 1$
A0_4	$\gamma = 3, k = 6$	$\gamma = 6, k = 4$	$\gamma = 1, k = 6$
A0_8	$\gamma = 2, k = 7$	$\gamma = 5, k = 5$	$\gamma = 1, k = 8$

Tabela 3.9 Optymalne wartości dla (γ, k) w zależności od modelu ruchu i topologii sieci

Brak wyników w Tabeli 3.9 dla modelu ruchu sieciowego A0_2 i topologii Sieci Włoskiej jest spowodowany tym, że współczynnik blokowania w tym scenariuszu wynosi 0% (algorytm umożliwia instalację tras dla wszystkich żądań bez blokowania). Jak zaznaczono wcześniej, wraz ze wzrostem obciążenia sieci wzrasta wymagana dokładność dla obliczenia kwantów (liczba tras k wymagana dla budowania kwantów rośnie). Brak uniwersalnej pary (γ, k) wymaga innego podejścia dla określenia wartości optymalnych parametrów przed rozpoczęciem działania algorytmu. W ramach niniejszej pracy proponowana jest poniższa strategia:

1. Wykryj 10 najkrótszych tras dla każdej pary węzłów w sieci.
2. Przeanalizuj statystyki ruchu sieciowego w sieci i zdefiniuj zbiór modeli ruchu sieciowego $H = \{h_1, h_2, \dots, h_{|H|}\}$.
3. Przeprowadź symulacje dla każdego modelu h_i w celu wykrycia najlepszej pary parametrów $\gamma(h_i)$ i $k(h_i)$.
4. Dla zbioru żądań D przeanalizuj ten zbiór i sklasyfikuj scenariusze do jednego z istniejących modeli ruchu m sieciowego ze zbioru H .
5. Oblicz kwanty przy wykorzystaniu współczynników $\gamma(h)$ i $k(h)$ odpowiedniego modelu ruchu.

Problem wyboru optymalnego klasyfikatora jest poza zakresem niniejszej pracy. Zagadnienie to jest poruszane np. w pracach [62]-[64].

3.2.4 Wyniki badań symulacyjnych dla proponowanej metody

W celu weryfikacji przydatności budowania kwantów na podstawie ograniczonej liczby k najkrótszych tras zostały prowadzone badania symulacyjne dla trzech wybranych topologii sieci i trzech modeli ruchu sieciowego (Tabela 3.8). Przeprowadzono 100 symulacji dla każdej pary (topologia sieci, model ruchu sieciowego). Jako podejście referencyjne został wybrany schemat w oparciu o metrykę kosztu łącza (wzór (3.1)).

Wyniki symulacji dotyczące znormalizowanej przewagi nowej metody nad podejściem referencyjnym pod względem współczynnika blokowania żądań są zaprezentowane w Tabeli 3.10.

Model ruchu sieciowego	Sieć USA	Sieć Włoska	Sieć Pionier
A0_2	8.24%	-	6.66%
A0_4	2.91%	4.69%	2.38%
A0_8	3.03%	2%	1.59%

Tabela 3.10 Znormalizowane wartości przewagi proponowanego podejścia nad podejściem referencyjnym pod względem współczynnika blokowania żądań

Analogicznie jak w poprzedniej Tabeli 3.9, brak danych dla modelu ruchu sieciowego A0_2 i topologii sieci Włoskiej jest spowodowany tym, że współczynnik blokowania w tym scenariuszu wynosi 0% (zarówno dla proponowanego podejścia, jak i dla podejścia referencyjnego). Z zaprezentowanych powyżej rezultatów wynika, że proponowane podejście jest lepsze od podejścia referencyjnego. Średni znormalizowany zysk odnośnie redukcji współczynnika blokowania żądań dla 8 scenariuszy dla proponowanej metody wynosi 3.94%. Maksymalny zysk redukcji współczynnika blokowania żądań wynosi 8.24% dla scenariusza z modelem ruchu A_02 i topologii sieci USA.

3.3 Wnioski

Pozytywne rezultaty symulacji potwierdzają możliwość praktycznego wykorzystania zaprezentowanej w niniejszym rozdziale metody. Koncepcja budowania kwantu ze zbioru wszystkich tras może być wykorzystana dla redukcji prawdopodobieństwa blokowania żądań w przypadku doboru tras ukierunkowanego na zrównoważenie stopnia wykorzystania przepustowości łączy. Wyniki badań symulacyjnych wskazują na przewagę tej metody nad podejściem referencyjnym nawet o 53.6%. Jednakże, wyznaczenie wszystkich możliwych tras pomiędzy każdą parą węzłów jest dość czasochłonne i mimo tego, że jest przydatne dla sieci o nieziennej topologii, powoduje problemy w przypadku zastosowania w algorytmach niezawodnego doboru tras zapewniających ochronę przed awariami wielokrotnymi. Powyższy problem można rozwiązać modyfikując sposób budowania kwantów poprzez zastosowanie zaprezentowanej metody budowania kwantów na podstawie ograniczonej liczby k najkrótszych tras.

We wszystkich analizowanych przypadkach modeli ruchu sieciowego i topologii sieci, widoczna jest przewaga pod względem współczynnika blokowania żądań proponowanego rozwiązania w porównaniu do podejścia referencyjnego.



4 Zastosowanie metod transformacyjnych do opracowania metody niezawodnego doboru tras o istotnie ograniczonym czasie obliczeń

W niniejszym rozdziale przedstawione są sposoby wykorzystania metod transformacyjnych w celu uzyskania redukcji czasu niezbędnego do realizacji niezawodnego doboru tras. Jak zostało opisane w rozdziale 2, możliwe jest zmniejszenie stopnia zapotrzebowania na przepustowości łączy dla tras zabezpieczających dzięki redukcji czasu potrzebnego do wyznaczenia zbiorów tras rozłącznych. Przypadek awarii wielokrotnej obejmujący sekwencję awarii pojedynczych występujących w małych odstępach czasu może być bowiem rozpatrywany jako sekwencja awarii pojedynczych.

Redukcja czasu obliczeń pozwala często zakończyć wykrywanie nowych tras po awarii danego elementu szybciej niż wystąpi kolejna awaria. Z tego powodu dla ochrony ruchu przed awarią wielokrotną jest często wystarczające wyznaczanie dla każdego żądania pary rozłącznych tras między kolejnymi awariami pojedynczymi zamiast k rozłącznych tras (gdzie $k > 2$). Ograniczenie liczby zapasowych tras potrzebnych dla ochrony ruchu sieciowego redukuje ilość zasobów łączy potrzebnych do instalacji tras zabezpieczających dla danego żądania, co z kolei zwiększa efektywność wykorzystania przepustowości łączy.

W podrozdziale 4.1 są opisane dwa znane podejścia doboru rozłącznych tras i wybrana metoda referencyjna. W podrozdziale 4.2 jest zaprezentowana nowa metoda wykrycia pary węzłowo-rozłącznych tras wykorzystująca autorską koncepcję transformacji planarnego grafu topologii sieci do postaci innej struktury zwanej w niniejszej pracy „mapą komórkową”. Jak zostało ukazane wykorzystanie grafu mapy komórkowej G^M dla doboru pary węzłowo-rozłącznych tras pozwala istotnie zredukować czas wykrycia tras, co zostało udowodnione wynikami symulacji. W Aneksie 8 jest omówiona możliwość zastosowania analogicznych metod transformacji w innych przypadkach (obejmujących grafy nieplanarne czy warianty

wyznaczania k rozłącznych tras). W Aneksie 9 została także naświetlona koncepcja budowania hybrydowych algorytmów łączących cechy metody zaprezentowanej w niniejszym rozdziale oraz metody z rozdziału poprzedniego.

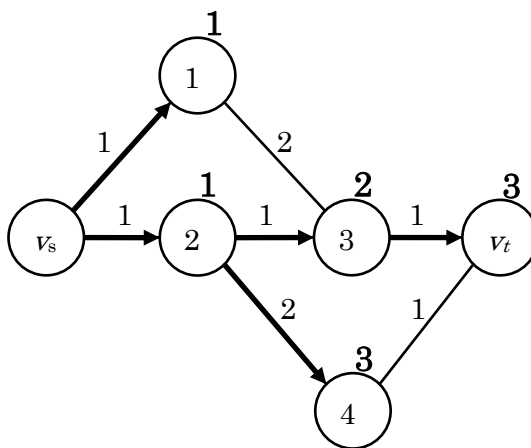
4.1 Znane metody doboru par węzłowo-rozłącznych tras

W niniejszym podrozdziale zostały opisane najważniejsze dostępne w literaturze metody doboru węzłowo-rozłącznych tras wraz z uzasadnieniem wyboru metody referencyjnej do badań symulacyjnych. W celu wyboru odpowiedniej metody referencyjnej, zostały przeanalizowane dwa znane podejścia: algorytm Suurballe-Tarjana [65] i algorytm Bhandariego [67].

4.1.1 Algorytm Suurballe-Tarjana

Algorytm Suurballe-Tarjana jest zaprojektowany dla wykrycia par węzłowo-rozłącznych tras i składa się z sześciu poniższych kroków.

Krok 1: Uruchom algorytm Dijkstry [50] w celu wykrycia drzewa najkrótszych ścieżek od źródłowego wierzchołka v_s do pozostałych wierzchołków w grafie G (Rys. 4.1).



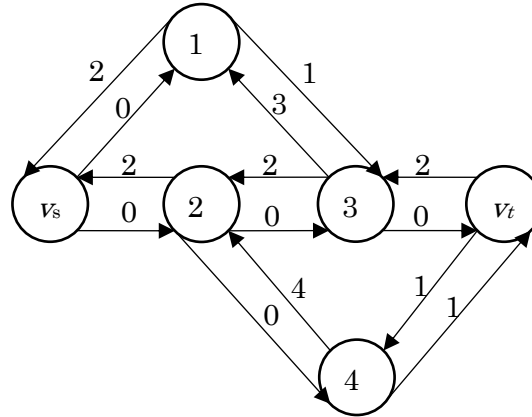
Rys. 4.1 Ilustracja Kroku 1 działania algorytmu Suurballe-Tarjana

Krok 2: Zastąp krawędzie grafu e_{ij} na skierowane łuki l_{ij} o koszcie:

$$c(l_{ij}) = c(e_{ij}) + c(P(v_s, v_i)) - c(P(v_s, v_j)) \quad (4.1)$$

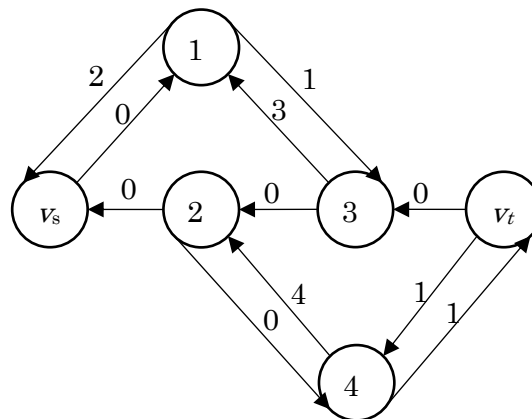
gdzie:

$c(P(v_s, v_i))$ – koszt najkrótszej trasy od źródłowego wierzchołka v_s do wierzchołka v_i .



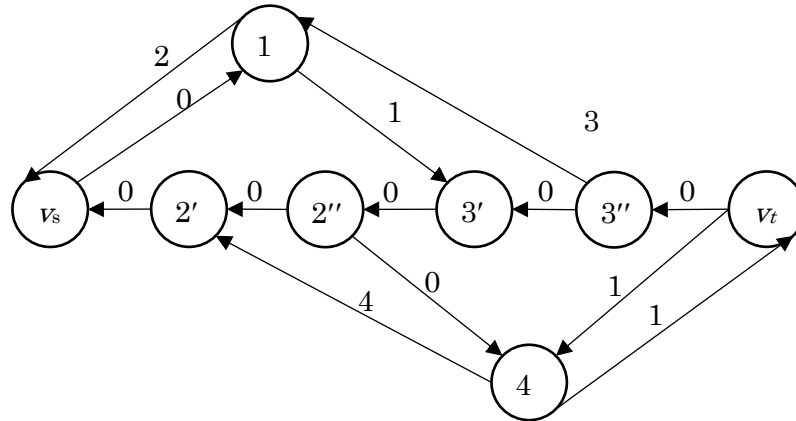
Rys. 4.2 Ilustracja Kroku 2 działania algorytmu Suurballe-Tarjana

Krok 3: Usuń z grafu G wszystkie łuki w najkrótszej ścieżce pomiędzy wierzchołkiem źródłowym v_s a docelowym v_t skierowane w kierunku węzła źródłowego v_s . Odwróć kierunki łuków w najkrótszej ścieżce (Rys. 4.3).



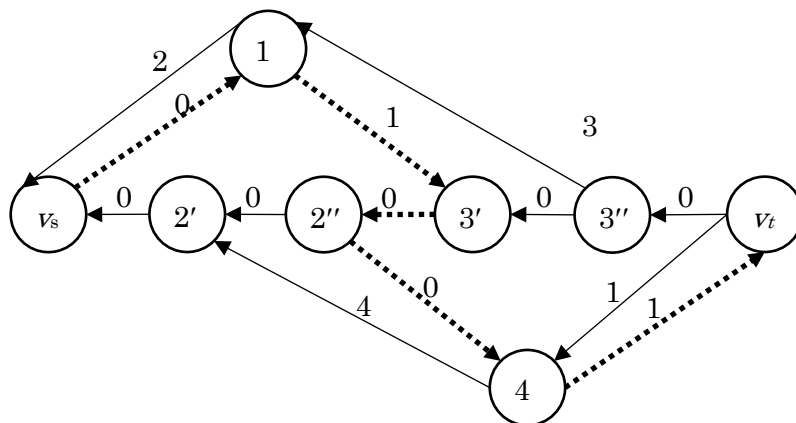
Rys. 4.3 Ilustracja Kroku 3 działania algorytmu Suurballe-Tarjana

Krok 4: Zastąp każdy wierzchołek tranzycyjny v w najkrótszej trasie $P(v_s, v_t)$ na dwa wierzchołki v' i v'' połączone łukiem $I_{v',v}$ o zerowym koszcie skierowanym w kierunku węzła źródłowego. Uczyń wszystkie łuki wychodzące z wierzchołka v wychodzącymi z v'' a wchodzące do wierzchołka v wchodzącymi do v' (Rys. 4.4).



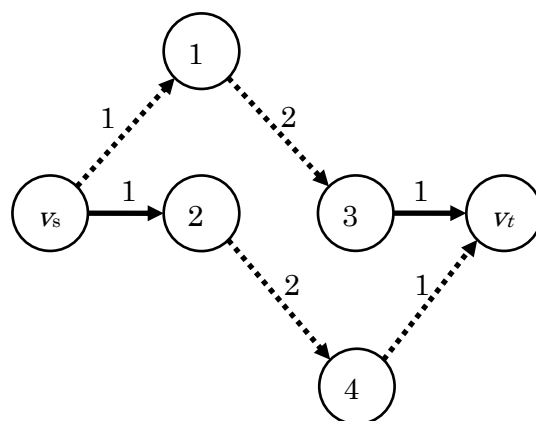
Rys. 4.4 Ilustracja Kroku 4 działania algorytmu Suurballe-Tarjana

Krok 5: Ponownie uruchom algorytm Dijkstry dla wykrycia najkrótszej trasy $P(v_s, v_t)$ (Rys. 4.5).



Rys. 4.5 Ilustracja Kroku 5 działania algorytmu Suurballe-Tarjana

Krok 6: Powrót do pierwotnego stanu grafu topologii sieci. Usuń wspólne krawędzie dla wyznaczonych dwóch najkrótszych ścieżek. Pozostałe krawędzie tworzą wynikową parę rozłącznych tras (Rys. 4.6).



Rys. 4.6 Ilustracja Kroku 6 działania algorytmu Suurballe-Tarjana

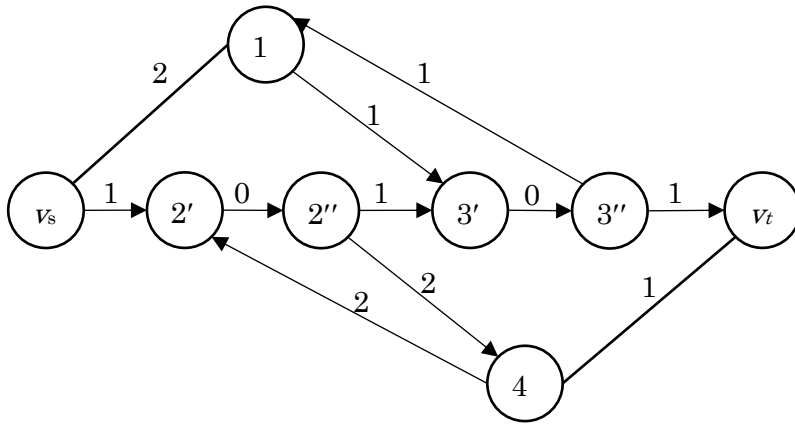
4.1.2 Algorytm Bhandariego

Algorytm Bhandariego działa w sposób podobny do powyższego algorytmu Suurballe-Tarjana. Jednakże w inny sposób dokonuje zmiany kosztu łuków i krawędzi, co powoduje obecność w grafie łuków o ujemnym koszcie. Z tego powodu w celu wykrycia najkrótszych tras algorytm Bhandariego korzysta ze zmodyfikowanego algorytmu Dijkstry [67] lub z algorytmu BFS [68]. Algorytm Bhandariego obejmuje pięć poniższych kroków.

Krok 1: Uruchom zmodyfikowany algorytm Dijkstry lub BFS dla wykrycia najkrótszej trasy pomiędzy wierzchołkami źródłowym a docelowym.

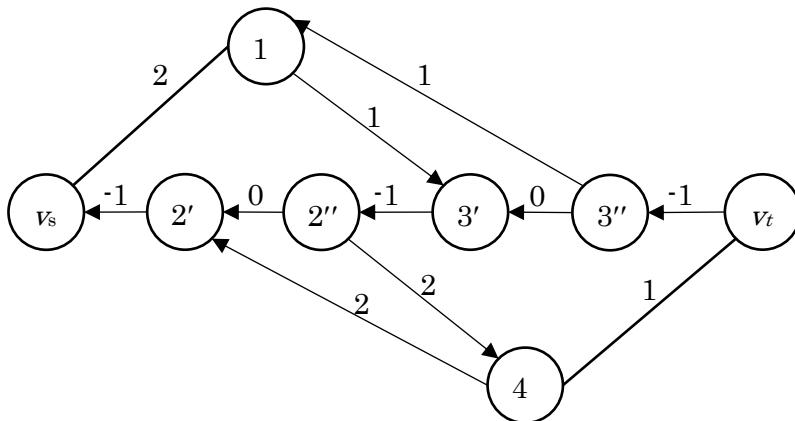
Krok 2: Zastąp krawędzie zawarte w wykrytej ścieżce na łuki skierowane w stronę węzła docelowego.

Krok 3: Zastąp każdy wierzchołek tranzycyjny v w najkrótszej trasie na dwa wierzchołki v' i v'' połączone przez łuk $l_{v',v''}$ o zerowym koszcie skierowany w kierunku węzła docelowego. Uczyń wszystkie łuki wychodzące z wierzchołka v wychodzącymi z v'' a wchodzące do wierzchołka v wchodzącymi do v' (przykład działania algorytmu jest podany na Rys. 4.7 dla tego samego grafu, który był wcześniej wykorzystany w przykładzie dla algorytmu Suurballe-Tarjana).



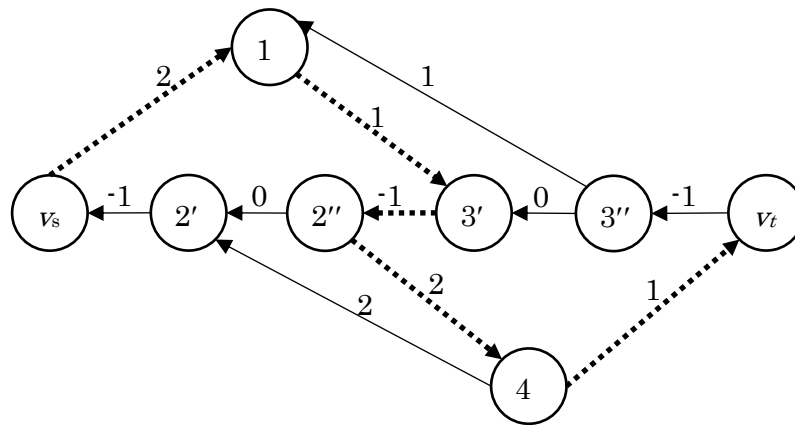
Rys. 4.7 Ilustracja Kroku 3 działania algorytmu Bhandariego

Krok 4: Odwróć kierunek wszystkich łuków w najkrótszej ścieżce. Uczyń ich koszty ujemnymi (Rys. 4.8).



Rys. 4.8 Ilustracja Kroku 4 działania algorytmu Bhandariego

Krok 5: Ponownie uruchom zmodyfikowany algorytm Dijkstry lub BFS dla wykrycia najkrótszej trasy (Rys. 4.9).



Rys. 4.9 Ilustracja Kroku 5 działania algorytmu Bhandariego

Krok 6: Powrót do pierwotnego stanu grafu topologii sieci. Usuń wspólne krawędzie dla dwóch najkrótszych ścieżek. Pozostałe krawędzie tworzą wynikową parę rozłącznych tras (wynik jest analogiczny do zaprezentowanego wcześniej na Rys. 4.6).

4.1.3 Wybór podejścia referencyjnego

W odróżnieniu od metody Suurballe-Tarjana, podczas modyfikacji kosztów algorytm Bhandariego operuje na ujemnych kosztach łuków, co z kolei nie pozwala wykorzystać algorytmu Dijkstry (algorytm Dijkstry nie działa na grafach z łukami/krawędziami o ujemnym koszcie). Z tego powodu algorytm wykorzystuje zmodyfikowany algorytm Dijkstry lub BFS. Algorytmy te nie są tak efektywne jak klasyczny algorytm Dijkstry [67]. Z drugiej strony, algorytm Bhandariego nie wymaga modyfikacji każdej krawędzi w grafie (lecz tylko tych które są incydentne względem węzłów tranzytowych zawartych w najkrótszej trasie). Takie modyfikacje nie wymagają dodatkowych obliczeń kosztów (np. podobnych do wzoru (4.1)). Wszystko to robi algorytm Bhandariego efektywniej niż metoda Suurballe-Tarjana pod względem czasu obliczeń [67]. Warto zwrócić uwagę, że optymalność działania algorytmu Bhandariego dotycząca kosztu wyznaczonych tras jest udowodniona matematycznie w pracy [67].

Z powyżej opisanych powodów jako algorytm referencyjny jest wybrany algorytm Bhandariego. W następnym podrozdziale są zaprezentowane podstawy autorskiej metody, która istotnie różni się od podejść Bhandariego oraz Suurballe-Tarjana.

4.2 Metoda transformacyjna dla redukcji czasu wykrycia pary węzłowo-rozłącznych tras w sieciach teleinformatycznych

W mniejszym podrozdziale jest ukazana metoda transformacyjna doboru tras niezawodnych, która zamiast wykorzystania grafu topologii sieci (w którym wierzchołki reprezentują węzły sieci) dla wykrycia pary węzłowo-rozłącznych tras korzysta ze struktury „mapy komórkowej”. Graf G^M mapy komórkowej jest grafem, w którym wierzchołki reprezentują nie węzły sieci, lecz cykle bazowe/cykle niepodzielne (cykle bazowe zostały opisane w sekcji 4.2.1). Proponowany algorytm pozwala wykryć zbiór cykli bazowych, które finalnie są transformowane do postaci par węzłowo-rozłącznych tras.

Jak zostało opisane poniżej, w topologii sieci szkieletowych zazwyczaj liczba węzłów jest większa niż liczba cykli niepodzielnych. Z tego powodu prezentowana metoda operuje na grafie G^M o mniejszej liczbie wierzchołków niż w grafie G co powoduje przewagę proponowanego algorytmu nad referencyjnymi rozwiązaniami z literatury. Wyniki symulacji potwierdzają tę przewagę dla wszystkich analizowanych sieci.

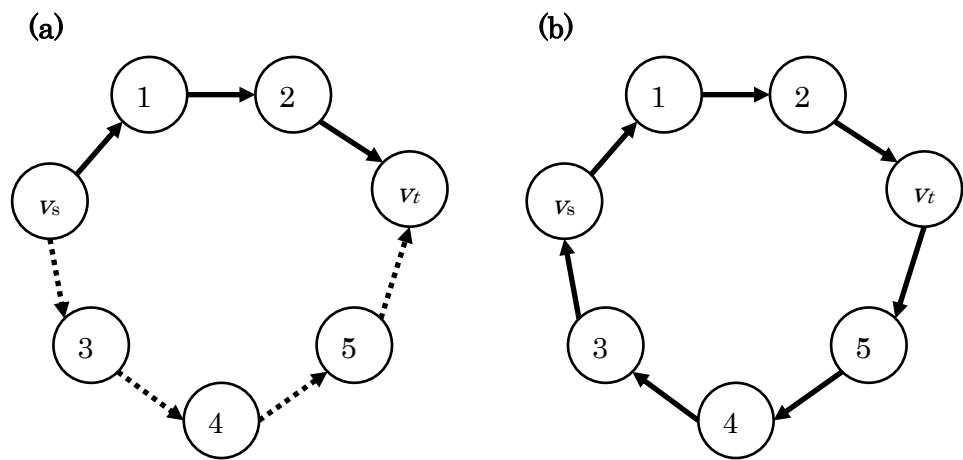
Niniejszy podrozdział ukazuje wyniki autorskich badań opublikowane w pracach [69], [57], [70].

4.2.1 Podstawy nowej metody

Nowa metoda jest oparta na dwóch głównych założeniach. Pierwszym założeniem jest twierdzenie, że para węzłowo-rozłącznych tras o najmniejszym koszcie w nieskierowanym grafie pomiędzy węzłami v_s i v_t jest równoważna cyklowi o najmniejszym koszcie zawierającemu te same węzły v_s i v_t . Równoważny cykl w niniejszej rozprawie oznacza cykl, który zawiera tę samą sekwencję krawędzi co odpowiadająca mu para węzłowo-rozłącznych tras. Na przykład na (Rys. 4.10(a)) są zaprezentowane dwie węzłowo-rozłączne trasy $P_1=(v_s, 1, 2, v_t)$ i $P_2=(v_s, 3, 4, 5, v_t)$, które zawierają te same krawędzie: $\{(v_s, 1), (1, 2), (2, v_t), (v_t, 5), (5, 4), (4, 3), (3, v_s)\}$ co cykl $S=(v_s, 1, 2, v_t, 5, 4, 3, v_s)$ na Rys. 2.10(b).

Udowodnienie tej tezy nie jest trudne. Załóżmy, że w nieskierowanym grafie G istnieje cykl S o najmniejszym koszcie zawierający węzły v_s i v_t . Cykl S można podzielić na dwie rozłączne części/trasy $P_1(v_s, v_t)$ i $P_2(v_t, v_s)$. Niech istnieje trasa

$P_3(v_s, v_t)$ skierowana w kierunku odwrotnym niż trasa $P_2(v_t, v_s)$ i zawiera te same krawędzie co trasa P_2 . W takim razie trasy $P_1(v_s, v_t)$ i $P_3(v_s, v_t)$ nie zawierają wspólnych krawędzi, wspólnych wierzchołków tranzytowych i mają wspólne wierzchołki źródłowy i docelowy. Z tego wynika, że trasy $P_1(v_s, v_t)$ i $P_3(v_s, v_t)$ są trasami węzłowo-rozłącznymi pomiędzy parą węzłów v_s i v_t . Koszt pary tras $P_1(v_s, v_t)$ i $P_3(v_s, v_t)$ jest sumą kosztów wszystkich krawędzi zawartych w $P_1(v_s, v_t)$ i $P_3(v_s, v_t)$ i jest równy sumarycznemu kosztowi wszystkich łączy cyklu S . Z tego wynika, że w grafie G nie istnieje taki zbiór krawędzi, które można ułożyć w cykl zawierający wierzchołki v_s i v_t , których sumaryczny koszt jest mniejszy niż koszt cyklu S . Pamiętając o tym, że trasy $P_1(v_s, v_t)$ i $P_3(v_s, v_t)$ zawierają te same krawędzie co cykl S (kierunek tras w grafie nieskierowanym nie ma wpływu na koszt krawędzi), można wnioskować, że w grafie G nie istnieje zbiór krawędzi, który można ułożyć w parę węzłowo-rozłącznych tras pomiędzy v_s i v_t o koszcie mniejszym niż para tras $P_1(v_s, v_t)$ i $P_3(v_s, v_t)$. Z tego wynika, że najkrótszy cykl zawierający węzły v_s oraz v_t w grafie G jest równoważny parze węzłowo-rozłącznych tras pomiędzy v_s i v_t .

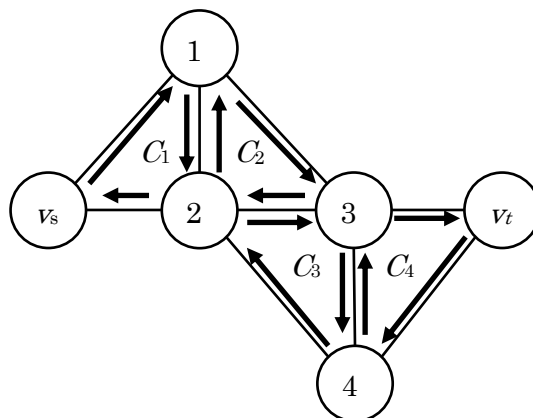


Rys. 4.10 Równoważność pary węzłowo-rozłącznych tras i cyklu

W taki sposób zadanie wykrycia pary węzłowo-rozłącznych tras o minimalnym koszcie w sieci pomiędzy wierzchołkami v_s i v_t może być przekształcone do zadania wykrycia cyklu o minimalnym koszcie zawierającym wierzchołki v_s i v_t . Take podejście zostało przebadane i wykorzystane po raz pierwszy w rozwiązaniu zorientowanym na ochronę transmisji broadcast (od pojedynczego węzła do pozostałych węzłów sieci) [71], [72]. Jednakże, w odróżnieniu od realizowanej w

powyższych pracach strategii wykrywania cykli w grafie reprezentującym topologię sieci, niniejszy podrozdział rozprawy prezentuje autorską metodę, której zadaniem jest wyznaczenie cykli w odniesieniu do topologii mapy komórkowej będącej strukturą pochodzącą od topologii sieci (otrzymaną w wyniku transformacji opisanej poniżej).

Drugim założeniem jest twierdzenie, że jakikolwiek cykl w grafie może być reprezentowany poprzez zbiór cykli niepodzielnych (cykli bazowych) tego grafu. Jako bazowe/niepodzielne cykle w niniejszej pracy są traktowane najkrótsze (względem liczby zawartych krawędzi – a nie względem metryk kosztu łączy) cykle grafu niezawierające krawędzi, które mogłyby podzielić takie cykle na mniejsze cykle. W przypadku grafu ukazanego na Rys. 4.1, graf ten ma dwa cykle bazowe: $C_1=(v_s, 1, 3, 2, v_s)$ oraz $C_2=(v_t, 4, 2, 3, v_t)$. W przypadku istnienia w grafie dwóch łączy $e_{1,2}$ oraz $e_{3,4}$ (Rys. 4.11), w grafie istnieją cztery cykle bazowe: $C_1=(v_s, 1, 2, v_s)$, $C_2=(1, 3, 2, 1)$, $C_3=(2, 3, 4, 2)$ oraz $C_4=(v_t, 4, 3, v_t)$.

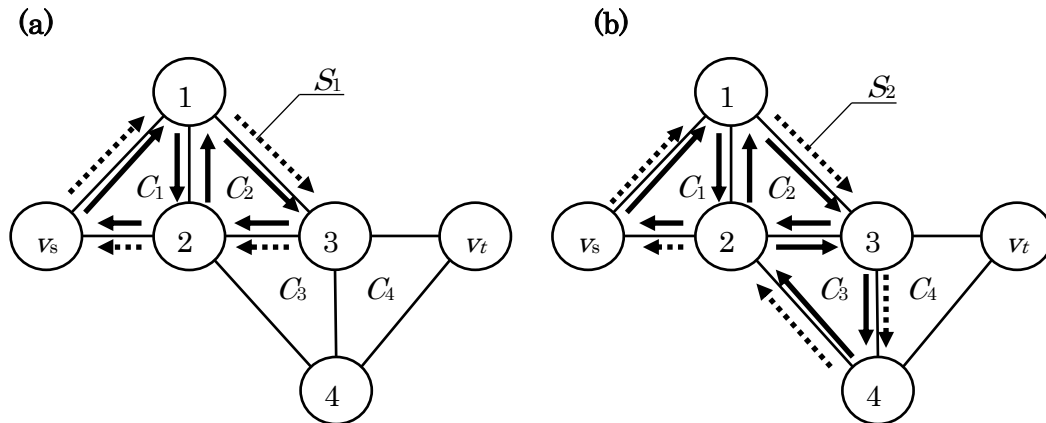


Rys. 4.11 Przykład cykli bazowych w grafie

W tym przypadku cykl $S_1=(v_s, 1, 3, 2, v_s)$ jest zbudowany w oparciu o zbiór cykli bazowych $C_1=(v_s, 1, 2, v_s)$, $C_2=(1, 3, 2, 1)$ – Rys. 4.12(a). Cykl $S_2=(v_s, 1, 3, 4, 2, v_s)$ jest natomiast określony poprzez zbiór cykli bazowych $C_1=(v_s, 1, 2, v_s)$, $C_2=(1, 3, 2, 1)$, $C_3=(2, 3, 4, 2)$ – Rys 4.12(b).

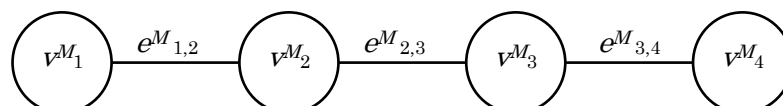
Warto zwrócić uwagę, że każdy cykl (w tym również cykl niepodzielny/bazowy) powstaje ze zbioru cykli bazowych (w przypadku cyklu niepodzielnego ten zbiór zawiera tylko jeden cykl bazowy), jednakże stwierdzenie odwrotne (tj. o tym, że można stworzyć cykl z dowolnego zbioru cykli bazowych) nie jest prawidłowe. Na

przykład zbiór cykli $\{C_1, C_4\}$ reprezentuje więcej niż jeden cykl, ponieważ nie istnieją wspólne krawędzie zawarte w tych dwóch cyklach bazowych. Z tego wynika, że każdy cykl (podzielny i niepodzielny) można przekształcić do postaci zbioru cykli bazowych dysponujących wspólnymi krawędziami (w przypadku, kiedy zbiór zawiera więcej niż jeden cykl bazowy).



Rys. 4.12 Przykład cykli podzielnych obejmujących zbiory cykli bazowych w grafie

Na podstawie powyższych założeń możemy zdefiniować graf innej struktury – graf mapy komórkowej G^M . Graf G^M mapy komórkowej tworzy para (V^M, E^M) , w której V^M jest zbiorem wierzchołków reprezentujących cykle bazowe grafu G , a zbiór E^M reprezentuje zbiory wspólnych krawędzi dla par cykli bazowych. Przykład grafu mapy komórkowej dla topologii sieci Rys. 4.11 jest zaprezentowany na Rys. 4.13, gdzie v^{M_1} , v^{M_2} , v^{M_3} , v^{M_4} są równoważne odpowiednio cyklom bazowym C_1 , C_2 , C_3 , C_4 na Rys. 4.11. Krawędzie $e^{M_{1,2}}$, $e^{M_{2,3}}$, $e^{M_{3,4}}$ świadczą o występowaniu wspólnych krawędzi odpowiednich cykli bazowych grafu G .



Rys. 4.13 Przykład mapy komórkowej dla topologii sieci ukazanej na Rys. 4.11

Są dwie charakterystyki w grafie G^M niezbędne dla wskazania najkrótszego cyklu zawierającego dane dwa węzły grafu G . Pierwszą wartością jest koszt cyklu bazowego $\alpha(v^M)$, który jest sumą kosztów wszystkich krawędzi zawartych w tym cyklu bazowym. Druga wartość jest kosztem krawędzi grafu mapy komórkowej $\alpha(e^M)$, która jest podwójnym kosztem odpowiadającego jej łącza w grafie G topologii sieci. W takim razie koszt dowolnego cyklu $\alpha(S)$ jest sumą kosztów wszystkich cykli bazowych zawartych w cyklu S pomniejszoną o wartość sumy kosztów wszystkich krawędzi mapy komórkowej pomiędzy tymi cyklami bazowymi:

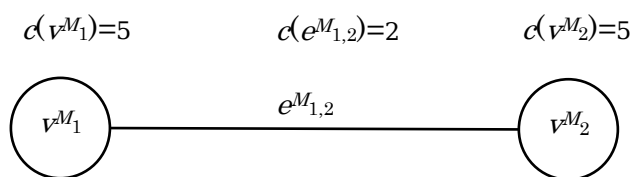
$$c(S) = \sum v^M_i - \sum e^M_j; \quad \forall v^M_i \in V^M(S), \forall e^M_j \in E^M(S) \quad (4.2)$$

gdzie:

$V^M(S)$ – zbiór wszystkich cykli bazowych zawartych w cyklu S ,

$E^M(S)$ – zbiór wszystkich krawędzi grafu mapy komórkowej pomiędzy cyklami bazowymi zawartymi w cyklu S .

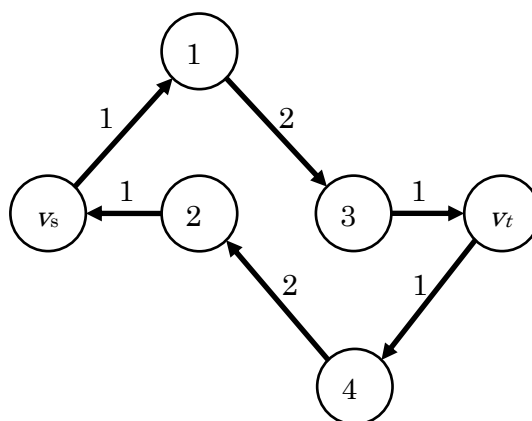
Z tego wynika, że np. topologia sieci z Rys. 4.1 może być transformowana do postaci grafu mapy komórkowej G^M zawierającego tylko trzy elementy: dwa cykle bazowe i jedną krawędź (Rys. 4.14).



Rys. 4.14 Przykład mapy komórkowej dla topologii sieci z Rys. 4.1

Cykl bazowy v^{M_1} zawiera węzły $V(v^{M_1}) = \{v_s, 1, 2, 3\}$ i krawędzie topologii sieci pomiędzy nimi. Sumaryczny koszt krawędzi grafu topologii sieci zawartych w cyklu bazowym v^{M_1} wynosi $\alpha(v^{M_1})=5$. Cykl bazowy v^{M_2} zawiera węzły $V(v^{M_2}) = \{v_t, 2, 3, 4\}$ i krawędzie topologii sieci pomiędzy nimi. Sumaryczny koszt krawędzi grafu topologii sieci zawartych w cyklu bazowym v^{M_2} wynosi $\alpha(v^{M_2})=5$. Krawędź mapy komórkowej $e^{M_{1,2}}$ zawiera jedną krawędź topologii sieci $E(e^{M_{1,2}}) = \{e_{2,3}\}$ o koszcie $\alpha(e_{2,3})=1$.

Z tego wynika, że koszt krawędzi mapy komórkowej wynosi $c(e^{M_{1,2}}) = 2$. Dla wykrycia pary węzłowo-rozłącznych tras pomiędzy węzłami v_s i v_t niezbędne jest połączenie cykli bazowych zawierających te węzły. Są to odpowiednio v^{M_1} dla v_s oraz v^{M_2} dla v_t . W wyniku tego uzyskujemy cykl S (Rys. 4.15), koszt którego możemy obliczyć za pomocą (4.2) jako: $c(S) = (5+5) - 2 = 8$.



Rys. 4.15 Przykład cyklu

Dla uzyskania pary węzłowo-rozłącznych tras wystarczy odwrócić kierunek części cyklu od v_t do v_s . W rezultacie, uzyskujemy wynik identyczny do wyniku działania algorytmu Bhandariego (Rys. 4.6).

Jednak, w przypadku proponowanej metody wystarczyła pojedyncza akcja w celu wykrycia cyklu, podczas gdy algorytm Bhandariego wykonałby w tym przypadku 11 iteracji w ramach algorytmu BFS lub zmodyfikowanego algorytmu Dijkstry.

Mapa komórkowa pozwala zbudować dowolne cykle za pomocą łączenia cykli bazowych. Połączone cykle bazowe zawierają krawędzie topologii grafu G sieci, które należą do łączonych cykli bazowych bez krawędzi topologii grafu G wspólnych dla łączonych cykli (oznaczonych poprzez krawędzie mapy komórkowej). Krawędzie mapy komórkowej pozwalają ocenić, które cykle bazowe można połączyć ze sobą. Koszty cykli bazowych oraz koszty krawędzi mapy komórkowej pozwalają łatwo obliczyć koszt dowolnego cyklu w grafie.

W taki sposób zadanie wykrycia pary węzłowo-rozłącznych tras pomiędzy węzłami v_s , v_t w grafie G można sprowadzić do zadania wykrycia najkrótszego cyklu



S w mapie komórkowej G^M odpowiadającej grafowi G , gdzie cykl S spełnia następujące warunki:

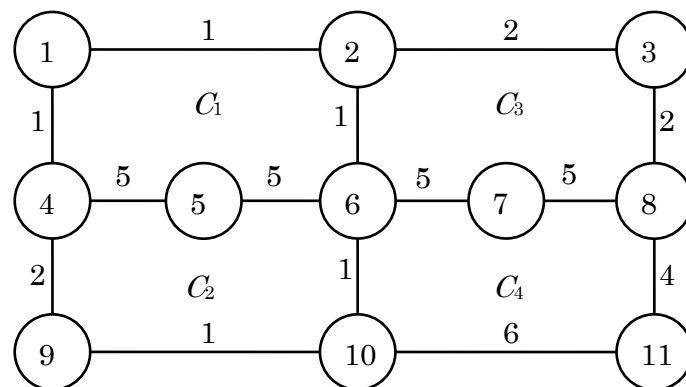
Warunek 1: Cykl S ma zawierać co najmniej jeden cykl bazowy ze zbioru cykli źródłowych V^{M_s} i co najmniej jeden cykl bazowy ze zbioru cykli docelowych V^{M_t} , gdzie zbiór cykli źródłowych V^{M_s} reprezentuje wszystkie cykle bazowe, które zawierają wierzchołek źródłowy v_s żądania, a zbiór cykli docelowych V^{M_t} reprezentuje wszystkie cykle bazowe, które zawierają wierzchołek docelowy v_t żądania.

Warunek 2: Zbiór S ma zawierać wierzchołki v_s i v_t grafu topologii sieci.

Warto zwrócić uwagę na fakt, że obecność w cyklu S cykli bazowych źródłowych/docelowych nie gwarantuje obecności w cyklu S węzłów źródłowych/docelowych. Dlatego warunek 1 równoważny warunkowi 2. Różnica w tych warunkach jest opisana poniżej.

W odróżnieniu od scenariuszy dla topologii sieci na Rys. 4.1, węzeł docelowy może być zawarty więcej niż w jednym cyklu bazowym. Na przykład węzeł 6 na Rys. 4.16 jest zawarty jednocześnie w kilku cyklach bazowych: C_1 , C_2 , C_3 i C_4 .

Z tego wynika, że w scenariuszach, gdzie węzeł 6 jest węzłem źródłowym, będą istniały cztery źródłowe cykle bazowe w grafie mapy komórkowej. Jeżeli węzeł 6 jest węzłem docelowym to graf mapy komórkowej będzie miał z kolei cztery docelowe cykle bazowe.



Rys. 4.16 Przykład topologii sieci i cykli bazowych

Cykli Bazowych (PPCB). Algorytm PPCB jest przedstawiony w formie pseudokodu poniżej.

Algorytm PPCB(G^M, G, v_s, v_t)

Parametry wejściowe:	
G^M	- Mapa komórkowa
G	- Graf topologii sieci
v_s	- Węzeł źródłowy żądania
v_t	- Węzeł docelowy żądania
Parametry wyjściowe:	
(P_1, P_2)	- Wykryta para tras
	Krok 1:
1	Zdefiniuj zbiór S_i dla każdego i -tego cyklu bazowego v^M_i w G^M . Oznacz koszt każdego cyklu $c(S_i)$ jako ∞ .
2	Zdefiniuj zbiory źródłowy V^M_s i docelowy V^M_t cykli bazowych na podstawie v_s i v_t .
3	Zdefiniuj pusty zbiór A dla cykli bazowych, które mają obliczone cykle S_i . Dodaj do A źródłowe cykle bazowe V^M_s .
	Dla każdego źródłowego cyklu bazowego $v^M_i \in V^M_s$:
	$S_i := v^M_i$;
	$c(S_i) := c(v^M_i)$;
4	Zdefiniuj pusty zbiór B dla cykli bazowych, które mają wykonane próby połączenia z sąsiednimi cyklami bazowymi;
	Krok 2:
5	Jeżeli: zbiór A nie jest pusty:
6	Dla cyklu bazowego $v^M_i \in A$ o minimalnym koszcie $c(S_i)$:
7	Usuń v^M_i z A ; Dodaj v^M_i do B ;
8	Dla każdego sąsiedniego cyklu bazowego $v^M_j \in V^M(v^M_i)$, który nie jest zawarty ani w B ($v^M_j \notin B$), ani w S_i :
9	Sprawdź warunek:
	$\begin{cases} E(v_s) \setminus E(E^M(S_i \cup v^M_j)) \geq 1 \\ E(v_t) \setminus E(E^M(S_i \cup v^M_j)) \geq 1 \end{cases}$
10	Jeżeli: warunek jest spełniony
11	Oblicz koszt nowego cyklu $c'(S_j)$:
	$c'(S_j) := c(S_i) + c(v^M_j) - \sum c(e^M_x)$;
	$\forall e^M_x \in (E^M(S_i) \cup E^M(v^M_j))$
12	Jeżeli: koszt nowego cyklu $c'(S_j)$ jest mniejszy niż $c(S_j)$:
13	Dodaj v^M_j do A ; usuń v^M_j z B jeżeli jest w B ;
	$S_j := (S_i \cup v^M_j)$; $c(S_j) := c'(S_j)$;
14	Jeżeli: koszt nowego cyklu $c'(S_j)$ jest mniejszy niż $c(S_j)$:
15	Dla każdego cyklu bazowego $\forall v^M_z \in (S_i \cup V^M(v^M_j))$:
16	Dodaj v^M_z do A ; usuń v^M_z z B jeżeli jest w B ;
	$S_z := (S_i \cup v^M_j)$; $c(S_z) := c'(S_j)$;
	Krok 3:
17	Wybierz ze zbioru B docelowy cykl bazowy v^M_t o minimalnym koszcie cyklu $c(S_i)$.
18	Podziel cykl S_i dla pary wierzchołków v_s i v_t żądania na dwie trasy (P_1, P_2) ;
	Zwróć (P_1, P_2) ;
	Koniec.

Algorytm PPCB jest algorytmem iteracyjnym i obejmuje trzy kroki. W Kroku 1 algorytm definiuje zbiory i ustawia stosowne wartości początkowe niezbędne dla poprawnego działania algorytmu. W Kroku 2 w sposób iteracyjny PPCB dokonuje prób połączenia cykli bazowych oraz aktualizuje stosowne zbiory i wartości. W Kroku 3 algorytm wybiera cykl o najmniejszym koszcie spośród wyznaczonych cykli zawierających źródłowy i docelowy wierzchołek żądania i dokonuje podziału tego cyklu na dwie trasy węzłowo-rozłączne.

Przykład

Przekazujemy do algorytmu PPCB mapę komórkową G^M (Rys. 4.18), która jest odwzorowaniem grafu topologii sieci G przedstawionej wcześniej na Rys. 4.16 oraz węzły 5 i 7 jako odpowiednio węzeł źródłowy v_s i docelowy v_t .

W Kroku 1 PPCB definiuje pusty zbiory S_i dla każdego cyklu bazowego i ustawia ich koszt jako ∞ (wiersz 1 algorytmu). Na podstawie informacji o wierzchołkach źródłowym i docelowym definiuje zbiory źródłowych cykli bazowych $V^M_s = V^M(v_s) = V^M(5) = (v^{M_1}, v^{M_2})$ i docelowych cykli bazowych $V^M_t = V^M(v_t) = V^M(7) = (v^{M_3}, v^{M_4})$ (wiersz 2 algorytmu). Algorytm definiuje zbiór A dla cykli bazowych dla których został policzony cykl S_i zawierający źródłowy wierzchołek i umieszcza do tego zbioru źródłowe cykle bazowe $V^M_s = (v^{M_1}, v^{M_2})$, ustawia koszty cykli S dla nich równe kosztom cykli bazowych $\alpha(S_1) = \alpha(v^{M_1}) = 13$, $\alpha(S_2) = \alpha(v^{M_2}) = 14$ (wiersz 3). PPCB tworzy pusty zbiór B dla tych cykli bazowych, dla których zostały wykonane próby połączenia z sąsiednimi cyklami bazowymi (wiersz 4 algorytmu).

Krok 2 wykonuje się iteracyjnie w sytuacji gdy zbiór A jest niepusty. Algorytm wybiera wśród cykli bazowych zawartych w $A = (v^{M_1}, v^{M_2})$ cykl o najmniejszym koszcie $\alpha(S)$ (wiersz 6 algorytmu). W pierwszej iteracji wybranym jest v^{M_1} z kosztem cyklu $\alpha(S_1) = 13$. Algorytm przenosi v^{M_1} ze zbioru A do zbioru B (wiersz 7). Następnie PPCB podejmuje próby połączenia cyklu S_1 z sąsiednimi cyklami bazowymi v^{M_2} i v^{M_3} . Algorytm sprawdza warunek obecności wierzchołków: źródłowego i docelowego w nowym cyklu, który powstanie w wyniku połączenia cykli bazowych v^{M_1} i v^{M_2} (wiersz 9 algorytmu).

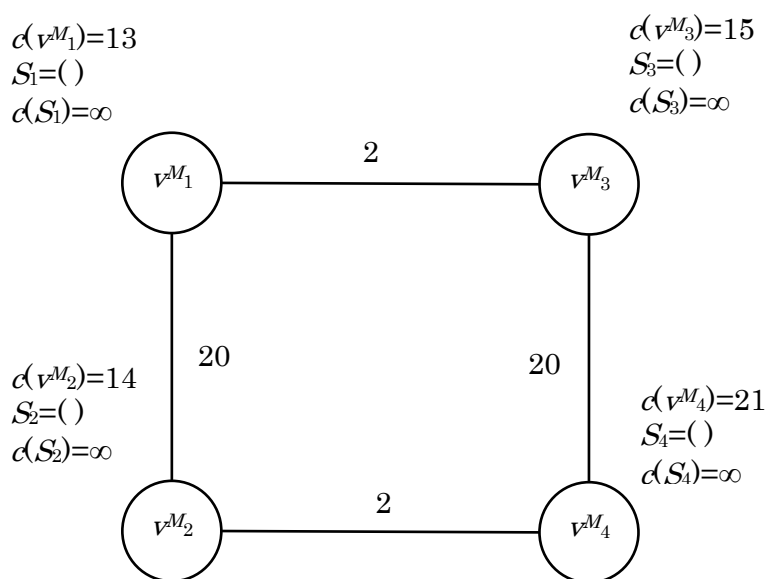
W wyniku takiego połączenia zostanie zbudowany cykl, w którym nie istnieje wierzchołek źródłowy. Faktycznie, jak widać na Rys. 4.17 taki cykl nie zawiera krawędzi incydentnej w stosunku do wierzchołka 5.

Z tego powodu pierwszy wzór z układu równań (4.3) nie jest prawidłowy, gdyż:

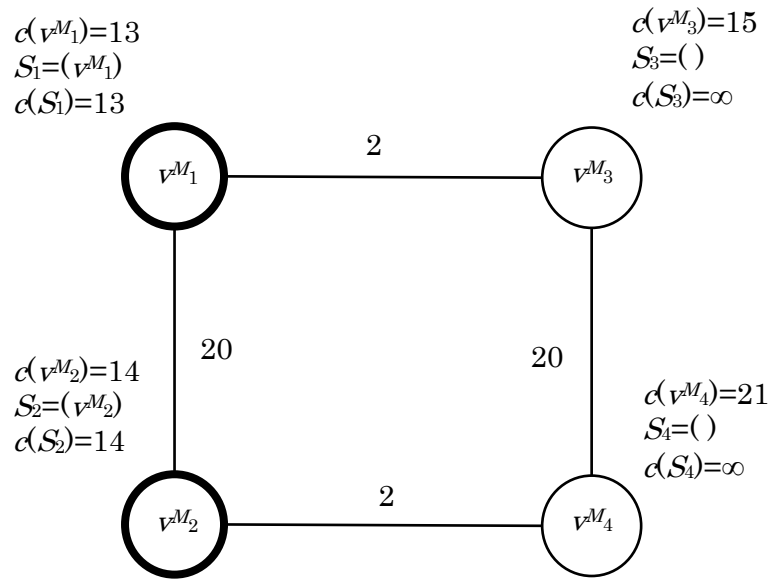
$$\begin{aligned} |E(v_5) \setminus E(E^M(S_1 \cup v^M_2))| &\geq 1 \\ |(e_{4,5}, e_{5,6}) \setminus E(E^M(v^M_1 \cup v^M_2))| &\geq 1 \\ |(e_{4,5}, e_{5,6}) \setminus E(e^M_{1,2})| &\geq 1 \\ |(e_{4,5}, e_{5,6}) \setminus (e_{4,5}, e_{5,6})| &\geq 1 \\ 0 &\geq 1 \end{aligned}$$

Dla drugiego sąsiedniego cyklu bazowego v^M_3 warunek 2 jest spełniony, więc PPCB oblicza koszt nowego cyklu $c'(S_3) = 13 + 15 - 2 = 26$ (wiersz 11 algorytmu). Wartość ta jest mniejsza niż aktualna wartość $\alpha(S_3) = \infty$, dlatego algorytm dodaje v^M_3 do zbioru A , aktualizuje zbiór $S_3 = (v^M_1, v^M_3)$ oraz jego koszt $\alpha(S_3) = 26$ (wiersze 12, 13). Koszt cyklu $\alpha(S_1)$ jest mniejszy, niż $c'(S_3) = 26$ (wiersz 14), dlatego cykl S_1 nie zostanie zaktualizowany. W przeciwnym razie oznaczałoby to, że mamy wykryty cykl o niższym koszcie nie tylko dla v^M_3 ale również dla v^M_1 .

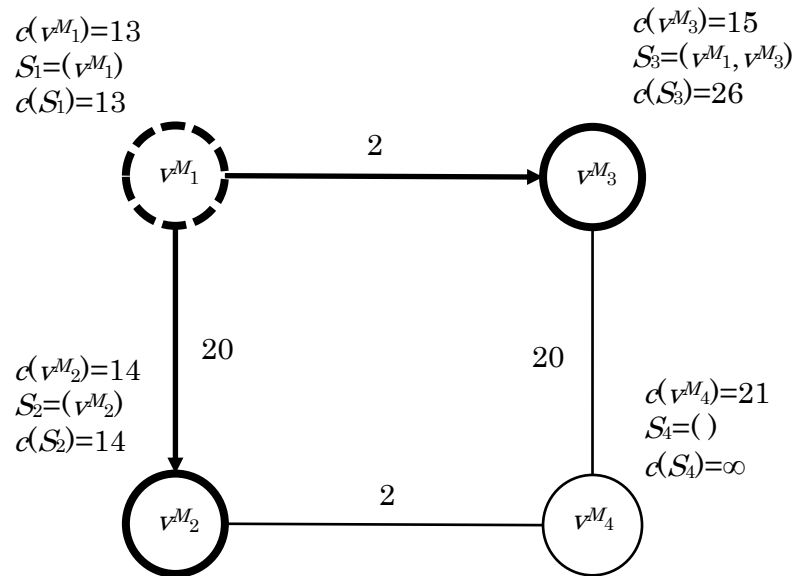
W takim scenariuszu dla prawidłowego działania PPCB warto byłoby zaktualizować S_1 i ponownie dodać v^M_1 do zbioru A , żeby powtórzyć próby połączenia nowego krótszego cyklu S_1 z sąsiednimi cyklami bazowymi. Scenariusz który wyjaśnia taki przypadek jest ukazany w Aneksie 7. Wynik pierwszej iteracji jest zaprezentowany na Rys. 4.20.



Rys. 4.18 Przykład mapy komórkowej



Rys. 4.19 Wynik Kroku 1 działania algorytmu PPCB



Rys. 4.20 Wynik działania pierwszej iteracji PPCB

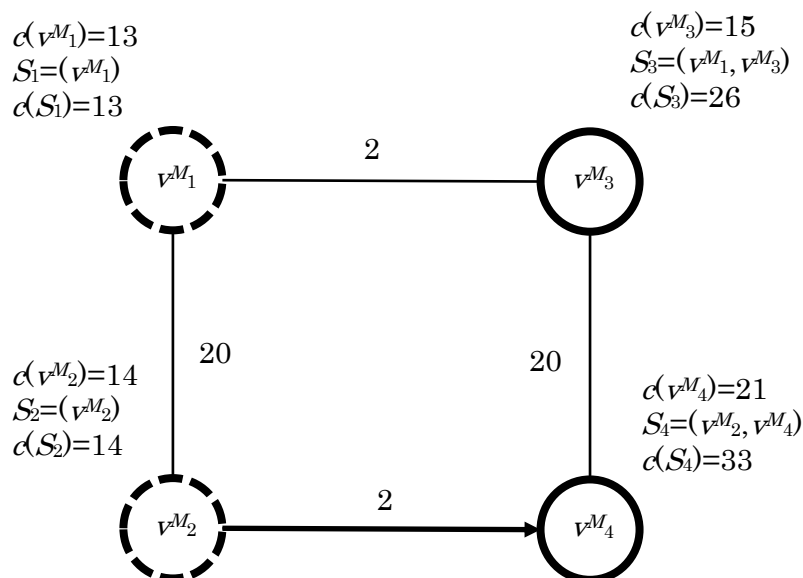
W drugiej iteracji wybranym jest v^{M_2} z kosztem cyklu $c(S_2)=14$. Algorytm usuwa v^{M_2} ze zbioru A i dodaje do zbioru B (wiersz 7). Cykl bazowy v^{M_2} ma tylko jednego sąsiada, który nie jest zawarty w B - v^{M_4} . PPCB sprawdza warunek obecności wierzchołków: źródłowego i docelowego w nowym cyklu, który powstanie w sposób połączenia cykli bazowych v^{M_2} i v^{M_4} . (wiersz 9). Warunek 2 jest spełniony, więc PPCB oblicza koszt nowego cyklu $c'(S_4) = 14 + 21 - 2 = 33$ (wiersz 11 algorytmu). Wartość

ta jest mniejsza niż aktualna wartość $\alpha(S_4) = \infty$, co powoduje umieszczenie v^{M_4} do zbioru A i aktualizację zbioru $S_4=(v^{M_2}, v^{M_4})$ oraz kosztu $\alpha(S_4) = 33$ (wiersze 12,13). Koszt cyklu $\alpha(S_2)$ jest mniejszy niż $c'(S_4) = 33$ (wiersz 14), dlatego cykl S_1 nie zostanie zaktualizowany. Wynik drugiej iteracji jest przedstawiony na Rys. 4.21.

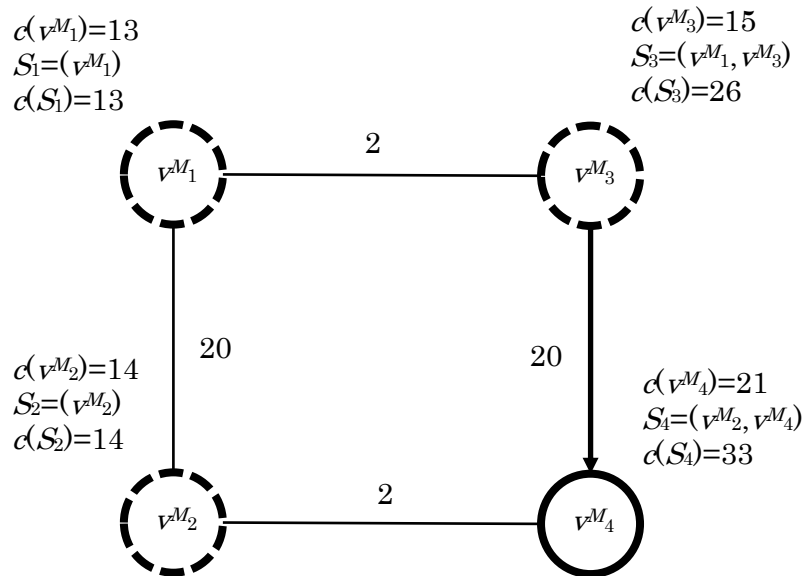
W trzeciej iteracji wybranym spośród cykli bazowych zawartych w $A=(v^{M_3}, v^{M_4})$ jest v^{M_3} z kosztem cyklu $\alpha(S_3)=26$. Algorytm usuwa v^{M_3} ze zbioru A i dodaje do zbioru B (wiersz 7). Cykl bazowy v^{M_3} ma tylko jednego sąsiada, który nie jest zawarty ani w B , ani w S_1 : v^{M_1} . Warunek obecności wierzchołków: źródłowego i docelowego w nowym cyklu, który powstanie w sposób połączenia cykli bazowych v^{M_2} i v^{M_4} nie jest spełniony (wiersz 9 algorytmu), gdyż:

$$\begin{aligned} |E(v_7) \setminus E(E^M(S_3 \cup v^{M_4}))| &\geq 1 \\ |(e_{6,7}, e_{7,8}) \setminus E(E^M((v^{M_1}, v^{M_3}) \cup v^{M_4}))| &\geq 1 \\ |(e_{6,7}, e_{7,8}) \setminus E(e^{M}_{3,4})| &\geq 1 \\ |(e_{4,5}, e_{5,6}) \setminus (e_{4,5}, e_{5,6})| &\geq 1 \\ 0 &\geq 1 \end{aligned}$$

Warunek 2 nie jest spełniony i PPCB rozpoczyna czwartą iterację. Wynik trzeciej iteracji jest przedstawiony na Rys. 4.22.

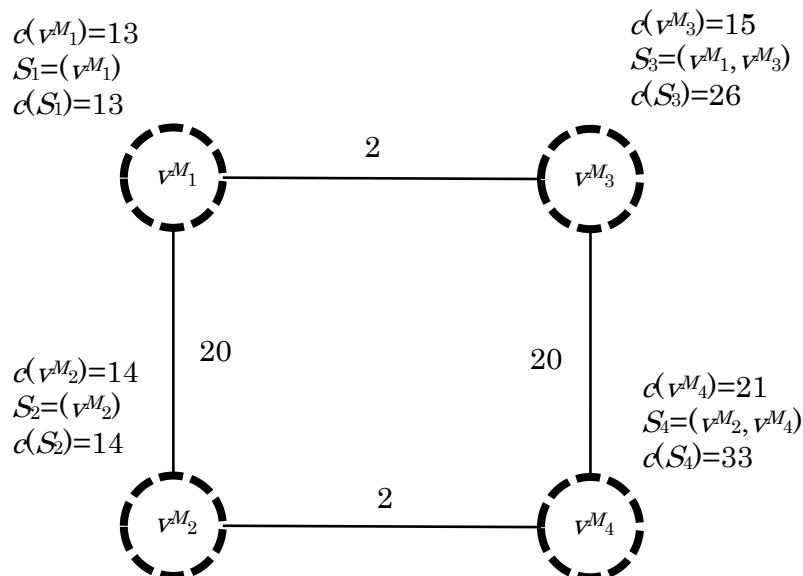


Rys. 4.21 Wynik działania drugiej iteracji PPCB

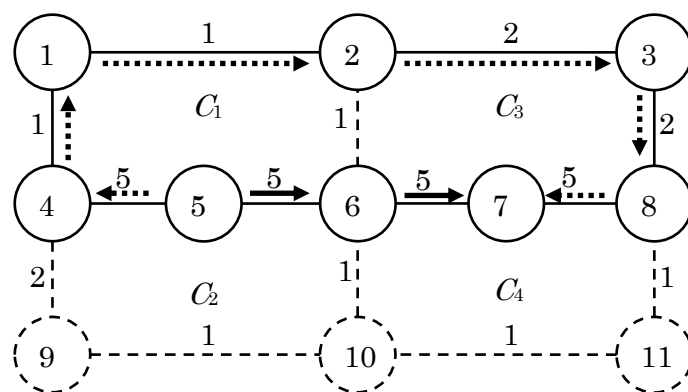


Rys. 4.22 Wynik działania trzeciej iteracji PPCB

W czwartej iteracji w A jest obecny tylko jeden cykl bazowy v^{M_4} . Algorytm usuwa v^{M_4} ze zbioru A i dodaje do zbioru B (wiersz 7). Cykl bazowy v^{M_4} nie ma żadnego sąsiada, który nie jest zawarty w ani w B , ani w S_4 . PPCB rozpoczyna Krok 3. Wynik czwartej iteracji jest ukazany na Rys. 4.23.



Rys. 4.23 Wynik działania czwartej iteracji PPCB



Rys. 4.24 Wykryta przez PPCB para tras

W Kroku 3 PPCB ma dwa docelowe cykle bazowe w zbiorze $B=(v^{M_3}, v^{M_4})$. Wśród nich algorytm wybiera cykl bazowy v^{M_3} o minimalnym koszcie $c(S_3) = 26$ (wiersz 17). PPCB dzieli cykl $S_3 = (v^{M_1}, v^{M_3})$, $e(S_3) = (e_{1,2}, e_{2,3}, e_{3,8}, e_{8,7}, e_{7,6}, e_{6,5}, e_{5,4}, e_{4,1})$ względem węzłów 5 i 6 żądania na dwie trasy $P_1=(5, 6, 7)$ i $P_2=(5, 4, 1, 2, 3, 8, 7)$ (wiersz 18). Algorytm PPCB kończy swoje działanie.

W przypadku, kiedy w Kroku 3 algorytmu zbiór B jest pusty, oznacza to, że PPCB nie wykrył żadnego cyklu, który zawiera wierzchołki źródłowy i docelowy.

Faktycznie, warunek 2 jest sprawdzany w wierszu 9 algorytmu nie w sposób intuicyjny. Logiczne jest bezpośrednio sprawdzić obecność wierzchołków w cyklu. Jednakże PPCB sprawdza hipotetyczną możliwość obecności wierzchołków w cyklach, które zostaną zbudowane na podstawie obecnych cykli w przyszłych iteracjach. Taka możliwość istnieje, jeżeli wierzchołek ma co najmniej 2 incydentne krawędzie topologii sieci. Jeżeli w wyniku połączenia cykli stopień wierzchołka docelowego zostaje zredukowany do równego 1 lub zera, oznacza to, że wierzchołek nie może istnieć w takim cyklu i nie może być wykorzystany dla budowania nowych cykli, ponieważ żadna z operacji wykonanej przez PPCB nie ma możliwości podniesienia stopnia wierzchołka.

Warto zwrócić uwagę, że bezpośrednia nieobecność docelowego wierzchołka w dowolnym cyklu S (jeżeli warunek 2 jest spełniony) nie wyklucza sytuacji, w której cykle uzyskane poprzez połączenie S z innymi cyklami nie będą zawierały wierzchołka docelowego. Algorytm PPCB w Kroku 1 wykrywa cykle zawierające wierzchołek źródłowy (niezależnie od tego czy cykle zawierają wierzchołek docelowy czy nie) i pojedyncze cykle bazowe. W Kroku 2 PPCB iteracyjnie podejmuje próby

budowania nowych cykli poprzez łączenie wcześniej wykrytych cykli S (wykrytych w poprzednich iteracjach i/lub w Kroku 1) z sąsiednimi cyklami bazowymi. Na końcu działania algorytmu każdy cykl bazowy ma wykryty cykl S o najmniejszym koszcie, który zawiera wierzchołek źródłowy (niezależnie od tego czy cykle zawierają wierzchołek docelowy czy nie) i w wyniku usunięcia krawędzi topologii sieci wspólnych dla cykli bazowych zawartych w S stopień docelowego wierzchołka nie został zredukowany do wartości równej 1 lub 0. To twierdzenie jest prawdziwe również i dla cykli bazowych, które zawierają wierzchołek docelowy. Pozwala to w Kroku 3 wybrać spośród takich docelowych cykli bazowych odpowiedni cykl S o najmniejszym koszcie i przekształcić go w parę węzłowo-rozłącznych tras dla pary wierzchołków: źródłowego i docelowego danego żądania.

4.2.2 Aktualizacja kosztów elementów mapy komórkowej

Instalacja trasy w sieciach o ograniczonych zasobach przepustowości powoduje redukcję dostępnej przepustowości łączy które zostały wykorzystane dla instalacji trasy. W doborze tras ukierunkowanym na zrównoważenie obciążenia sieci aktualizujemy koszt łączy sieci po wykryciu tras obsługi każdego żądania. Wykorzystanie metod transformacji grafu topologii sieci powoduje nowe wyzwanie: aktualizację kosztów krawędzi w grafie mapy komórkowej, który powstaje na podstawie grafu topologii sieci.

Jednym z możliwych rozwiązań wydaje się powtórna transformacja grafu topologii sieci do postaci grafu mapy komórkowej po aktualizacji kosztów łączy oraz zastosowanie tej struktury do wyznaczenia tras dla kolejnego żądania. Jednak takie rozwiązanie wymaga powtórnych transformacji po instalacji tras każdego żądania. Z kolei wymaga to dodatkowych obliczeń i nie jest efektywne w przypadku rozwiązań ukierunkowanych na redukcję czasu obliczeń. Drugim rozwiązaniem jest transformacja grafu topologii sieci do struktury, która pozwala w łatwy sposób aktualizować swoje koszty na podstawie zaistniałych zmian w wartościach metryki kosztu łączy sieci bez konieczności budowania takiej struktury na nowo.

Mapa komórkowa jest zaprojektowana w taki sposób, że pozwala aktualizować swoje koszty na podstawie zmian wartości metryki kosztu łączy sieci. W niniejszej metodzie, jeżeli wiadomo, że metryka kosztu łączy sieci $c(e_i)$ ulega zmianie o $\Delta c(e_i)$ to odpowiednie koszty dotyczące mapy komórkowej mają są aktualizowane zgodnie z formułami (4.4) i (4.5).

$$c'(v_j^M) = c(v_j^M) + \Delta c(e_i); \forall v_j^M \in V^M(e_i) \quad (4.4)$$

$$c'(e_j^M) = c(e_j^M) + 2\Delta c(e_i); \forall e_j^M \in E^M(e_i) \quad (4.5)$$

Możliwość szybkiej aktualizacji kosztów pozwala korzystać z mapy komórkowej zawczasu wygenerowanej na podstawie grafu topologii sieci. Taka możliwość czyni mapę komórkową przydatną w doborze tras ukierunkowanym jednocześnie na zrównoważenie obciążenia sieci oraz na redukcję czasu wyznaczania par węzłowo-rozłącznych tras.

4.2.3 Wyniki symulacji

Przeprowadzone symulacje miały na celu porównanie efektywności autorskiego algorytmu PPCB oraz referencyjnej metody Bhandariego. Analizie podlegały następujące parametry:

- prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$,
- średni czas potrzebny na wyznaczenie tras $\overline{t_r(P)}$ dla wszystkich żądań w zbiorze,
- średnia wartość współczynnika wykorzystania zasobów sieci \overline{U} (obliczona zgodnie ze wzorem (3.13)),
- średnia liczba przeskoków w trasach $|\overline{E(P)}|$.

Optyczne sieci szkieletowe charakteryzują się topologią niezmienną w długim czasie. Awaria łącza jest w danym momencie w sensie znaczeniowym równoważna wykorzystaniu 100% jego przepustowości. Podobnie, awaria węzła co do znaczenia odpowiada wykorzystaniu 100% przepustowości wszystkich łączy incydentnych względem tego węzła. Z tego powodu, w przypadku wyznaczania tras tuż po wystąpieniu awarii wystarczy zaktualizować koszty mapy komórkowej zgodnie z formułami (4.4), (4.5) zamiast dokonywania transformacji nowego grafu topologii sieci. Metoda szybkiej aktualizacji mapy komórkowej w przypadku dodania nowych



elementów jest z kolei opisana w Aneksie 10. Wszystko to pozwala zawczasu transformować topologię sieci do postaci mapy komórkowej i aktualizować mapę w przypadku usunięcia lub dodania nowych elementów. Ze względu na to, że wymagana jest pojedyncza transformacja grafu topologii sieci do grafu mapy komórkowej, czas takiej transformacji nie jest wzięty pod uwagę.

Z kolei, czas aktualizacji kosztów mapy komórkowej jest niezbędny dla poprawnego doboru tras w sieciach o ograniczonych zasobach przepustowości. Dlatego wyniki dotyczące pomiaru czasu podczas symulacji obejmują również czas aktualizacji kosztów mapy komórkowej.

Analiza właściwości algorytmów doboru tras niezawodnych została przeprowadzona dla trzech omówionych topologii sieci szkieletowych (USA, Włoskiej, PIONIER) zakładając, że każde łącze każdej z tych sieci oferowało 80 kanałów transmisji. Jednym z założeń było, że każde żądanie instalacji tras dotyczyło rezerwacji przepustowości jednego kanału. Zostały przeanalizowane następujące scenariusze dotyczące stopnia obciążenia sieci:

- 80% - zbiór żądań instalacji tras pomiędzy 80% wszystkich możliwych par węzłów wybranymi losowo;
- 90% - zbiór żądań instalacji tras pomiędzy 90% wszystkich możliwych par węzłów wybranymi losowo;
- 100% - zbiór żądań instalacji tras pomiędzy wszystkimi możliwymi parami węzłów.

Dla każdej analizowanej topologii sieci i rozmiaru zbioru żądań-obciążenia sieci zostało przeprowadzonych 100 symulacji.

Znormalizowane wartości przewagi PPCB nad podejściem referencyjnym są zaprezentowane w Tabeli 4.1. Wyniki symulacji wskazują na znaczną redukcję czasu obliczeń (nawet 73.54%) uzyskaną w przypadku proponowanej metody PPCB kosztem niewielkiego wzrostu prawdopodobieństwa blokowania żądań (maksimum o 3.06%) w porównaniu z rezultatami osiągniętymi dla referencyjnego algorytmu Bhandariego.

PPCB jest algorytmem heurystycznym, a więc suboptymalnym i nie zawsze wykrywa optymalne trasy. Z tego powodu, jak widać w Tabeli 4.1 notuje ujemne

wartości w obszarze przewagi w liczbie przeskoków $|\overline{E(P)}|$ (minimum -0.19%, maksimum -3.17%). Dłuższe trasy doprowadzają do tego, że algorytm PPCB gorzej niż algorytm Bhandariego wykorzystuje zasoby przepustowości sieci \overline{U} , co, z kolei, zwiększa prawdopodobieństwo blokowania żądań. Jednakże, PPCB ma znaczną przewagę nad algorytmem Bhandariego we wszystkich przeanalizowanych scenariuszach pod względem czasu obliczeń co czyni algorytm przydatnym dla doboru tras niezawodnych w przypadkach, w których czas obliczeń ma znaczenie.

Sieć	$\overline{t_A(P)}$ (%)	$\overline{\rho_B}$ (%)	$ \overline{E(P)} $ (%)	\overline{U} (%)
$\Omega = 80\%$				
USA	21.76	-1.33	-2.87	-3.98
Włoska	26.93	-0.02	-0.52	-0.08
Pionier	73.54	-0.21	-0.19	-0.81
$\Omega = 90\%$				
USA	19.71	-2.20	-2.91	-3.04
Włoska	26.10	-0.09	-0.53	-0.60
Pionier	73.51	-0.02	-0.39	-0.76
$\Omega = 100\%$				
USA	16.84	-3.06	-3.17	-2.46
Włoska	26.11	-0.05	-0.54	-0.57
Pionier	73.48	-0.32	-0.59	-0.74

Tabela 4.1 Znormalizowane wartości przewagi algorytmu PPCB nad referencyjnym algorytmem Bhandariego

4.3 Wnioski

Analiza symulacyjna potwierdziła przewagę algorytmu PPCB pod względem czasu obliczeń dla wszystkich topologii sieci i poziomów obciążenia wziętych pod uwagę. Najlepsze wyniki zostały osiągnięte dla sieci Pionier. Topologia sieci Pionier jest typowa dla sieci szkieletowych, które powstały na podstawie sieci o topologii pierścienia (np. SDH).

Metody transformacyjne pozwalają uzyskać przewagę w kontekście jednych parametrów kosztem wartości innych parametrów. Na przykład kompresja dźwięku jest osiągalna za cenę strat jakości lub przy wzroście czasu obliczeń. Zaproponowana

w niniejszym rozdziale metoda osiąga redukcję czasu obliczeń kosztem wzrostu liczby przeskoków wykrytych tras. PPCB jest algorytmem heurystycznym pseudooptymalnym. Niestety nie udało się w sposób analityczny wyznaczyć stopnia tej pseudooptymalności względem rozwiązania optymalnego. Duże rozproszenie wyników symulacji (różnica 16-krotna: od -0.19% do -3.17%) dla liczby przeskoków w trasie wskazują na to, że nie jest to zadanie trywialne. Wykrycie korelacji pomiędzy topologią sieci a wzrostem liczby przeskoków w trasie dla proponowanego algorytmu względem optymalnego rozwiązania jest jednym z kierunków przyszłych badań.

Warto zauważyć, że maksymalny wzrost liczby przeskoków dla proponowanej metody doprowadził do wzrostu prawdopodobieństwa blokowania żądań co najwyżej o 3.06%, lecz jednocześnie obniżył czas obliczeń o 16.84%. W najlepszym przypadku osiągnięta redukcja czasu doboru tras wyniosła 73.54% kosztem wzrostu prawdopodobieństwa blokowania żądań jedynie o 0.21%.

Powyższe rezultaty świadczą o tym, że wykorzystanie metod transformacyjnych w routingu niezawodnym pozwala osiągnąć znaczną redukcję czasu obliczeń. Dalsze badania w tym kierunku mogą pomóc w projektowaniu nowych efektywnych algorytmów doboru tras niezawodnych.



5 Podsumowanie

W niniejszej rozprawie podjęto próbę konstrukcji algorytmów doboru tras niezawodnych, które poprzez zastosowanie autorskich strategii transformacyjnych miały doprowadzić do istotnej redukcji czasu wyznaczania tras, jak i znacząco zredukować stopień blokowania połączeń.

Problem szybkości wyczerpania dostępnych zasobów sieci rozległej (przepustowości łączy) jest niewątpliwie większy w przypadku mechanizmów ochrony transmisji bazujących na wyznaczanych zawczasu trasach zabezpieczających, niż w sytuacji wyznaczania tych tras w sposób reaktywny (tzn. po awarii). Dlatego też w rozdziale 3 zaproponowano strategię doboru tras ukierunkowaną na redukcję prawdopodobieństwa blokowania żądań poprzez dążenie do zrównoważenia obciążenia łączy sieci. Opisana w rozdziale 3 autorska metoda, zainspirowana założeniami znanymi z mechaniki kwantowej uwzględniająca a priori stopień prognozowanych konfliktów o zasoby łączy sieci po ich ewentualnym wykorzystaniu przez aktualnie wyznaczaną trasę, była w stanie uzyskać istotną przewagę w kontekście wartości współczynnika blokowania żądań nawet o 53.6% w porównaniu z algorytmem referencyjnym.

Czas niezbędny do wyznaczenia tras transmisji często wpływa na łączną efektywność mechanizmu przywracania połączeń (np. w scenariuszu, w którym trasy alternatywne nie zostały wyznaczone do momentu wystąpienia awarii lub w sytuacji awarii wielokrotnej stanowiącej sekwencję awarii pojedynczych). W szczególności, jeśli algorytm byłby w stanie wyznaczyć trasy transmisji adekwatne do bieżącego stanu sieci (tj. po awarii) w wystarczająco krótkim czasie (tzn. zanim nastąpi kolejna awaria z sekwencji awarii wielokrotnej), to problem ochrony przed awarią wielokrotną k elementów sieci (z niezbędnymi k trasami zabezpieczającymi dla każdej trasy podstawowej) zostałby sprowadzony do zadania wyznaczenia jedynie pojedynczej trasy zabezpieczającej dla każdej trasy podstawowej, co w znacznie mniejszym stopniu angażowałoby zasoby sieci, redukując tym samym ryzyko blokowania żądań i umożliwiając obsługę większej ich liczby.

Niniejsza rozprawa pokazała, że przezwycięzenie powyższego problemu jest możliwe. W szczególności, jak zostało ukazane w rozdziale 4, dzięki zastosowaniu autorskiej strategii transformacji grafu topologii sieci do postaci tzw. „mapy

komórkowej”, w której wierzchołki reprezentują nie węzły sieci lecz cykle bazowe grafu topologii sieci, z uwagi charakterystyki zaproponowanej strategii oraz ze względu na znacznie mniejszą liczbę wierzchołków mapy komórkowej w porównaniu z liczbą wierzchołków grafu topologii sieci, możliwe było uzyskanie redukcji czasu obliczeń tras transmisji nawet o 73.54% w porównaniu z rozwiązaniem referencyjnym wykorzystującym klasyczny graf topologii sieci.

Teza rozprawy została tym samym udowodniona. Strategie zaprezentowane w rozdziałach 3 i 4 dodatkowo zyskały uznanie kapituły konkursu na najlepszą prezentację w grupie młodych pracowników nauki, podczas dwóch edycji konferencji KSTiT 2016 i KSTiT 2015, podczas których zdobyły I i III miejsce.

Podsumowując, oryginalny dorobek przedstawiony w niniejszej rozprawie obejmuje opracowanie:

- (1) metody doboru tras redukującej wartość współczynnika blokowania żądań nawet o 53.6% w porównaniu z algorytmem referencyjnym zaprezentowanej w rozdziale 3.1 wraz z:
 - (1.1) dowodem poprawności działania algorytmu DFSm wykorzystywanego przez metodę kwantową z rozdziału 3 przedstawionym w Aneksie 1,
 - (1.2) analizą złożoności algorytmicznej metody doboru tras kwantowych z rozdziału 3 zaprezentowaną w Aneksie 2,
 - (1.3) analizą badań efektywności metody kwantowej z rozdziału 3 zaprezentowaną w Aneksie 3,
 - (1.4) analizą doboru właściwych proporcji współczynników α i β dla metody kwantowej z rozdziału 3 zaprezentowaną w Aneksie 4,
 - (1.5) przedstawieniem w Aneksie 5 szczegółowych wyników symulacji dla metody kwantowej z rozdziału 3,
- (2) modyfikacji metody doboru tras z rozdziału 3.1 mającą na celu również redukcję czasu obliczeń kwantów ukazaną w rozdziale 3.2,
- (3) metody doboru tras niezawodnych, która poprzez wykorzystanie struktury mapy komórkowej charakteryzuje się znacznie większą szybkością działania (do 73.54% w porównaniu z algorytmem referencyjnym) ukazanej w rozdziale 4 wraz z:
 - (3.1) analizą złożoności algorytmicznej algorytmu PPCB wykorzystywanego przez metodę z rozdziału 4 przedstawioną w Aneksie 6,
 - (3.2) szczegółowym omówieniem wybranych aspektów funkcjonowania algorytmu PPCB (Aneks 7),
 - (3.3) modyfikacją algorytmu PPCB dla wyznaczania par tras krawędziowo-rozłącznych, dyskusją możliwości zastosowania rozwiązania dla grafów o topologii nieplanarnej, jak i w przypadku konieczności wyznaczenia zbiorów trzech węzłowo-rozłącznych tras (Aneks 8),
 - (3.4) analizą możliwości włączenia kwantowej strategii redukcji współczynnika blokowania żądań z rozdziału 3 do metody wyznaczania

tras bazującej na strukturze mapy komórkowej (rozdział 4) przedstawioną w Aneksie 9,

- (3.5) wskazówkami dotyczącymi rozwiązania problemu szybkiej aktualizacji mapy komórkowej po zmianie topologii sieci dla metody z rozdziału 4 (Aneks 10).

Jako że propozycje rozszerzeń zawarte w Aneksach 8 i 9 zostały w nich przedstawione wyłącznie od strony koncepcyjnej, szczegółowa analiza ich właściwości stanowi materiał potencjalnych przyszłych badań.



Aneks 1 – Udowodnienie poprawności działania algorytmu *DFS_m*

W niniejszym aneksie zostanie wykazane, że algorytm *DFS_m* (używany przez metodę z rozdziału 3) wykrywa wszystkie możliwe trasy w grafie G pomiędzy daną parą węzłów. W tym celu zostaną poniżej zdefiniowane cztery tezy, które zostaną następnie udowodnione.

Teza 1: W wyniku działania wszystkich kopii *DFS_m* na grafie G , które są kopiami drugiej rekurencji algorytmu *DFS_m* uruchomionej na wierzchołku źródłowym v_s , zostaną wykryte wszystkie istniejące w grafie G ścieżki $R(v_s, *)$ o zdefiniowanym źródłowym wierzchołku v_s i długości jednego przeskoku $\forall P \in R(v_s, *), c(P)=1$.

Załóżmy, że w grafie G istnieje ścieżka P o długości jednego przeskoku o zdefiniowanym źródłowym wierzchołku v_s : $c(P(v_s, v_t))=1$, która nie została wykryta w wyniku działania żadnej kopii *DFS_m* będącymi kopiami drugiej rekurencji algorytmu *DFS_m* uruchomionego na wierzchołku v_s . Przypadek ten jest możliwy w sytuacji, gdy został spełniony jeden z następujących warunków:

- (a) kopia *DFS_m* uruchomiona na docelowym wierzchołku v_t nie wykryła ścieżki (nie dodała v_t do wektora trasy P);
- (b) kopia *DFS_m* nie została uruchomiona na wierzchołku docelowym v_t .

Algorytm *DFS_m* został zaprojektowany w taki sposób, że spełnienie Kroku 1 (tj. dodanie wierzchołka, na którym został uruchomiony do wektora trasy) jest bezwarunkowe. Z tego powodu nie jest możliwe spełnienie warunku (a). Jeżeli liczba przeskoków ścieżki jest równa jeden: $c(P(v_s, v_t))=1$, to ścieżka nie zawiera wierzchołków tranzytowych. Oznacza to, że wierzchołek docelowy jest sąsiadem wierzchołka źródłowego: $v_t \in \mathcal{V}(v_s)$. Jednak, wynikiem działania algorytmu *DFS_m* pierwszej rekurencji jest uruchomienie kopii *DFS_m* na każdym wierzchołku sąsiednim względem wierzchołka źródłowego v_s . Oznacza to, że spełnienie warunku (b) nie jest możliwe, ponieważ kopia *DFS_m* zostanie uruchomiona dla $\forall v_i: v_i \in \mathcal{V}(v_s)$, w przypadku których jest wiadome, że $v_t \in \mathcal{V}(v_s)$. To z kolei udowadnia niemożliwość istnienia w grafie G ścieżki o długości jednego przeskoku i o zdefiniowanym źródłowym wierzchołku v_s , która nie zostanie wykryta w wyniku

działania DFSm uruchomionego na wierzchołku v_s . Tym samym **teza 1 zostaje udowodniona**.

Teza 2: W wyniku działania wszystkich kopii DFSm na grafie G , które są kopiami trzeciej rekurencji algorytmu DFSm uruchomionej na wierzchołku źródłowym v_s , zostaną wykryte wszystkie istniejące w grafie G ścieżki $R(v_s, *)$ o zdefiniowanym źródłowym wierzchołku v_s i długości dwóch przeskoków $\forall P \in R(v_s, *)$, $c(P)=2$.

Załóżmy, że w grafie G istnieje ścieżka P o długości dwóch przeskoków o zdefiniowanym źródłowym wierzchołku v_s : $c(P(v_s, v_t))=2$, która nie została wykryta w wyniku działania żadnej kopii DFSm będących kopiami trzeciej rekurencji algorytmu DFSm uruchomionego na wierzchołku v_s . Jest to możliwe gdy został spełniony jeden z opisanych wcześniej (podczas udowodnienia tezy 1) warunków.

Jak zostało opisane powyżej, nie jest możliwe spełnienie warunku (a) ze względu na sposób działania algorytmu. Jeżeli długość ścieżki jest równa dwóm przeskokom: $c(P(v_s, v_t))=2$, to ścieżka zawiera jeden wierzchołek tranzytowy v_x . To z kolei oznacza, że wierzchołek docelowy jest sąsiadem wierzchołka tranzytowego: $v_t \in \mathcal{V}(v_x)$. Wierzchołek tranzytowy jest z kolei sąsiadem wierzchołka źródłowego: $v_x \in \mathcal{V}(v_s)$, tworząc ścieżkę $P(v_s, v_x)=(v_s, v_x)$ o długości jednego przeskoku $c(P(v_s, v_x))=1$. Ta ścieżka zostanie jednak wykryta przez kopię DFSm drugiej rekurencji **zgodnie z udowodnioną tezą 1**. Wiadomo, że w przypadku gdy wierzchołek na którym algorytm DFSm został uruchomiony nie jest wierzchołkiem docelowym, wynikiem działania algorytmu DFSm jest uruchomienie kopii DFSm na każdym wierzchołku sąsiednim względem wierzchołka, na którym kopia DFSm została uruchomiona. Oznacza to, że spełnienie warunku (b) nie jest możliwe, ponieważ kopia DFSm zostanie uruchomiona dla $\forall v_i; v_i \in \mathcal{V}(v_x)$ – gdy jest wiadome, że $v_t \in \mathcal{V}(v_x)$. To z kolei udowadnia niemożliwość istnienia w grafie G ścieżki o długością dwóch przeskoków i o zdefiniowanym źródłowym wierzchołku v_s , która nie zostanie wykryta w wyniku działania DFSm uruchomionego na wierzchołku v_s . Tym samym **teza 2 zostaje udowodniona**.

Teza 3: W wyniku działania wszystkich kopii DFSm na grafie G , które są kopiami k -tej rekurencji (gdzie $k > 3$) algorytmu DFSm uruchomionej na wierzchołku źródłowym v_s , zostaną wykryte wszystkie istniejące w grafie G ścieżki $R(v_s, *)$ o zdefiniowanym źródłowym wierzchołku v_s i długości wynoszącej $k-1$ przeskoków $\forall P \in R(v_s, *)$, $c(P)=k-1$.

Załóżmy, że w grafie G istnieje ścieżka P o długości $k-1$ przeskoków o zdefiniowanym źródłowym wierzchołku v_s : $\alpha(P(v_s, v_t))=k-1$, która nie została wykryta w wyniku działania żadnej kopii DFSm, które są kopiami k -tej rekurencji algorytmu DFSm uruchomionej na wierzchołku v_s . Jest to możliwe jeżeli został spełniony jeden opisanych wcześniej warunków (wskazanych podczas udowodnienia tezy 1 i tezy 2).

Jak zostało opisane powyżej, nie jest możliwe spełnienie warunku (a) ze względu na sposób działania algorytmu. Jeżeli długość ścieżki jest równa $k-1$ przeskokom: $\alpha(P(v_s, v_t))=k-1$, to ścieżka zawiera $k-3$ wierzchołki tranzytowe. Niech v_x jest wierzchołkiem poprzednikiem wierzchołka docelowego w ścieżce $P(v_s, v_t)=(v_s, \dots, v_x, v_t)$. Załóżmy, że na wierzchołku v_x została uruchomiona kopia DFSm, która jest kopią $(k-1)$ -rekurencji algorytmu DFSm uruchomionego na wierzchołku źródłowym v_s .

W takim razie, wynikiem działania kopii DFSm $(k-1)$ -rekurencji uruchomionej na wierzchołku v_x jest uruchomienie kopii DFSm k -tej rekurencji na $\forall v_t; v_t \in \mathcal{V}(v_x)$. Pamiętając o tym, że wierzchołek docelowy v_t jest sąsiadem wierzchołka v_x ($v_t \in \mathcal{V}(v_x)$), możemy wnioskować że kopia DFSm k -tej rekurencji zostanie uruchomiona na wierzchołku docelowym v_t .

To z kolei oznacza, że spełnienie warunku (b) nie jest możliwe, jeżeli na wierzchołku v_x została uruchomiona kopia DFSm, która jest kopią $(k-1)$ -rekurencji algorytmu DFSm uruchomionego na wierzchołku źródłowym v_s , gdzie v_x jest wierzchołkiem poprzednikiem wierzchołka docelowego w ścieżce $P(v_s, v_t)=(v_s, \dots, v_x, v_t)$.

Następujące założenie pozwala udowodnić tezę 3 dla $k=4$: Jeżeli w grafie G istnieje ścieżka P o długości trzech przeskoków (dla $k=4$; $k-1=3$) o zdefiniowanym źródłowym wierzchołku v_s : $\alpha(P(v_s, v_t))=3$, to ta ścieżka zostanie wykryta jeżeli na wierzchołku v_x należącym do trasy $P(v_s, v_t)=(v_s, \dots, v_x, v_t)$ zostanie uruchomiona kopia DFSm trzeciej rekurencji. Jeżeli ścieżka $P(v_s, v_t)$ zawiera trzy przeskoki: $\alpha(P(v_s, v_t))=3$, to ścieżka od źródłowego wierzchołka v_s do ostatniego tranzytowego wierzchołka w trasie $P(v_s, v_t)=(v_s, \dots, v_x, v_t)$ zawiera dwa przeskoki $\alpha(P(v_s, v_x))=2$.

Zgodnie z udowodnioną wcześniej tezą 2: w wyniku działania wszystkich kopii DFSm na grafie G , które są kopiami trzeciej rekurencji algorytmu DFSm uruchomionej na wierzchołku źródłowym v_s , zostaną wykryte wszystkie istniejące w

grafie G ścieżki $R(v_s, *)$ o zdefiniowanym źródłowym wierzchołku v_s i długości wynoszącej dwa przeskoki. Oznacza to, że ścieżka $P(v_s, v_x)$ zostanie wykryta poprzez kopię DFSm trzeciej rekurencji. Ten fakt, że v_x nie jest wierzchołkiem docelowym powoduje uruchomienie kopii DFSm czwartej rekurencji dla $\forall v_t: v_t \in \mathcal{V}(v_x)$, gdzie $v_t \in \mathcal{V}(v_x)$.

Z kolei oznacza to, że na v_t zostanie uruchomiana kopia DFSm czwartej rekurencji i trasa $P(v_s, v_t)$ o trzech przeskokach zostanie wykryta (**nie jest możliwe spełnienie warunku (b)**).

W analogiczny sposób możemy udowodnić **tezę 3 dla $k=5$** . Ponieważ **teza 3 dla $k=4$** została udowodniona powyżej, pozwala to udowodnić **tezę 3 dla $k=6$** i tak dalej dla **jakiegokolwiek kolejnej wartości k** .

Teza 4: W wyniku działania wszystkich kopii DFSm algorytmu uruchomionego na wierzchołku źródłowym v_s , zostaną wykryte wszystkie istniejące w grafie G ścieżki $R(v_s, v_t)$ o zdefiniowanym źródłowym wierzchołku v_s i zdefiniowanym docelowym wierzchołku v_t .

Jeżeli algorytm DFSm został uruchomiony na źródłowym wierzchołku v_s to **zgodnie z tezą 1** jego kopie wykryją wszystkie ścieżki w grafie G długości jednego przeskoku: $\forall P \in R(v_s, *)$, $\mathcal{L}(P)=1$. Również **zgodnie z tezą 2** jego kopie wykryją wszystkie ścieżki o dwóch przeskokach: $\forall P \in R(v_s, *)$, $\mathcal{L}(P)=2$. **Zgodnie z tezą 3** jego kopie wykryją wszystkie ścieżki o długości $k-1$ przeskoków: $\forall P \in R(v_s, *)$, $\mathcal{L}(P)=k-1$, $3 < k < |\mathcal{V}(G)|$.

To z kolei oznacza, że zostaną wykryte wszystkie istniejące ścieżki w grafie G o długości $0 < \mathcal{L}(P) < |\mathcal{V}(G)|$ o zdefiniowanym źródłowym wierzchołku v_s . Pamiętając o tym, że nie jest możliwe istnienie w sieci trasy pomiędzy dwoma węzłami o:

- ujemnej liczbie przeskoków,
- zerowej liczbie przeskoków,
- liczbie przeskoków większej niż $|\mathcal{V}(G)|-1$,

możemy wnioskować, że algorytm DFSm uruchomiony na wierzchołku źródłowym v_s wykrywa zbiór wszystkich ścieżek istniejących w grafie G o zdefiniowanym wierzchołku źródłowym v_s : $R(v_s, *)$. Logiczne jest, że zbiór wszystkich tras o zdefiniowanych wierzchołkach źródłowym v_s oraz docelowym v_t jest podzbiorem



wykrytego poprzez kopie algorytmu DFSm zbioru: $R(v_s, v_t) \subset R(v_s, *)$. Algorytm działa w taki sposób, że po uruchomieniu swojej kopii na wierzchołku docelowym doda wykrytą trasę do zbioru Z (Krok 2.1).

Z tego wynika że: w wyniku działania wszystkich kopii DFSm algorytmu uruchomionego na wierzchołku źródłowym v_s , zostaną wykryte wszystkie istniejące w grafie G ścieżki o zdefiniowanym źródłowym wierzchołku v_s i zdefiniowanym docelowym wierzchołku v_t : $R(v_s, v_t)$, które zostaną przechowane w zbiorze Z , co **dowodzi prawdziwości tezy 4**. Udowodnienie tezy 4 z kolei kończy dowodzenie poprawnego działania algorytmu DFSm.



Aneks 2 – Złożoność algorytmiczna metody doboru tras kwantowych

W niniejszym aneksie została przebadana algorytmiczna złożoność metody doboru tras kwantowych opisanej w rozdziale 3.

Pod względem ilości wymaganych zasobów możemy podzielić metodę inspirowaną mechaniką kwantową na dwie części:

1. Procedura statyczna – wykonywana jedynie dla podanej topologii sieci.
2. Procedura dynamiczna – wykonywana za każdym razem w celu obsługi nadchodzących żądań instalacji tras w sieci.

Procedura statyczna ma na celu zbudować zbiór wszystkich kwantów sieci Q . Mówiąc inaczej, procedura uruchamia algorytm budowania kwantu dla każdej pary węzłów sieci. Z tego wynika, że złożoność obliczeniowa procedury statycznej wynosi:

$$f_1 = \frac{|V|(|V| - 1)}{2} \times f_{ABK} \quad (5.1)$$

gdzie:

f_{ABK} – Złożoność algorytmiczna **A**lgorytmu **B**udowania **K**wantu.

Na złożoność algorytmiczną algorytmu budowania kwantu składa się złożoność algorytmiczna każdego z pięciu kroków algorytmu zgodnie ze wzorem (5.2).

W Kroku 1 algorytm budowania kwantów wykonuje tyle iteracji ile wynosi liczba łączy $|E|$ sieci. Maksymalna liczba łączy sieci w najgorszym przypadku jest równa liczbie par węzłów w sieci. Tyle samo iteracji algorytm wykona w Kroku 4 (wzór (5.3)).

Liczba iteracji w Kroku 2 zależy od złożoności algorytmicznej algorytmu DFSm (wzór (5.4)). Liczba iteracji w Kroku 3 zależy od liczby łączy zbioru E zawartych we wszystkich trasach P w zbiorze Z (wzór 5.5).

$$f_{ABK} = f_{ABK(1)} + f_{ABK(2)} + f_{ABK(3)} + f_{ABK(4)} + f_{ABK(5)} \quad (5.2)$$

gdzie:

- $f_{ABK(1)}$ – złożoność algorytmiczna Kroku 1 algorytmu budowania kwantu;
- $f_{ABK(2)}$ – złożoność algorytmiczna Kroku 2 algorytmu budowania kwantu;
- $f_{ABK(3)}$ – złożoność algorytmiczna Kroku 3 algorytmu budowania kwantu;
- $f_{ABK(4)}$ – złożoność algorytmiczna Kroku 4 algorytmu budowania kwantu;
- $f_{ABK(5)}$ – złożoność algorytmiczna Kroku 5 algorytmu budowania kwantu.

$$f_{ABK(1)} = f_{ABK(4)} = \frac{|V|(|V| - 1)}{2} \quad (5.3)$$

$$f_{ABK(2)} = f_{DFSm} \quad (5.4)$$

$$f_{ABK(3)} = \sum_{i=1}^{i=|Z|} |E(P_i(Z))| \quad (5.5)$$

Liczba iteracji w Kroku 5 jest niezmienna i nie zależy ani od liczby elementów sieci, ani od topologii sieci.

$$f_{ABK(5)} = const \quad (5.6)$$

Logiczne jest, że sumaryczna liczba łączy zawartych we wszystkich trasach w najgorszym przypadku jest równa liczbie iteracji algorytmu DFSm. Z tego wynika, że liczba operacji w Kroku 3 dla dowolnego kwantu w najgorszym przypadku jest zależna od liczby operacji algorytmu DFSm opisanej wyrażeniem (5.7).

$$f_{ABK(3)} = f_{DFS_m} \quad (5.7)$$

W takim razie liczba operacji algorytmu budowania kwantu przyjmuje postać:

$$f_{ABK} = |V|(|V| - 1) + 2 \times f_{DFS_m} \quad (5.8)$$

Warto zauważyć, że działanie algorytmu DFS_m na grafie o dwóch wierzchołkach wymaga dwóch uruchomień DFS_m. DFS_m rozpoczyna działanie na wierzchołku źródłowym i uruchamia rekurencyjnie kopię dla wierzchołka docelowego. Dla grafu o trzech wierzchołkach, DFS_m również rozpoczyna swoją pracę z wierzchołka źródłowego i w najgorszym przypadku uruchamia dwie kopie DFS_m dla pozostałych dwóch wierzchołków. Każda kopia DFS_m z kolei wymaga w najgorszym przypadku uruchomienia jeszcze jednej kopii DFS_m. Warto pamiętać, że nie jest możliwe uruchomienie nowej kopii DFS_m na wierzchołkach zawartych w lokalnych kopiach wektora P . Dlatego maksymalna możliwa liczba kopii DFS_m, która może być uruchomiona dla dowolnego wierzchołka, jest równa $|V| - |P|$. W takim razie, algorytmiczna złożoność obliczeniowa najgorszego przypadku algorytmu DFS_m dla grafu o $|V|$ wierzchołkach wynosi:

$$f_{DFS_m} = 1 + (|V| - 1) \times (|V| - 2) \times (|V| - 3) \times \dots \times 1 \quad (5.9)$$

Zapisana w innej postaci algorytmiczna złożoność obliczeniowa najgorszego przypadku wygląda w następujący sposób:

$$f_{DFS_m} = 1 + (|V| - 1)! \quad (5.10)$$

Z tego wynika, że algorytmiczna złożoność obliczeniowa najgorszego przypadku dla budowania kwantu jest równa:

$$f_{ABK} = |V|(|V| - 1) + 2 \times (1 + (|V| - 1)!) = 2 - |V| + |V|^2 + 2 \times (|V| - 1)! \quad (5.11)$$

Podsumowując powyższe rozważania, możemy zdefiniować algorytmiczną złożoność obliczeniową najgorszego przypadku dla budowania zbioru wszystkich kwantów Q dla dowolnej sieci o liczbie węzłów $|V|$ poprzez wzór (5.12).



$$\begin{aligned}
 f_1 &= \frac{|V|(|V| - 1)(2 - |V| + |V|^2 + 2 \times (|V| - 1)!)}{2} = \\
 &= |V| \cdot |V|! - |V|! + \frac{|V|^4}{2} - |V|^3 + 1 \frac{|V|^2}{2} - |V|
 \end{aligned}
 \tag{5.12}$$

Algorytm budowania kwantu wymaga rezerwacji zasobów pamięci dla przechowywania informacji o dwóch węzłach (węzle źródłowym i docelowym), pamięci dla przechowywania macierzy sąsiedztwa, wektora wszystkich łączy w sieci, jak i zbioru wszystkich kwantów. Algorytm powinien również mieć dostępną pamięć w wymiarze potrzebnym dla poprawnej pracy algorytmu DFSm. Z rozważań tych wynika, że algorytmiczna złożoność pamięciowa metody statycznej może być obliczona w następujący sposób:

$$n_1 = 2 + n(M) + n(E) + n(Q) + n(DFS_{m_{max}}) \tag{5.13}$$

gdzie:

- $n(M)$ – liczba elementów pamięci wymaganych dla przechowywania macierzy sąsiedztwa M ;
- $n(E)$ – liczba elementów pamięci wymaganych dla przechowywania wektora łączy sieci;
- $n(Q)$ – liczba elementów pamięci wymaganych dla przechowywania wszystkich kwantów;
- $n(DFS_{m_{max}})$ – liczba elementów pamięci wymaganych dla przechowywania lokalnych danych przy maksymalnej liczbie kopii algorytmu DFSm uruchomionych jednocześnie.

Liczba elementów wymagana dla przechowywania macierzy sąsiedztwa wynosi $|V|^2$. Liczba elementów wymagana dla przechowywania wektora łączy sieci jest równa liczbie łączy sieci, która z kolei jest równa liczbie par węzłów w sieci $|V|(|V|-1)/2$. Liczba elementów wymagana dla przechowywania wszystkich kwantów w sieci wyrażona wzorem (5.14).

$$n(Q) = |Q| \times n(q) \quad (5.14)$$

gdzie:

$n(q)$ – liczba elementów pamięci wymaganych dla przechowywania pojedynczego kwantu q .

Liczba wszystkich kwantów w sieci jest równa liczbie par węzłów w sieci $|V|(|V|-1)/2$. Każdy kwant zawiera dwa elementy w pamięci dla przechowywania pary węzłów $v_1(q)$, $v_2(q)$ oraz przechowywania wektora $W(q)$.

$$n(q) = 2 + n(W(q)) \quad (5.15)$$

Rozmiar wektora $W(q)$ odpowiada liczbie łączy sieci, która z kolei jest równa liczbie par węzłów w sieci $|V|(|V|-1)/2$. Z tego wynika, że rozmiar pamięci wymaganej do przechowywania wszystkich kwantów w sieci może być obliczony w następujący sposób:

$$n(Q) = \frac{|V|(|V|-1)}{2} \times \frac{|V|(|V|-1)}{2} = \frac{|V|^2(|V|-1)^2}{4} \quad (5.16)$$

Wszystkie kopie DFSm będą uruchomione po kolei. Każda kopia DFSm operuje następującymi danymi:

- M – macierz sąsiedztwa;
- v_s – wierzchołek źródłowy;
- v_t – wierzchołek docelowy;
- Z – zbiór wszystkich tras;
- P – wykryta trasa.

Warto zwrócić uwagę, że macierz sąsiedztwa oraz zbiór wszystkich tras są wspólnymi dla wszystkich kopii DFSm. To pozwala w ramach implementacji algorytmu nie tworzyć kopii całych macierzy M i zbioru Z dla każdej kopii DFSm, a tylko kopii wskaźników ln_m oraz ln_z na tamte struktury. W ogólnym przypadku maksymalna ilość pamięci wymaganej dla wykrycia wszystkich możliwych tras w sieci dla pary węzłów poprzez algorytm DFSm (pamięciowa złożoność algorytmiczna) jest opisana wzorem (5.17).

$$n(DFS_{max}) = n(M) + n(Z) + n(DFS) \times |DFS| \quad (5.17)$$

gdzie:

$n(Z)$ – liczba elementów pamięci wymaganych dla przechowywania wszystkich tras;

$n(DFS)$ – liczba elementów pamięci wymaganych dla przechowywania lokalnych danych na potrzeby algorytmu DFS;

$|DFS|$ – liczba kopii algorytmu DFS uruchomionych jednocześnie.

Maksymalna liczba kopii algorytmu DFS, która może być uruchomiona dla sieci jest równa liczbie węzłów $|V|$ w tej sieci. Liczba elementów wymagana dla przechowywania macierzy sąsiedztwa jest równa $|V|^2$. Najgorszym przypadkiem pod względem pamięciowej złożoności dla przechowywania zbioru wszystkich możliwych tras jest graf o $|V|$ wierzchołkach, w którym każdy wierzchołek ma połączenie z wszystkimi pozostałymi wierzchołkami grafu. W takim razie, maksymalnie długie trasy zawierają wszystkie wierzchołki grafu. Każda z takich tras zaczyna się z wierzchołka źródłowego i kończy się wierzchołkiem docelowym. Z tego wynika, że są dostępne $|V|-2$ pozycje w najdłuższej trasie, które mogą zawierać jakikolwiek z dostępnych wierzchołków. Liczba wierzchołków, które mogą zajmować dostępne pozycje również wynosi $|V|-2$, ponieważ wierzchołki źródłowy i docelowy są zawarte w trasie. Pamiętając o tym, że nie jest możliwe powtarzanie wierzchołków w trasie, możemy określić hipotetyczną maksymalną liczbę tras zawierających $|V|$ węzłów, która jest liczbą możliwych permutacji [73] $|V|-2$ elementów (wierzchołków) i wynosi:

$$|(c(P) = |V| - 1)|_{max} = (|V| - 2)! \quad (5.18)$$

W takim razie, maksymalna liczba tras zawierających $|V|-1$ węzłów jest równa liczbie $(|V|-3)$ -elementowych wariacji [73] bez powtórzeń z $|V|-2$ elementowego zbioru:

$$|(c(P) = |V| - 2)|_{max} = (|V| - 2) \times (|V| - 3) \times \dots \times 2 \quad (5.19)$$



W analogiczny sposób możemy zdefiniować maksymalną liczbę wszystkich możliwych tras zawierających dowolną liczbę węzłów $|V|-i$ (oprócz tras o maksymalnej długości i tras zawierających tylko dwa elementy):

$$|(c(P) = |V| - i - 1)|_{max} = (|V| - 2) \times (|V| - 3) \times \dots \times (i + 1) \quad (5.20)$$

lub inaczej:

$$|(c(P) = |V| - i - 1)|_{max} = \frac{(|V| - 2)!}{i!} \quad (5.21)$$

Pamiętając o tym, że w takim grafie (każdy wierzchołek ma połączenie z każdym) jest obecna jedna trasa zawierająca dwa elementy (wierzchołek źródłowy i docelowy), możemy zdefiniować maksymalną liczbę elementów wymaganych dla przechowywania wszystkich tras w najgorszym przypadku:

$$n(Z) = 2 + |V| \cdot (|V| - 2)! + \sum_{i=2}^{|V|-2} (|V| - i + 1) \frac{(|V| - 2)!}{i!} \quad (5.22)$$

lub, zwracając uwagę na to, że:

$$(|V| - i + 1) \frac{(|V| - 2)!}{i!} = |V| \cdot (|V| - 2)!; i = 1 \quad (5.23)$$

wzór (5.22) przyjmuje postać:

$$n(Z) = 2 + \sum_{i=1}^{|V|-2} (|V| - i + 1) \frac{(|V| - 2)!}{i!} \quad (5.24)$$

Każda kopia algorytmu DFSm przechowuje następujące lokalne dane:

- wskaźnik na macierz sąsiedztwa (rozmiar: jeden element);
- indeks wierzchołka źródłowego (rozmiar: jeden element);
- indeks wierzchołka docelowego (rozmiar: jeden element);
- wskaźnik na zbiór wykrytych tras (rozmiar: jeden element);
- wektor wykrytej trasy.

Z powyższej analizy wynika, że dowolna kopia DFSm przechowuje następującą liczbę elementów:

$$n(DFS_m) = 4 + n(P) \quad (5.25)$$

gdzie:

$n(P)$ – liczba elementów wymagana dla przechowywania wektora wykrytej trasy P ;

Pierwsze wywołanie algorytmu DFSm, która jest uruchomione na wierzchołku źródłowym zawiera tylko jeden element w trasie. Pierwsza rekurencja DFSm zawiera dwa elementy w trasie, druga rekurencja trzy elementy itd. do maksymalnej liczby elementów w trasie, które nie może być większa niż $|V|$. Z tego wynika, że maksymalna liczba elementów niezbędnych w celu przechowywania lokalnych danych przez wszystkie wywołania algorytmu DFSm działające jednocześnie jest sumą wartości ciągu arytmetycznego od $4+1$ do $4+|V|$ i może być zdefiniowana jako:

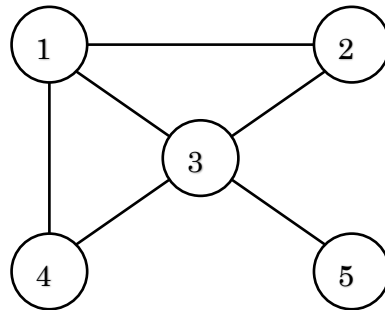
$$\begin{aligned} n(DFS_m) \times |DFS_m| &= \frac{1 + (4 + |V|)}{2} (4 + |V|) - \frac{(1 + 4)}{2} 4 = \frac{|V|^2 + 9|V| + 20}{2} - 10 \\ &= \frac{|V|(|V| + 9)}{2} \end{aligned} \quad (5.26)$$

Pamiętając o tym, że maksymalny rozmiar macierzy sąsiedztwa jest równy $|V|^2$ i bazując na wzorach (5.24) oraz (5.26), możemy przekształcić wzór (5.17) i zdefiniować pamięciową złożoność algorytmiczną najgorszego przypadku dla budowania trasy kwantowej jako:

$$n_1 = |V|^2 + 2 + \frac{|V|(|V| + 9)}{2} + \sum_{i=1}^{|V|-2} (|V| - i + 1) \frac{(|V| - 2)!}{i} \quad (5.27)$$

Warto zwrócić uwagę, że złożoność algorytmiczna (5.12) oraz pamięciowa (5.27) zostały powyżej poddane analizie dla najgorszego przypadku. Jednakże w rzeczywistości nie są spotykane sieci transportowe optyczne o topologii „każdy z każdym”. Zdefiniowanie średniej złożoności algorytmicznej dla niniejszej metody nie jest możliwe w standardowy sposób. Jak widać ze wzorów (5.1), (5.12), (5.17) i (5.27) największy wpływ na złożoność procedury statycznej ma złożoność algorytmu DFSm. Liczba iteracji algorytmu DFSm istotnie zależy od topologii sieci i

wybranej pary węzłów. Na przykład, na Rys. 5.1 jest zaprezentowana topologia sieci o 5 węzłach dla której zależnie od wybranej pary węzłów liczba iteracji algorytmu DFSm się różni nawet 5.5-krotnie.



Rys. 5.1 Przykład topologii sieci

W celu zbudowania kwantu $v_1(q)=5$, $v_2(q)=3$ są niezbędne dwie iteracje DFSm dotyczące odpowiednio wywołania algorytmu DFSm na węźle 5 i uruchomienia rekurencji DFSm na węźle 3.

Z kolei, w celu zbudowania kwantu $v_1(q)=2$, $v_2(q)=4$ wymaganych jest 11 iteracji:

- | | |
|------------------------|-----------|
| 1) DFSm: | (2) |
| 2) DFSm-1 rekurencji: | (2,3) |
| 3) DFSm-1 rekurencji: | (2,1) |
| 4) DFSm-2 rekurencji: | (2,3,5) |
| 5) DFSm-2 rekurencji: | (2,3,4) |
| 6) DFSm-2 rekurencji: | (2,3,1) |
| 7) DFSm-2 rekurencji: | (2,1,3) |
| 8) DFSm-2 rekurencji: | (2,1,4) |
| 9) DFSm-3 rekurencji: | (2,3,1,4) |
| 10) DFSm-3 rekurencji: | (2,1,3,4) |
| 11) DFSm-3 rekurencji: | (2,1,3,5) |

Zaprezentowany powyżej przykład dowodzi, że liczba iteracji DFSm zależy nie tylko od liczby węzłów sieci ale również od topologii sieci i konkretnych par węzłów. Dodatkowo, zależy również od tego, który węzeł z danej pary węzłów został wybrany jako węzeł źródłowy. Wystarczy zmienić kwant $v_1(q)=5$, $v_2(q)=3$ na $v_1(q)=3$, $v_2(q)=5$ (jest to ta sama para węzłów sieciowych), a liczba iteracji DFSm wzrasta

z 2 do 5. Z tego wynika, że określenie stopnia korelacji pomiędzy złożonością DFSm i liczbą węzłów w sieci nie jest możliwe.

Tak czy inaczej, nie jest dobrym podejściem opieranie się na algorytmicznej złożoności najgorszego przypadku dla analizy przydatności nowej metody dla konkretnej sieci. Z tego powodu w niniejszym aneksie został zaprezentowany autorski sposób wyznaczenia oczekiwanej złożoności algorytmu DFSm dla dowolnej sieci w oparciu o metody probabilistyczne. Strategia ta nie jest przydatna dla analizy złożoności dla pojedynczej pary węzłów, jednak ma możliwość zastosowania dla analizy oczekiwanej złożoności algorytmu DFSm uruchomionego dla każdej pary węzłów.

Niniejsza analiza jest prowadzona dla sieci o $|V|$ węzłach i $|E|$ łączach. Wtedy każda instancja DFSm, która jest uruchomiona na dowolnym węźle sieciowym v_i ma trzy możliwości działania:

- zakończyć swoje działanie z powodu tego, że v_i jest węzłem docelowym v_t . ($v_i = v_t$);
- uruchomić kopię DFSm na każdym sąsiednim węźle sieciowym, jeżeli jest on zawarty w wykrytej trasie $P(v_i \neq v_t), (|\mathcal{V}(v_i)|/P) > 0$;
- zakończyć swoje działanie, gdy wszystkie sąsiednie węzły sieciowe są zawarte w trasie $P(v_i \neq v_t), (|\mathcal{V}(v_i)|/P) = 0$.

Prawdopodobieństwo tego, że węzeł v_i nie jest docelowym można wyznaczyć w następujący sposób:

$$p(v_i \neq v_t) = \begin{cases} 1; & v_i = v_s \\ 1 - p(v_i = v_t | v_i \neq v_s); & v_i \neq v_s \end{cases} \quad (5.28)$$

Wiadomo, że wśród $|V|$ węzłów sieci jest tylko jeden węzeł docelowy. Jednocześnie wiadomo również, że każdy z węzłów zawartych w P został wcześniej zbadany i nie jest węzłem docelowym. Z tego wynika, że w dowolnym momencie w sieci są węzły w liczbie $|V| - |\mathcal{V}(P)|$, z których jeden jest węzłem docelowym. Z tego powodu prawdopodobieństwo tego że v_i jest węzłem docelowym możemy określić za pomocą wzoru (5.29).

$$p(v_i = v_t | v_i \neq v_s) = \frac{1}{|V| - |V(P)|} \quad (5.29)$$

Wzór (5.28) może więc ulec przekształceniu do postaci:

$$p(v_i \neq v_t) = \begin{cases} 1; & v_i = v_s \\ 1 - \frac{1}{|V| - |V(P)|}; & v_i \neq v_s \end{cases} \quad (5.30)$$

Pod względem statystyki oczekiwana liczba węzłów sąsiednich dla węzła v_i wynosi $|E|/|V|$. Jednak instancje DFSm zostaną uruchomione tylko i wyłącznie dla tych węzłów sąsiednich, które nie są zawarte w P . Również wiadomo, że dla przypadku ($v_i \neq v_s$) co najmniej jeden z węzłów sąsiednich jest na pewno zawarty w P . W oparciu o ten fakt możemy określić oczekiwaną liczbę wywołań DFSm w następnej ($i+1$) iteracji, które zostaną uruchomione na węzłach $\forall v_{i+1} \in \mathcal{V}(v_i)$ sąsiednich względem węzła v_i jako:

$$|DFS_{i+1}| = \begin{cases} \frac{|E|}{|V|}; & v_i = v_s \\ |DFS_i; (v_i \neq v_t)| \times \left(\frac{|E|}{|V|} - 1\right) \times p(v_{i+1} \notin P); & v_i \neq v_s \end{cases} \quad (5.31)$$

gdzie:

$|DFS_i; (v_i \neq v_t)|$ – liczba wywołań DFSm i -tej rekurencji uruchomionych nie na węźle docelowym (jest opisana poniżej);

Ogólna liczba węzłów w sieci, które mogą zostać wybrane jako węzły sąsiednie węzła v_i jest równa $|V|-2$. Ponieważ nie jest możliwe uruchomienie kopii DFSm na dwóch wierzchołkach: v_i oraz na poprzedniku v_i (łącze prowadzące do poprzednika jest usunięte ze zbioru E we wzorze (5.31)). Nie biorąc pod uwagę braku możliwości przejścia do poprzednika, prawdopodobieństwo tego że węzeł v_{i+1} nie jest zawarty w P jest równe:

$$p(v_{i+1} \notin P) = \begin{cases} 1; & v_i = v_s \\ 1 - \frac{|V(P)| - 1}{|V| - 2}; & v_i \neq v_s \end{cases} \quad (5.32)$$

Bazując na wzorach (5.31) i (5.32), możemy określić oczekiwaną liczbę wywołań DFSm następnej ($i+1$) rekurencji uruchomionych na węzłach sąsiednich względem węzła v_i jako:

$$|DFS_{m_{i+1}}| = \begin{cases} \frac{|E|}{|V|}; & v_i = v_s \\ |DFS_{m_i}; (v_i \neq v_t)| \times \left(\frac{|E|}{|V|} - 1\right) \times \left(1 - \frac{|V(P)| - 1}{|V| - 2}\right); & v_i \neq v_s \end{cases} \quad (5.33)$$

Po uruchomieniu algorytmu DFSm na wybranym węźle źródłowym zostanie uruchomiony jeden egzemplarz kodu algorytmu. Oczekiwana najbardziej prawdopodobna liczba wywołań DFSm pierwszej rekurencji wynosi:

$$|DFS_{m_1}| = \frac{|E|}{|V|} \quad (5.34)$$

Z kolei, każda kopia może być uruchomiona na węźle docelowym z prawdopodobieństwem określonym wzorem (5.29). Instancje DFSm, które nie zostały uruchomione na węźle docelowym mogą wywołać iteracje drugiej rekurencji. Oczekiwana liczba wywołań DFSm pierwszej rekurencji, które zostały uruchomione nie na węźle docelowym wynosi:

$$|DFS_{m_1}; (v_x \neq v_t)| = |DFS_{m_1}| \times \left(1 - \frac{1}{|V| - |V(P)|}\right) \quad (5.35)$$

Wiadomo, że liczba węzłów zawartych w P przed uruchomieniem instancji algorytmu DFSm pierwszej rekurencji jest równa $|V(P)|=1$, ponieważ trasa zawiera tylko węzeł docelowy. W tym przypadku zachodzi następująca równość:

$$|DFS_{m_1}; (v_x \neq v_t)| = |DFS_{m_1}| \times \left(1 - \frac{1}{|V| - 1}\right) \quad (5.36)$$

W analogiczny sposób możemy wyznaczyć najbardziej oczekiwaną liczbę wywołań algorytmu DFSm i -tej rekurencji, które zostały uruchomione nie na węźle docelowym dysponując najbardziej oczekiwaną wartością liczby wywołań algorytmu DFSm i -tej rekurencji (wzór (5.37)).

$$|DFSm_i; (v_x \neq v_t)| = |DFSm_i| \times \left(1 - \frac{1}{|V| - |V(P)|}\right) \quad (5.37)$$

Zwracając uwagę na fakt, że liczba węzłów zawartych w P przed uruchomieniem algorytmu DFSm i -tej rekurencji jest równa $|V(P)|=i$, możemy przekształcić wzór (5.37) w:

$$|DFSm_i; (v_x \neq v_t)| = |DFSm_i| \times \left(1 - \frac{1}{|V| - i}\right) \quad (5.38)$$

Oczekiwana liczba wywołań DFSm drugiej rekurencji zgodnie ze wzorem (5.33) wynosi:

$$|DFSm_2| = |DFSm_1; (v_x \neq v_t)| \times \left(\frac{|E|}{|V|} - 1\right) \times \left(1 - \frac{|V(P)| - 1}{|V| - 2}\right) \quad (5.39)$$

Wiadomo, że liczba węzłów zawartych w P przed uruchomieniem algorytmu DFSm pierwszej rekurencji jest równa $|V(P)|=1$, ponieważ trasa zawiera tylko węzeł docelowy. W tym przypadku wzór (5.37) przyjmuje postać:

$$|DFSm_2| = |DFSm_1; (v_x \neq v_t)| \times \left(\frac{|E|}{|V|} - 1\right) \times \left(1 - \frac{1-1}{|V|-2}\right) \quad (5.40)$$

Analogicznie, pamiętając o tym, że liczba węzłów w P przed uruchomieniem algorytmu DFSm i -tej rekurencji jest równa $|V(P)|=i$, możemy obliczyć najbardziej oczekiwaną liczbę kopii $(i+1)$ -tej rekurencji dysponując najbardziej oczekiwaną liczbą wywołań algorytmu DFSm i -tej rekurencji, które zostały uruchomione nie na węźle docelowym:

$$|DFSm_{i+1}| = |DFSm_i; (v_x \neq v_t)| \times \left(\frac{|E|}{|V|} - 1\right) \times \left(1 - \frac{i-1}{|V|-2}\right) \quad (5.41)$$

Liczba wszystkich wywołań algorytmu DFSm może być określona wzorem (5.42).

$$|DFS_m| = 1 + \sum_{i=1}^{|V|-1} |DFS_{m_i}| \quad (5.42)$$

gdzie:

$$|DFS_{m_i}| = \begin{cases} \frac{|E|}{|V|}; & i = 1 \\ |DFS_{m_{i-1}}(v_x \neq v_t)| \times \left(\frac{|E|}{|V|} - 1\right) \times \left(1 - \frac{i-2}{|V|-2}\right); & i > 1 \end{cases}$$

$$|DFS_{m_{i-1}}(v_x \neq v_t)| = |DFS_{m_{i-1}}| \times \left(1 - \frac{1}{|V|-i-1}\right)$$

Operując na zbiorze równań (5.42), możliwe jest obliczenie w sposób iteracyjny (liczba iteracji wynosi $|V|-1$) wartości oczekiwanej liczby wywołań algorytmu DFS_m niezbędnych dla wykrycia wszystkich tras pomiędzy dowolną parą węzłów w sieci o znanej liczbie węzłów $|V|$ i łączy sieciowych $|E|$. Ze względu na to, że w ramach zaproponowanej w niniejszej pracy metody, algorytm DFS_m jest uruchamiany dla każdej pary węzłów w sieci (5.1), stosowanie liczby uruchomień algorytmu DFS_m ($|DFS_m|$) we wzorze (5.42) jest przydatne dla analizy prawdopodobnej oczekiwanej złożoności zamiast wartości f_{DFS_m} we wzorze (5.8).

Tak czy inaczej, wysokie złożoności: obliczeniowa oraz pamięciowa nie przeszkodami dla wdrożenia metody w sieciach optycznych. W trakcie symulacji przeprowadzonych w ramach badań dla topologii rzeczywistych sieci, obliczenie wszystkich kwantów nie zajmowało więcej niż 9 minut na procesorze Intel Core™ i5-5200U, a wielkość wykorzystanej pamięci RAM nie przekroczyła progu 2 GB. Oprócz tego, topologie sieci transportowych są niezmiennie w długim czasie a obliczenie kwantów jest wymagane jedynie raz (w momencie wdrożenia rozwiązania), a następnie wyłącznie w momencie zmiany topologii sieci. Warto również pamiętać, że węzły CZRS nie są zwykłymi routerami i dysponują znacznymi zasobami pamięci.

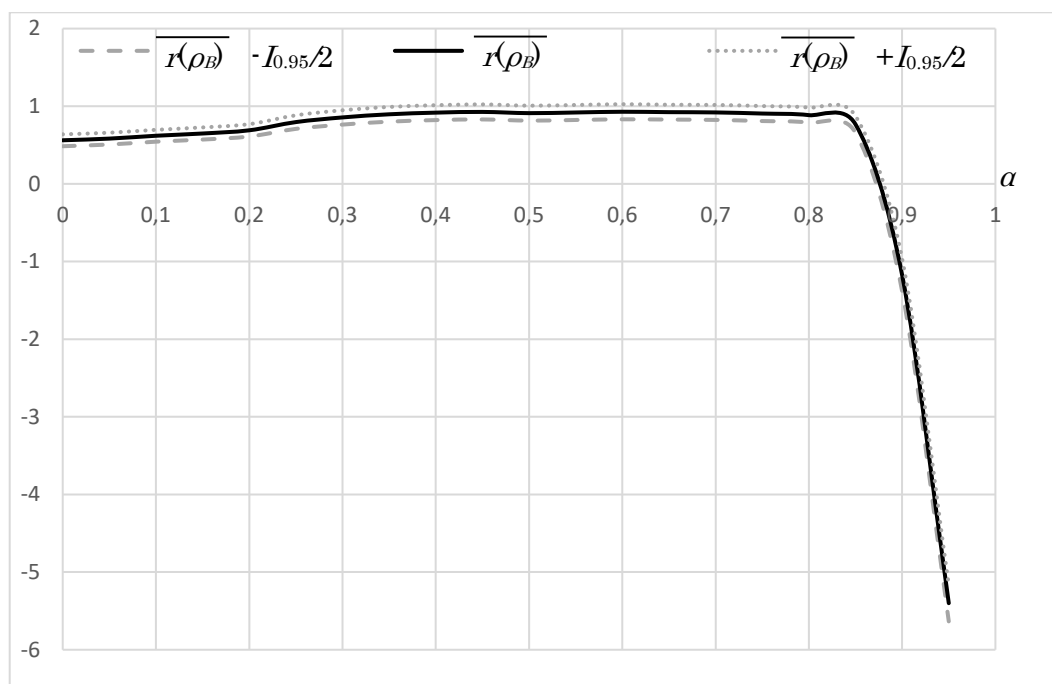
W każdym razie widoczne są ograniczenia dla wdrożenia algorytmu w scenariuszach:

- niskiej wartości dostępnej pamięci CZRS;
- topologii sieci zmiennej w krótkim czasie;
- sieci o dużej liczbie węzłów lub/i sieć o gęstej topologii.

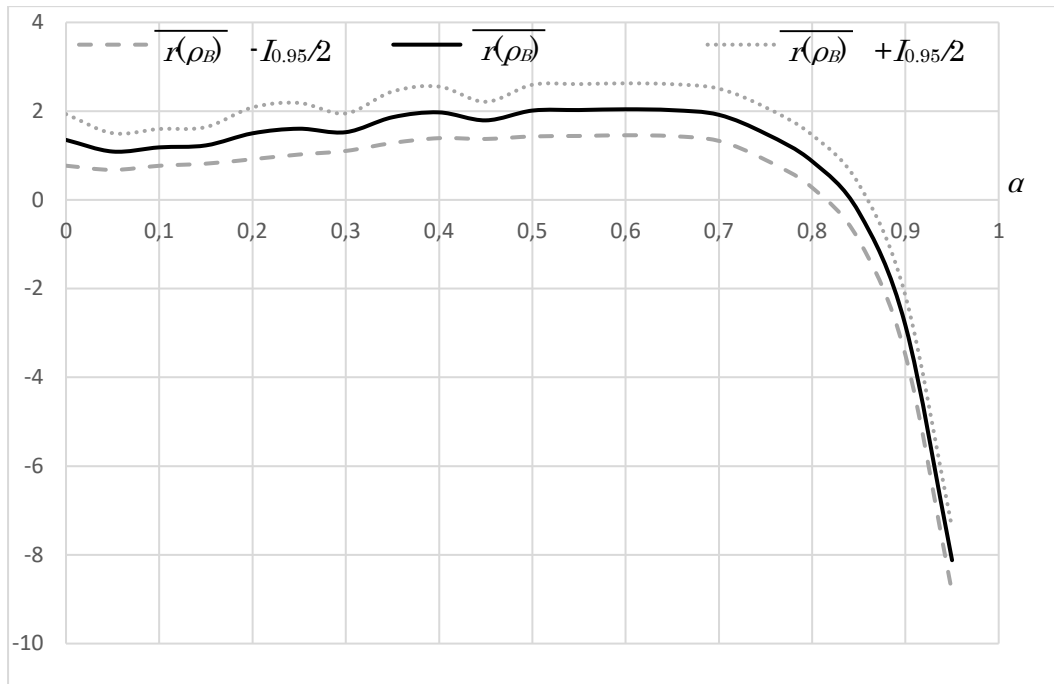
Aneks 3 – Wyniki badań efektywności metody kwantowej

W niniejszym aneksie są prezentowane wyniki badań efektywności metody kwantowej z podrozdziału 3.1.

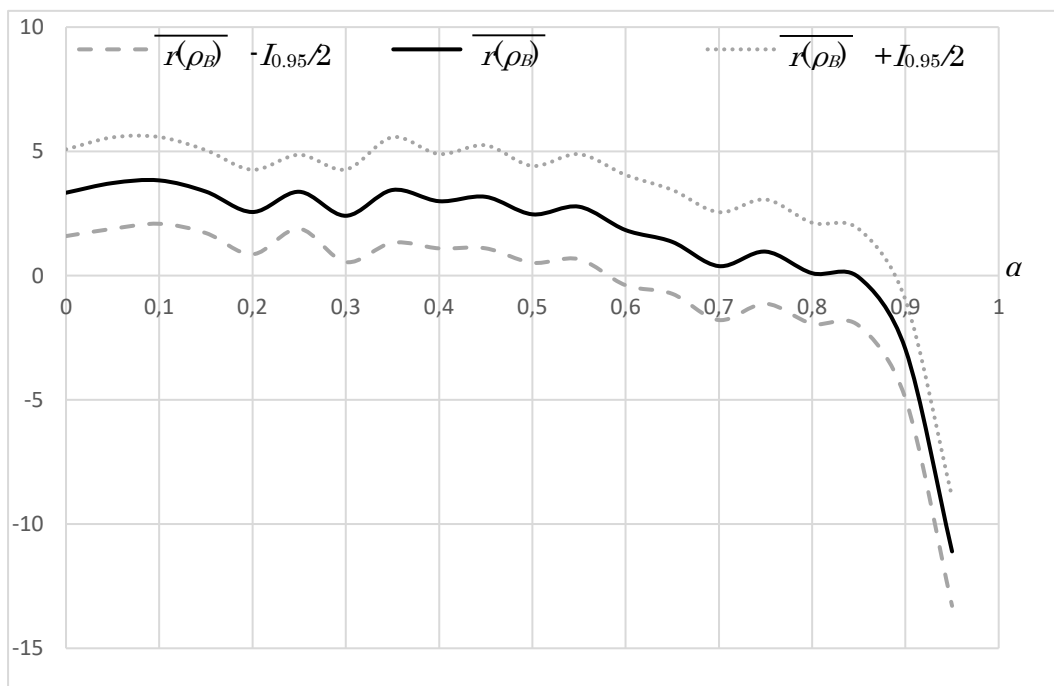
Na Rys. 5.2 jest zaprezentowana zależność średniej przewagi proponowanej w niniejszej rozprawie metody obliczenia wartości metryki kosztu łącza nad metoda referencyjną $\overline{r(\rho_B)}$ w procentach oraz granice przedziału ufności o poziomie 0.95 dla tejże wartości $\overline{r(\rho_B)} \pm I_{0.95}/2$ dla modelu ruchu A0_4 i topologii sieci USA. Analogiczne wykresy dla topologii sieci USA są zaprezentowane na Rys. 5.3 (dla modelu ruchu A0_5), Rys. 5.4 (dla modelu ruchu A0_6) i Rys. 5.5 (dla modelu ruchu A0_12).



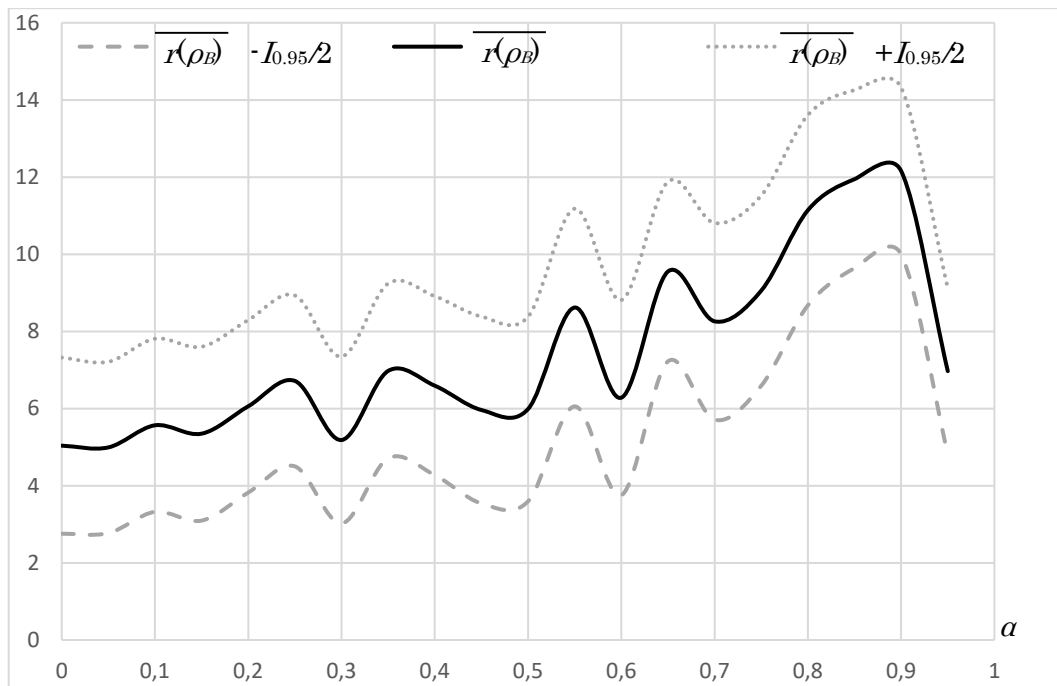
Rys. 5.2 Wyniki badań efektywności zaproponowanej metody dla topologii sieci USA i modelu ruchu A0_4



Rys. 5.3 Wyniki badań efektywności zaproponowanej metody dla topologii sieci USA i modelu ruchu A0_5



Rys. 5.4 Wyniki badań efektywności zaproponowanej metody dla topologii sieci USA i modelu ruchu A0_6



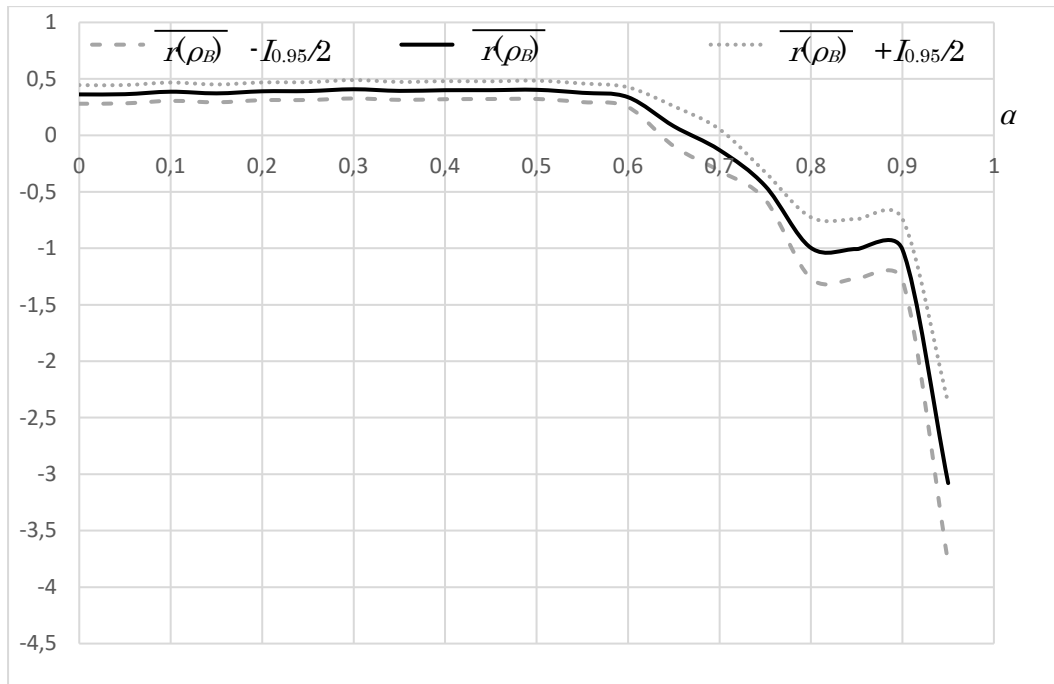
Rys. 5.5 Wyniki badań efektywności zaproponowanej metody dla topologii sieci USA i modelu ruchu A0_12

Na powyższych wykresach często widoczna jest prognozowany początkowy spadek, a następnie wzrost optymalnej wartości α przy wzroście obciążenia sieci. Dla modelu ruchu A0_4 optymalna wartość α znajduje się w obszarze 0.8-0.85; dla modelu A0_5 w obszarze 0.5-0.7, a dla modelu A0_6 spada do ≈ 0.1 . Dla modelu ruchu A0_12 obciążenie sieci jest bardzo wysokie, toteż znaczenie prognozowanych zawczasu konfliktów rośnie, a optymalna wartość α wzrasta do obszaru 0.85-0.9.

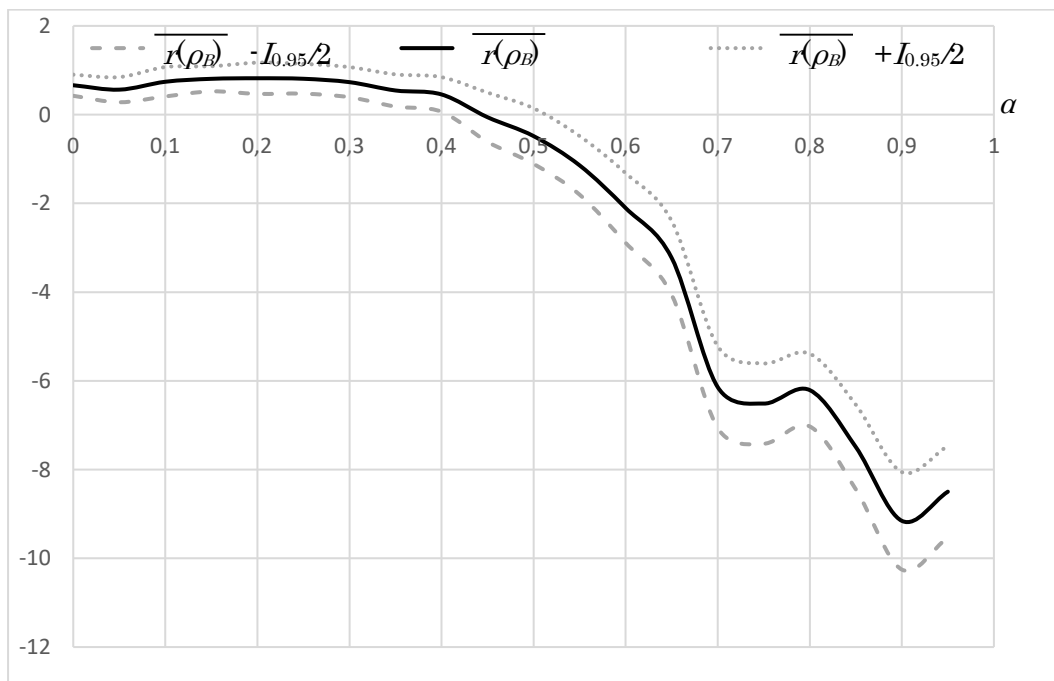
Również widoczne jest, że proponowane podejście ma przewagę nad podejściem referencyjnym także w przypadku niewykorzystania mechanizmu prognozowania konfliktów (dla przypadku $\alpha = 0$).

Warto zwrócić uwagę na występowanie prognozowanych lokalnych ekstremów w zależności od optymalnej wartości α (która z kolei zależy od obciążenia), jak również na wzrost ich liczby ze wzrostem obciążenia sieci.

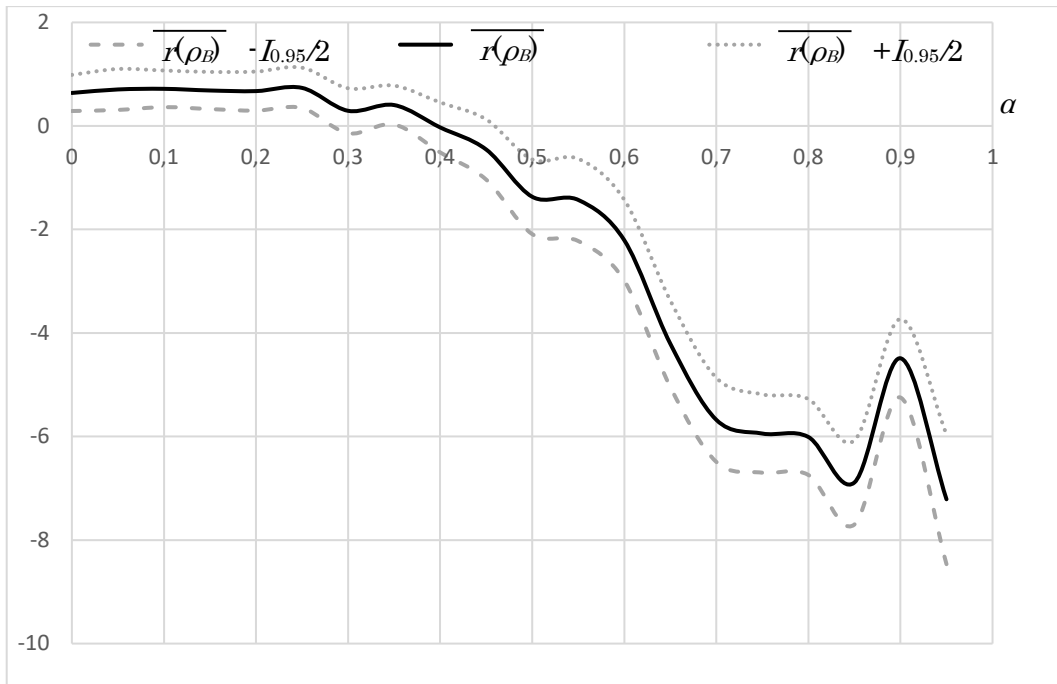
Powyższe trendy są charakterystyczne również dla wyników uzyskanych w badaniach dla topologii sieci Włoskiej na Rys. 5.6-5.9.



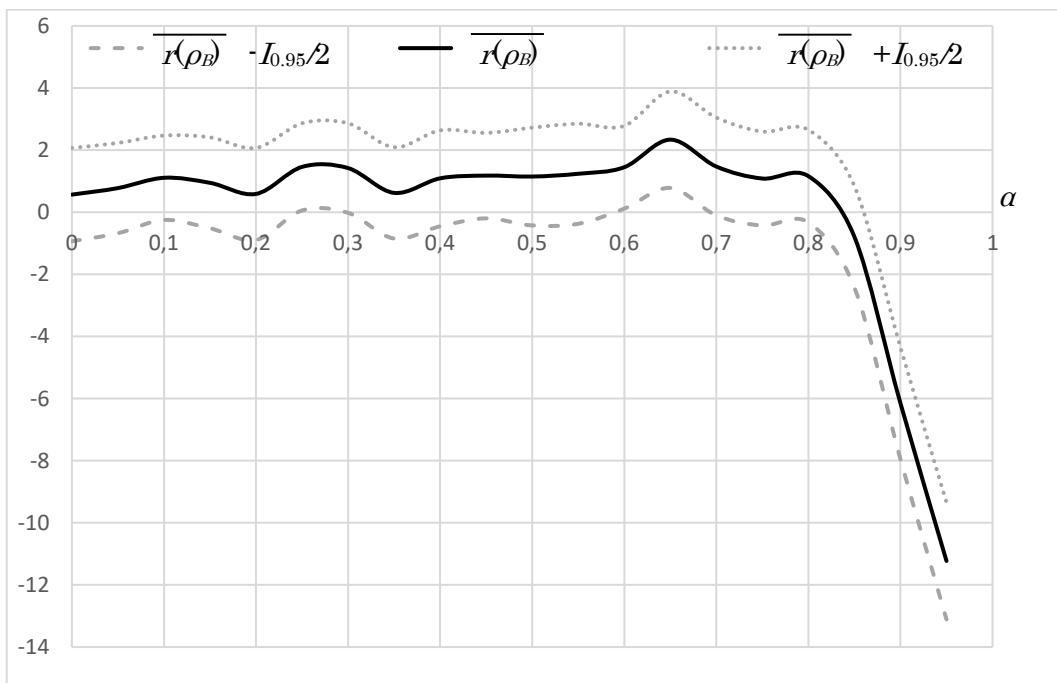
Rys. 5.6 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Włoskiej i modelu ruchu A0_4



Rys. 5.7 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Włoskiej i modelu ruchu A0_5

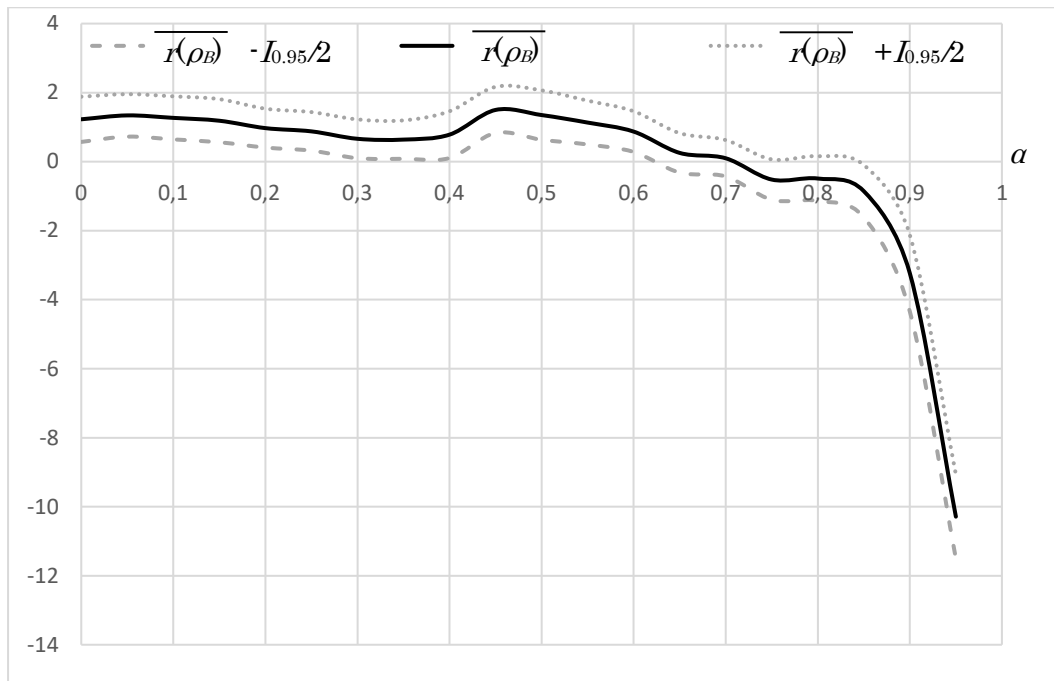


Rys. 5.8 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Włoskiej i modelu ruchu A0_6

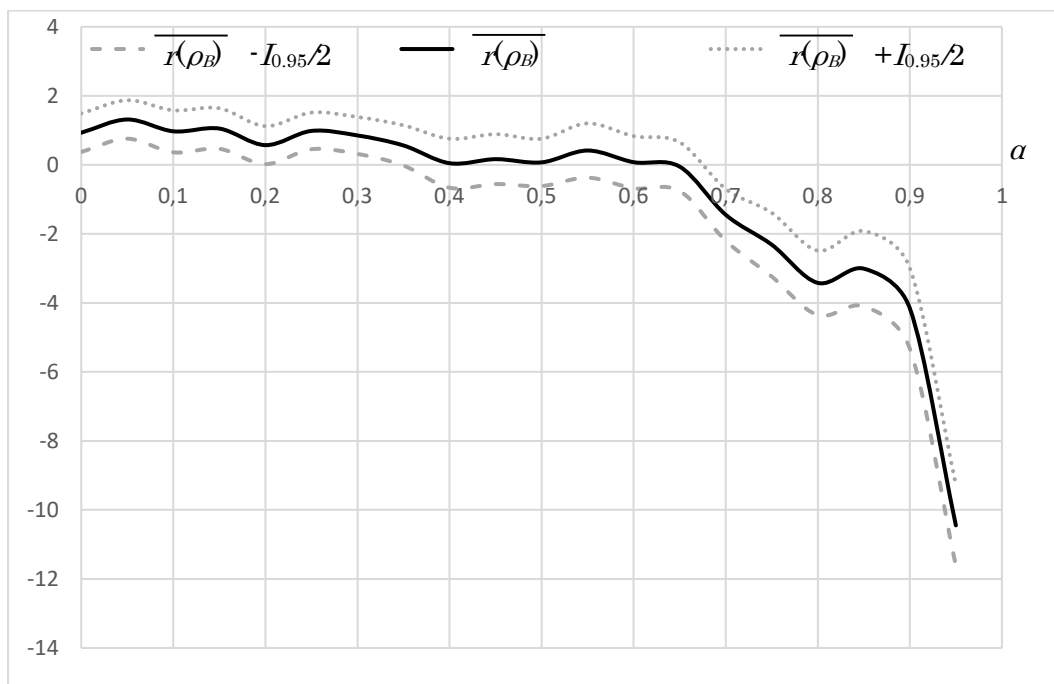


Rys. 5.9 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Włoskiej i modelu ruchu A0_12

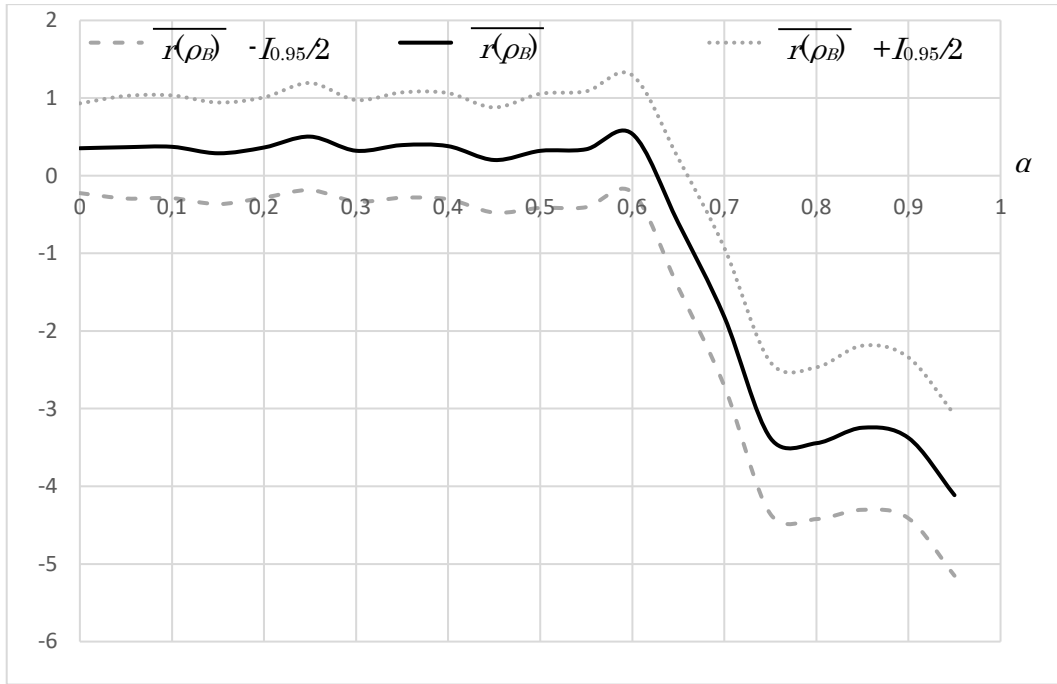
Niewielka różnica w stosunku do wyników osiągniętych dla sieci Włoskiej jest widoczna dla sieci Pionier na Rys. 5.10-5.13.



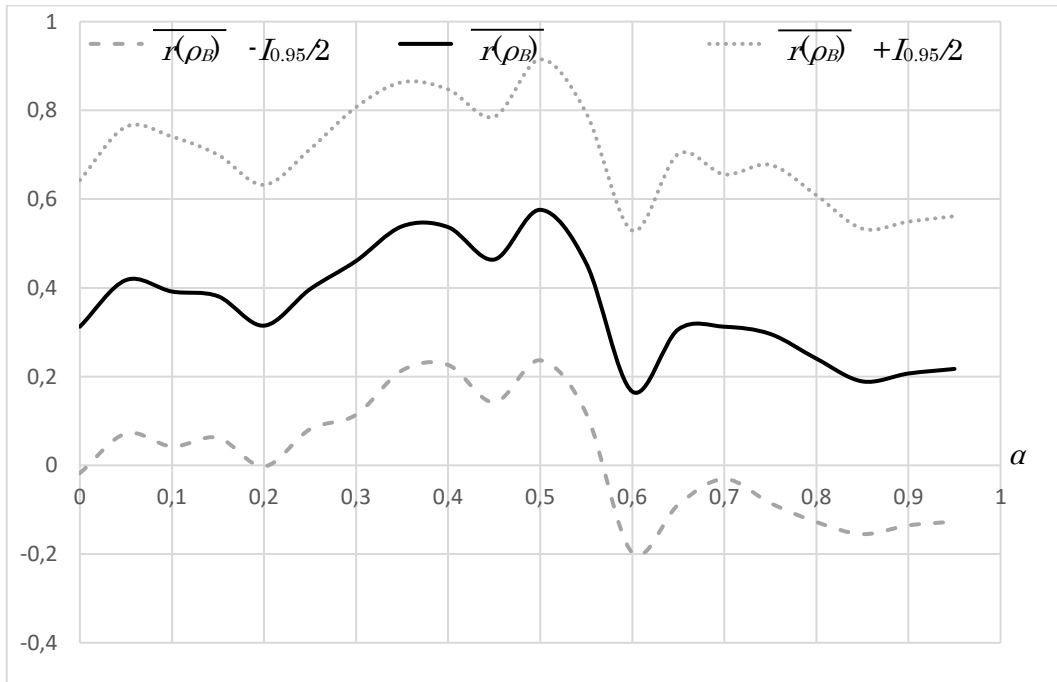
Rys. 5.10 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Pionier i modelu ruchu A0_4



Rys. 5.11 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Pionier i modelu ruchu A0_5



Rys. 5.12 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Pionier i modelu ruchu A0_6



Rys. 5.13 Wyniki badań efektywności zaproponowanej metody dla topologii sieci Pionier i modelu ruchu A0_12

Dla modelu ruchu A0_4 w sieci Pionier są już dość wyraźnie widoczne lokalne ekstrema (w porównaniu do sieci USA i sieci Włoskiej). Powodem tego jest dość niska

pojemność informacyjna topologii sieci Pionier w porównaniu do topologii sieci USA i sieci Włoskiej. Prawdopodobieństwo blokowania żądań dla sieci Pionier dla modelu ruchu A0_4 wyniosło 50.4% dla metody referencyjnej. Z kolei dla sieci USA ta wartość jest równa 1.7%, a dla sieci Włoskiej 1.9%. Model A0_4 przyczynia się do zbyt wysokiego obciążenia dla sieci Pionier. Podobne wartości prawdopodobieństwa blokowania żądań są osiągane przez sieć USA i sieć Włoską tylko w przypadku modelu ruchu A0_12 (odpowiednio 70% i 51.7%).

Mimo tego widoczny jest początkowy spadek optymalnej wartości α_{opt} ze wzrostem obciążenia od $\alpha_{opt} \approx 0.45$ dla modelu A0_4 do $\alpha_{opt} \approx 0.05$ dla modelu A0_5. Modele te dla sieci Pionier już powodują za wysokie obciążenia w (odróżnieniu od sieci USA i sieci Włoskiej). Dalszy wzrost optymalnej wartości α_{opt} występuje dla modelu A0_6 do $\alpha_{opt} \approx 0.6$ w odróżnieniu od dwóch pozostałych topologii, dla których ten wzrost występował dopiero w modelu A0_12.

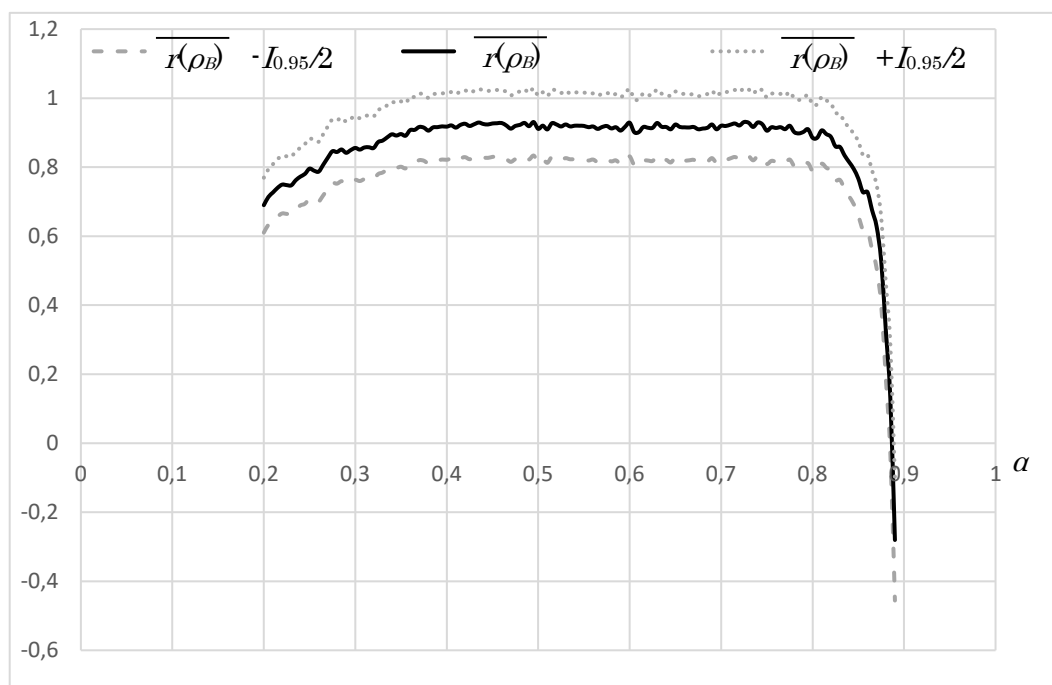
Widoczna jest także różnica pomiędzy wartościami hipotetycznymi (zaprezentowanymi w sekcji 3.1.5) a rzeczywistymi (z niniejszego aneksu). W wyniku dalszego zwiększania obciążenia sieci Pionier (model A0_12) optymalna wartość α_{opt} znowu spada (do $\alpha_{opt} \approx 0.5$). Trudno jest zdefiniować powód takiego nieoczekiwanego zachowywania α_{opt} , tym bardziej, że obciążenie wynikające z modelu A0_12 wielokrotnie przekracza możliwości sieci Pionier i odbiegają od typowego obciążenia sieci. Prawdopodobieństwo blokowania żądań dla modelu A0_12 i sieci Pionier wyniosło 87.3%. Znaczy to, że trasy wyłącznie 12.7% żądań mogą być zainstalowane. W takich warunkach trudno mówić w ogóle o efektywności doboru tras.

Aneks 4 – Dobór ilorazu α/β

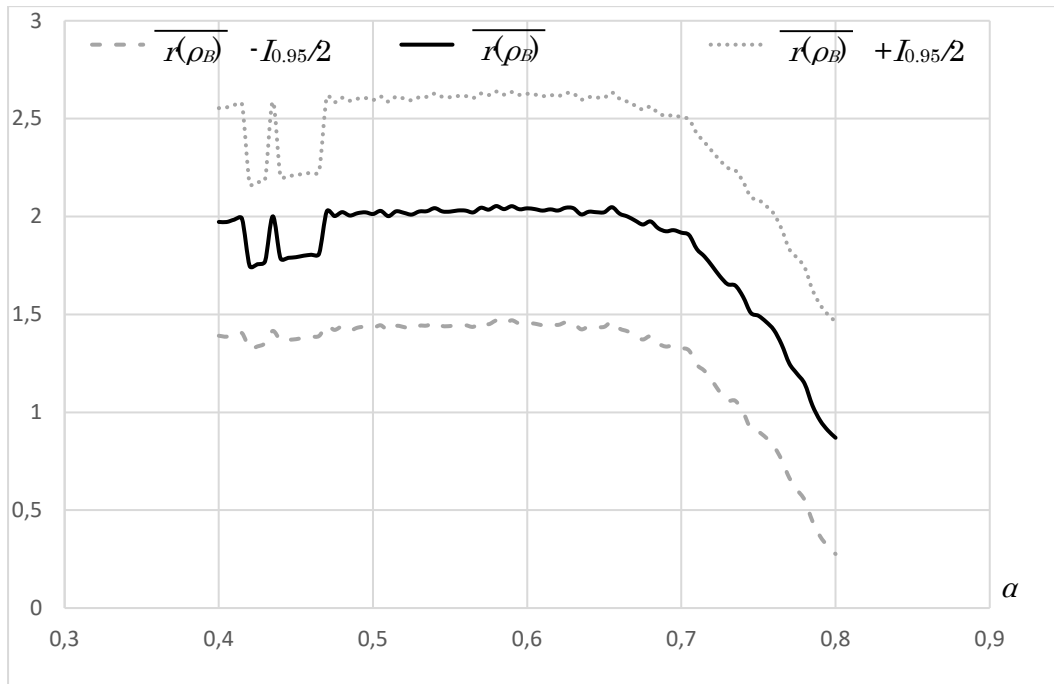
W niniejszym aneksie są prezentowane wyniki badań doboru ilorazu α/β dla metody kwantowej opisanej w podrozdziale 3.1.

Optymalne wartości α_{opt} uzyskane w trakcie doboru ilorazu α/β są zaprezentowane na Rys. 5.14-5.25 dla każdej topologii sieci i modelu ruchu sieciowego oraz podsumowane w Tabeli 5.1.

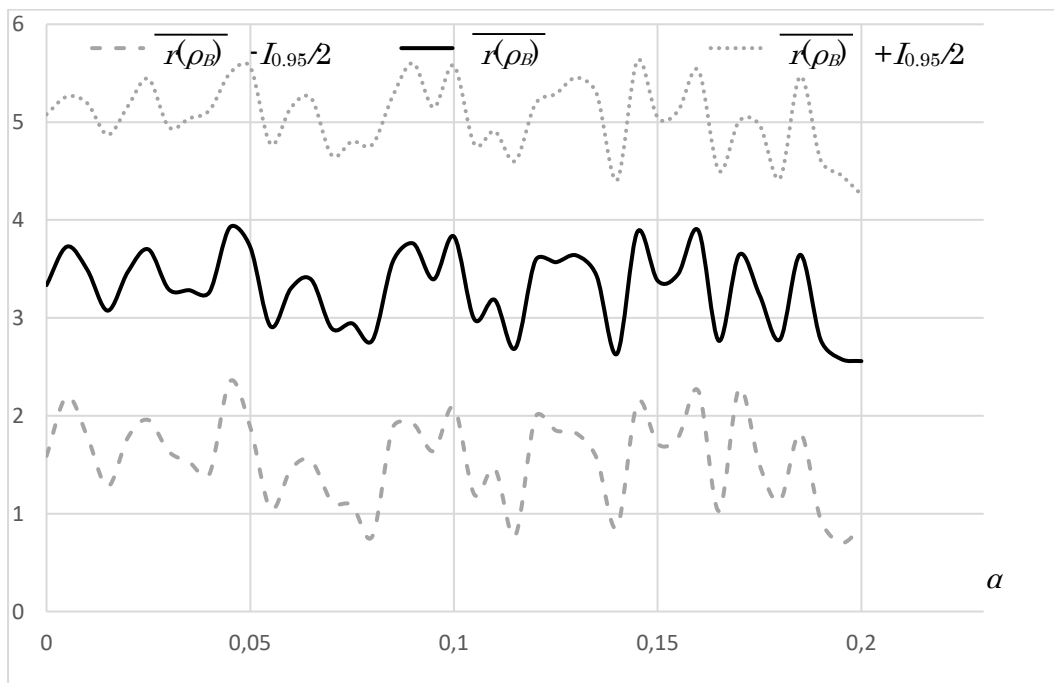
Wykresy uzyskane w trakcie pierwszego etapu symulacji dotyczącego doboru ilorazu α/β nie są zaprezentowane w niniejszym aneksie z powodu ich dużego podobieństwa do wyników ukazanych na Rys. 5.2-5.13.



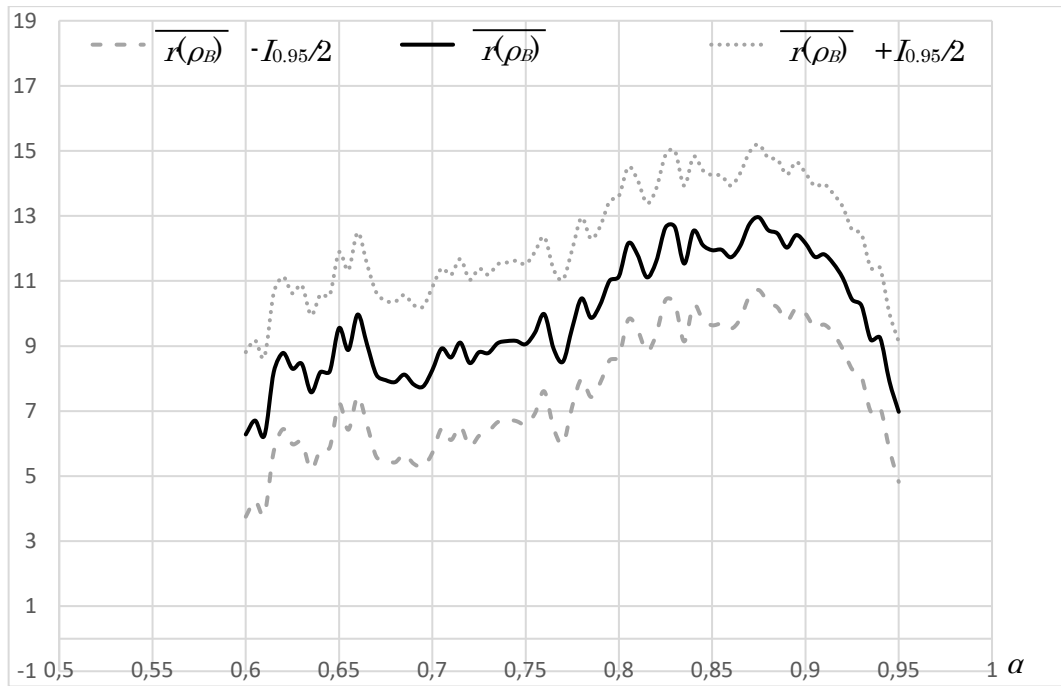
Rys. 5.14 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć USA, model A0_4)



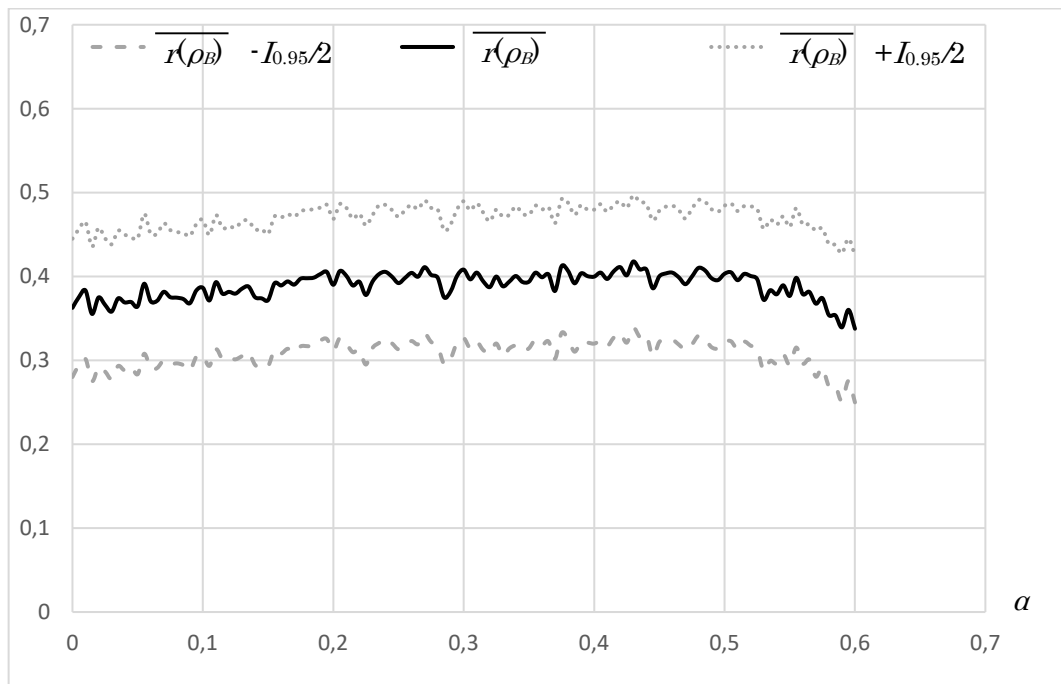
Rys. 5.15 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu a/b (sieć USA, model A0_5)



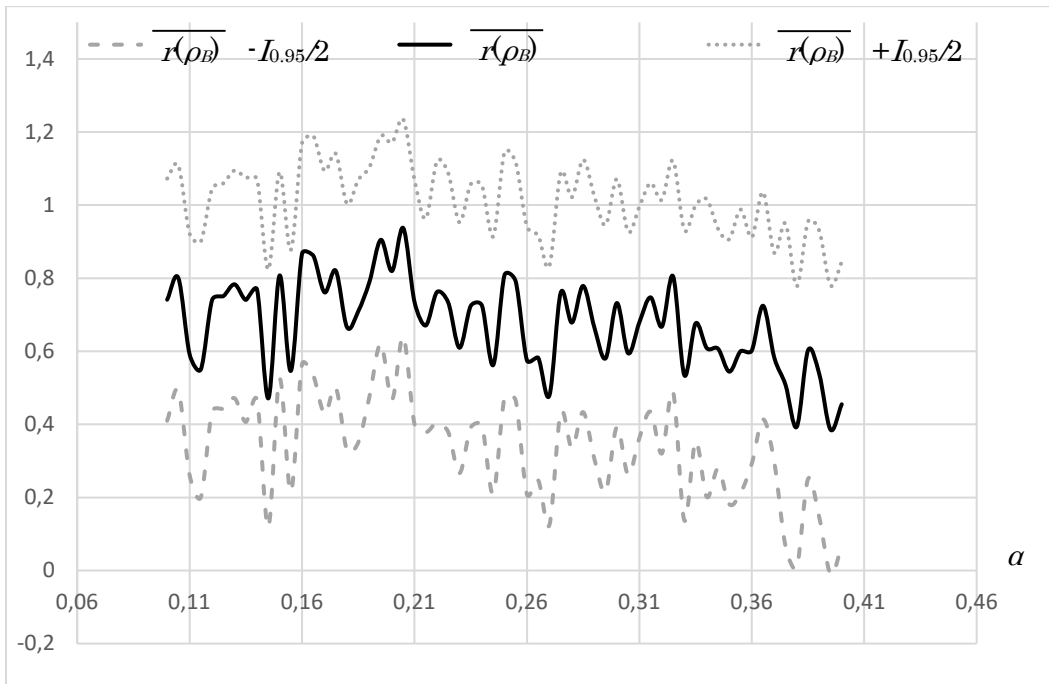
Rys. 5.16 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu a/b (sieć USA, model A0_6)



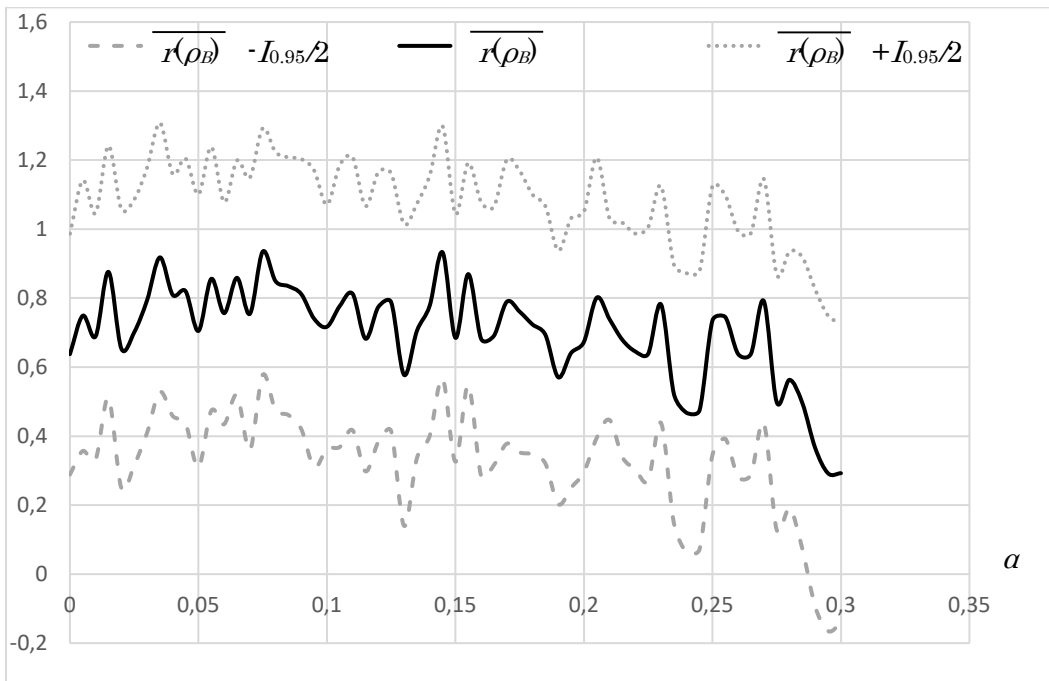
Rys. 5.17 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć USA, model A0_12)



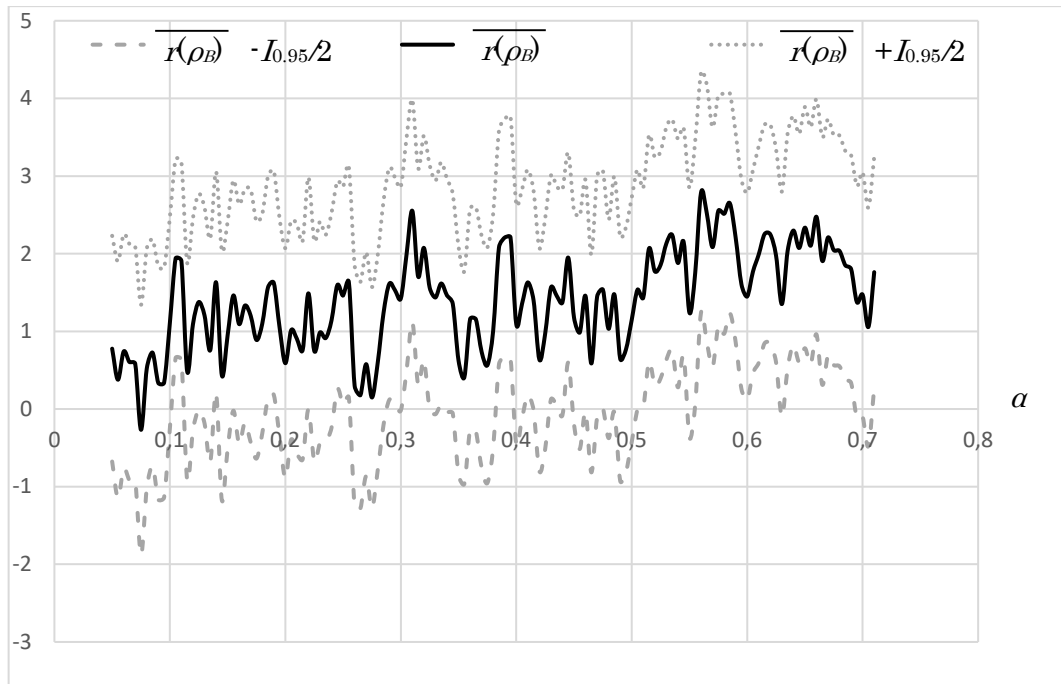
Rys. 5.18 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Włoska, model A0_4)



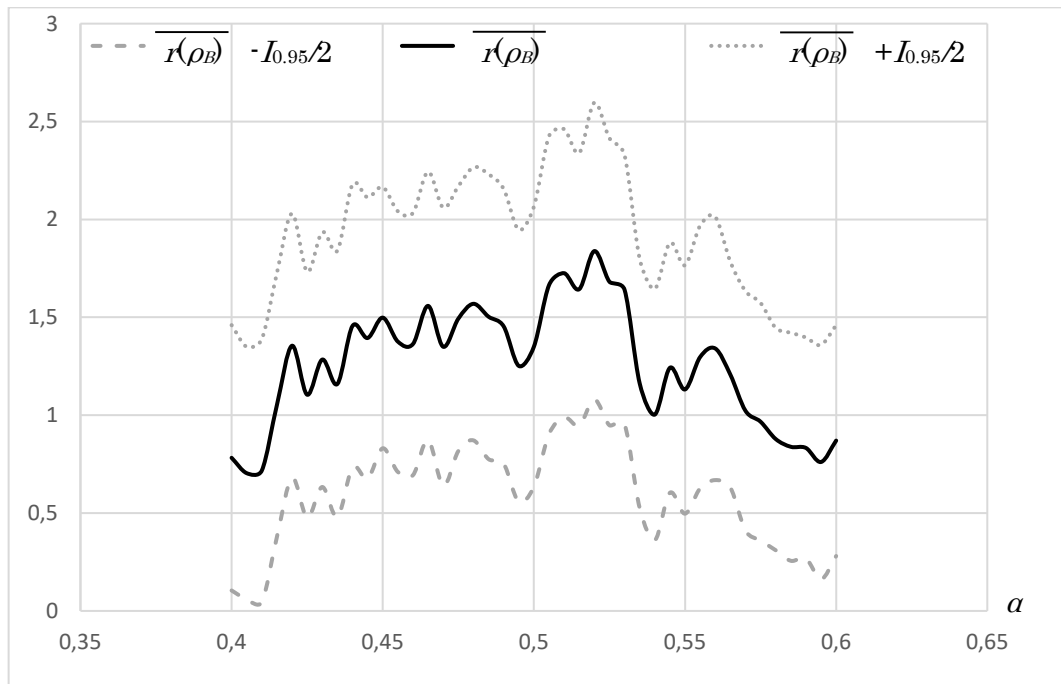
Rys. 5.19 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Włoska, model A0_5)



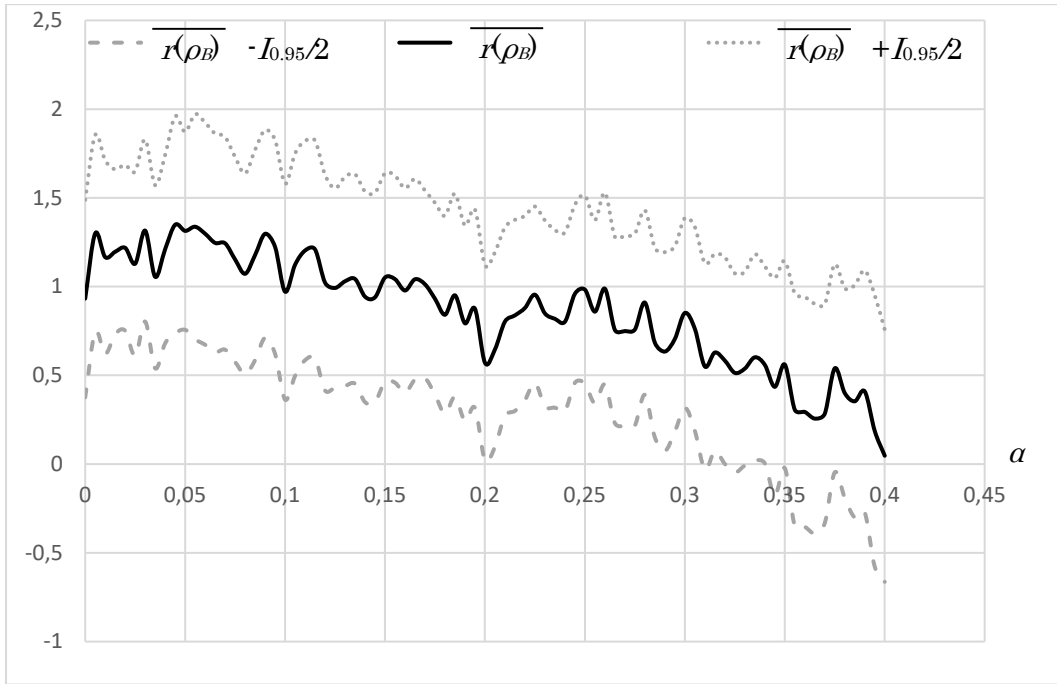
Rys. 5.20 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Włoska, model A0_6)



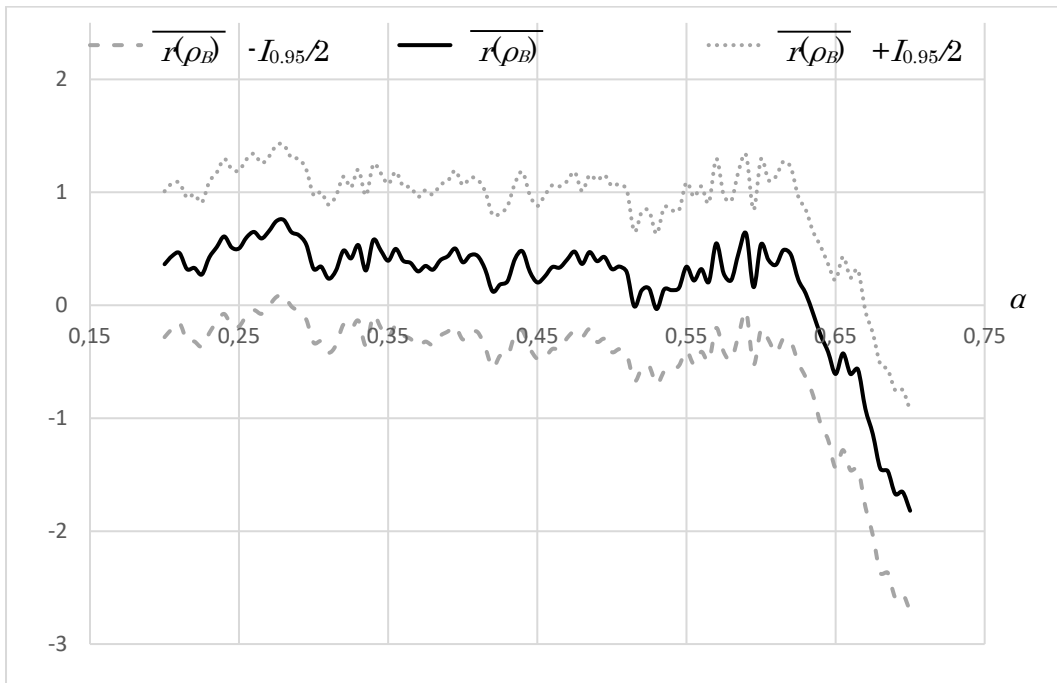
Rys. 5.21 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Włoska, model A0_12)



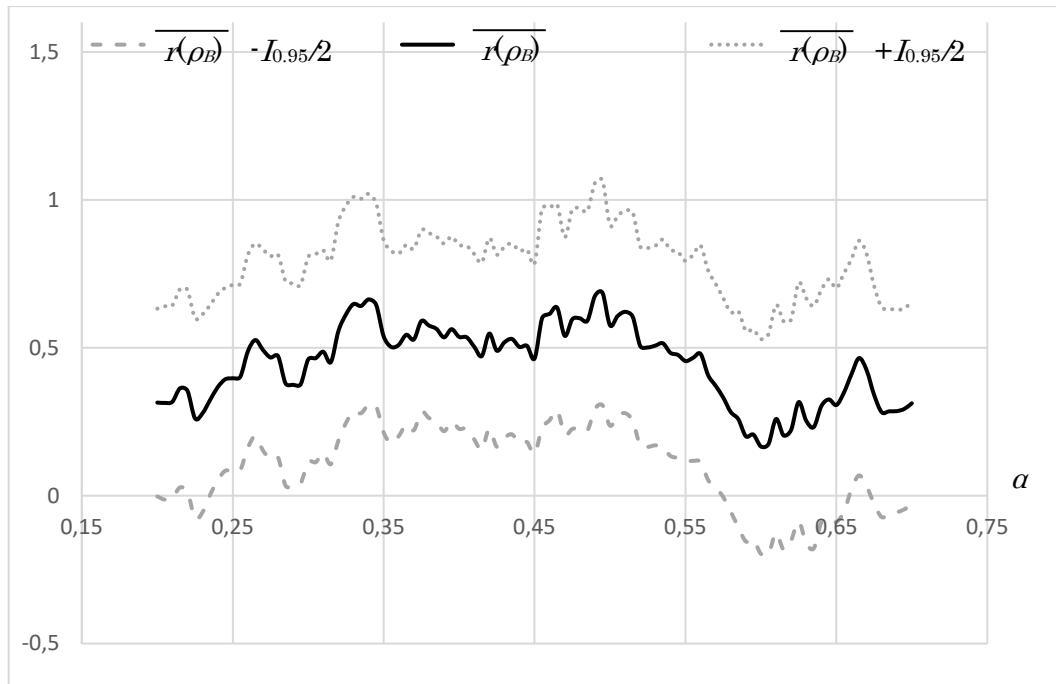
Rys. 5.22 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Pionier, model A0_4)



Rys. 5.23 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Pionier, model A0_5)



Rys. 5.24 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Pionier, model A0_6)



Rys. 5.25 Wyniki drugiego etapu symulacji w trakcie doboru ilorazu α/β (sieć Pionier, model A0_12)

Model ruchu sieciowego	Sieć USA	Sieć Włoska	Sieć Pionier
A0_4	0.600	0.430	0.520
A0_5	0.600	0.205	0.055
A0_6	0.045	0.145	0.280
A0_12	0.850	0.585	0.495

Tabela 5.1 Optymalne wartości α_{opt} dla modeli ruchu i topologii sieci



Aneks 5 – Wyniki symulacji algorytmu kwantowego

W niniejszym aneksie są prezentowane szczegółowe wyniki symulacji metody kwantowej opisanej w podrozdziale 3.1.

Parametr	Wynik
Model ruchu A0_4. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	1.728
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.192
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\mathcal{P})}$ (ms)	1.399
Długość przedziału ufności $I_{0.95}[\overline{t_r(\mathcal{P})}]$ (ms)	0.061
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	67.005
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.190
Średnia liczba przeskoków w trasie $ \overline{E(\mathcal{P})} $	3.394
Długość przedziału ufności dla $I_{0.95}[\overline{ \mathcal{E}(\mathcal{P}) }]$	0.054
Model ruchu A0_4. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	0.801
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.136
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\mathcal{P})}$ (ms)	1.644
Długość przedziału ufności $I_{0.95}[\overline{t_r(\mathcal{P})}]$ (ms)	0.060
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	63.288
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.216
Średnia liczba przeskoków w trasie $ \overline{E(\mathcal{P})} $	3.190
Długość przedziału ufności dla $I_{0.95}[\overline{ \mathcal{E}(\mathcal{P}) }]$	0.004



Model ruchu A0_5. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	9.617
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.459
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\overline{P})}$ (ms)	1.399
Długość przedziału ufności $I_{0.95}[\overline{t_r(\overline{P})}]$ (ms)	0.065
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	74.020
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.367
Średnia liczba przeskoków w trasie $ \overline{E(\overline{P})} $	3.388
Długość przedziału ufności dla $I_{0.95}[\overline{ \overline{E(\overline{P})} }]$	0.009
Model ruchu A0_5. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	7.857
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.229
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\overline{P})}$ (ms)	1.628
Długość przedziału ufności $I_{0.95}[\overline{t_r(\overline{P})}]$ (ms)	0.069
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	70.467
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.245
Średnia liczba przeskoków w trasie $ \overline{E(\overline{P})} $	3.184
Długość przedziału ufności dla $I_{0.95}[\overline{ \overline{E(\overline{P})} }]$	0.005
Model ruchu A0_6. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	21.989
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	1.753
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\overline{P})}$ (ms)	1.287
Długość przedziału ufności $I_{0.95}[\overline{t_r(\overline{P})}]$ (ms)	0.057
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	77.481
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.273

Średnia liczba przeskoków w trasie $\overline{ E(\mathcal{P}) }$	3.580
Długość przedziału ufności dla $I_{0.95}[\overline{ E(\mathcal{P}) }]$	0.037
Model ruchu A0_6. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	18.774
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	1.453
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\mathcal{P})}$ (ms)	1.536
Długość przedziału ufności $I_{0.95}[\overline{t_r(\mathcal{P})}]$ (ms)	0.065
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	76.808
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.080
Średnia liczba przeskoków w trasie $\overline{ E(\mathcal{P}) }$	3.443
Długość przedziału ufności dla $I_{0.95}[\overline{ E(\mathcal{P}) }]$	0.026
Model ruchu A0_12. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	69.985
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	1.479
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\mathcal{P})}$ (ms)	0.888
Długość przedziału ufności $I_{0.95}[\overline{t_r(\mathcal{P})}]$ (ms)	0.046
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	59.775
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	2.026
Średnia liczba przeskoków w trasie $\overline{ E(\mathcal{P}) }$	3.967
Długość przedziału ufności dla $I_{0.95}[\overline{ E(\mathcal{P}) }]$	0.049
Model ruchu A0_12. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	57.950
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	1.481
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\mathcal{P})}$ (ms)	1.198
Długość przedziału ufności $I_{0.95}[\overline{t_r(\mathcal{P})}]$ (ms)	0.049

Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	79.384
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	2.218
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.701
Długość przedziału ufności dla $I_{0.95}[\overline{E(P)}]$	0.043

Tabela 5.2 Wyniki symulacji dla sieci USA

Warto zwrócić uwagę, że ze wzrostem obciążenia sieci rośnie współczynnik wykorzystania zasobów sieci. Jest to logiczne, ponieważ im większe są wymagania dotyczące przydziału zasobów sieci (dla instalacji tras większej liczby żądań) tym więcej zasobów sieci zostanie finalnie zaangażowanych dla obsługi żądań.

Jednak jak jest widoczne w Tabeli 5.2, w przypadku wyznaczania wartości metryki kosztu łącza zgodnie ze wzorem (3.1), stopień wykorzystania zasobów sieci \overline{U} (%) dla modelu ruchu sieciowego A0_12 jest niższy w porównaniu do swojej wartości dla modelu ruchu sieciowego A0_6 (odpowiednio: 59.775% i 76.808%). W tym samym momencie współczynnik blokowania żądań wzrósł do 69.985% dla modelu A0_12 z 21.989% dla modelu A0_6.

Mówiąc inaczej, z dalszym wzrostem obciążenia sieci wzrasta liczba nierealizowanych żądań instalację tras i (mimo tego) równolegle wzrasta ilość niewykorzystanych zasobów przepustowości sieci. Ten efekt mówi o tym, że algorytm referencyjny w przypadku takiego wysokiego obciążenia jak A0_12 nie ma możliwości efektywnego przydziału zasobów przepustowości sieci (ponad 40% zasobów łącza pozostaje niewykorzystanych), co skutkuje wzrostem liczby odrzuconych żądań z powodu wyczerpania przepustowości jednych łącza sieci przy niewykorzystaniu dostępnych zasobów na innych łączach.

Z kolei, można zauważyć, że proponowana w niniejszej pracy metoda działa bardziej efektywnie. Ze wzrostem obciążenia wzrasta również współczynnik wykorzystanych zasobów sieci: z 76.808% dla modelu ruchu A0_6 do 79.384% dla modelu ruchu A0_12.

Parametr	Wynik
Model ruchu A0_4. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	1.871
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.209
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	0.927
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.034
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	61.717
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.220
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.348
Długość przedziału ufności dla $I_{0.95}[\overline{ \overline{E(P)} }]$	0.005
Model ruchu A0_4. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	1.479
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.190
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	1.076
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.033
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	59.846
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.223
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.327
Długość przedziału ufności dla $I_{0.95}[\overline{ \overline{E(P)} }]$	0.004
Model ruchu A0_5. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	9.905
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.451
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	0.885
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.044
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	69.758

Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.278
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.430
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.008
Model ruchu A0_5. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	9.191
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.399
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	1.041
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.023
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	66.158
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.252
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.365
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.007
Model ruchu A0_6. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	17.584
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.500
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	0.799
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.050
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	75.993
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.274
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.489
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.010
Model ruchu A0_6. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	16.777
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.434
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	0.976

Długość przedziału ufności $I_{0.95}[\overline{t_t(P)}]$ (ms)	0.036
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	74.445
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.255
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.437
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.008
Model ruchu A0_12. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	51.667
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	1.205
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_t(P)}$ (ms)	0.678
Długość przedziału ufności $I_{0.95}[\overline{t_t(P)}]$ (ms)	0.059
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	78.986
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.300
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.382
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.037
Model ruchu A0_12. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	49.628
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	1.046
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_t(P)}$ (ms)	0.804
Długość przedziału ufności $I_{0.95}[\overline{t_t(P)}]$ (ms)	0.055
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	79.970
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.194
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	3.275
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.026

Tabela 5.3 Wyniki symulacji dla sieci Włoskiej

Dla wykorzystanych w niniejszej pracy modeli ruchu sieciowego, sieć Włoska w warunkach wysokiego obciążenia uzyskuje lepsze wyniki niż sieć USA. Nie znaczy to, że sieć Włoska ma większą pojemność informacyjną niż sieć USA; ten efekt jest powodowany modelem ruchu sieciowego. W symulacjach wykorzystane są modele ruchu obejmujące żądania instalacji tras pomiędzy każdą parę węzłów sieci.

Z kolei znaczy to, że jest możliwych 378 żądań dla sieci USA (topologia tej sieci zawiera 28 węzłów) oraz 210 żądań dla sieci Włoskiej (zawierającej 21 węzłów). Z tego wynika, że średni przepływ przez sieć USA jest wyższy niż średni przepływ przez sieć Włoską o 80%. Inna charakterystyka, która może być zinterpretowana nieprawidłowo dotyczy redukcji wartości czasu niezbędnego do wyznaczenia tras wraz ze wzrostem obciążenia sieci ukazanego w Tabeli 5.3. Obciążenie sieci w żaden sposób nie przyspiesza działania algorytmów doboru tras. Redukcja ta jest spowodowana tym, że ze wzrostem obciążenia sieci wzrasta liczba żądań dla których nie jest możliwy dobór tras.

Parametr	Wynik
Model ruchu A0_4. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	50.394
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.459
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	1.411
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.064
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	69.240
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.442
Średnia liczba przeskoków w trasie $\overline{ E(P) }$	4.52
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.013
Model ruchu A0_4. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	49.029
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.672
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	1.700

Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.068
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	68.079
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	0.678
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	4.338
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.013
Model ruchu A0_5. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań na instalację tras $\overline{\rho_B}$ (%)	59.722
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.805
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	1.281
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.056
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	66.808
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.001
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	4.475
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.023
Model ruchu A0_5. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	58.611
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.963
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(P)}$ (ms)	1.586
Długość przedziału ufności $I_{0.95}[\overline{t_r(P)}]$ (ms)	0.062
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	65.557
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.220
Średnia liczba przeskoków w trasie $ \overline{E(P)} $	4.378
Długość przedziału ufności dla $I_{0.95}[\overline{ E(P) }]$	0.021
Model ruchu A0_6. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	71.887

Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.744
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\overline{P})}$ (ms)	1.225
Długość przedziału ufności $I_{0.95}[\overline{t_r(\overline{P})}]$ (ms)	0.053
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	56.471
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.219
Średnia liczba przeskoków w trasie $ \overline{E(\overline{P})} $	4.659
Długość przedziału ufności dla $I_{0.95}[\overline{ E(\overline{P}) }]$	0.025
Model ruchu A0_6. Metryk kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	71.717
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.766
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\overline{P})}$ (ms)	1.425
Długość przedziału ufności $I_{0.95}[\overline{t_r(\overline{P})}]$ (ms)	0.062
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	53.091
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.275
Średnia liczba przeskoków w trasie $ \overline{E(\overline{P})} $	4.367
Długość przedziału ufności dla $I_{0.95}[\overline{ E(\overline{P}) }]$	0.200
Model ruchu A0_12. Metryka kosztu łącza (3.1).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	87.303
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.437
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\overline{P})}$ (ms)	0.994
Długość przedziału ufności $I_{0.95}[\overline{t_r(\overline{P})}]$ (ms)	0.029
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	47.317
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.316
Średnia liczba przeskoków w trasie $ \overline{E(\overline{P})} $	4.654
Długość przedziału ufności dla $I_{0.95}[\overline{ E(\overline{P}) }]$	0.037

Model ruchu A0_12. Metryka kosztu łącza (3.2).	
Prawdopodobieństwo blokowania żądań instalacji tras $\overline{\rho_B}$ (%)	86.992
Długość przedziału ufności $I_{0.95}[\overline{\rho_B}]$ (%)	0.477
Średni czas doboru tras dla wszystkich żądań w zbiorze $\overline{t_r(\mathcal{P})}$ (ms)	1.279
Długość przedziału ufności $I_{0.95}[\overline{t_r(\mathcal{P})}]$ (ms)	0.056
Średni współczynnik wykorzystania zasobów sieci \overline{U} (%)	41.781
Długość przedziału ufności $I_{0.95}[\overline{U}]$ (%)	1.530
Średnia liczba przeskoków w trasie $ \overline{E(\mathcal{P})} $	4.024
Długość przedziału ufności dla $I_{0.95}[\overline{ \overline{E(\mathcal{P})} }]$	0.025

Tabela 5.4 Wyniki symulacji dla sieci Pionier

W Tabeli 5.4 widoczne jest, że model A0_4 powoduje dość wysokie obciążenie dla sieci Pionier. Już w przypadku tego modelu współczynnik blokowania żądań jest wysoki (około 50%). Opisany wcześniej dla sieci USA efekt (wzrost współczynnika blokowania żądań przy redukcji współczynnika wykorzystania zasobów) który mówi o niemożliwości efektywnego doboru tras jest widoczny dla sieci Pionier już dla modelu A0_5. Warto zwrócić uwagę, że w odróżnieniu od sytuacji z siecią USA (gdzie algorytm referencyjny z powodu wysokiego obciążenia w modelu ruchu sieciowego A0_12 gubił zdolność efektywnego przydziału zasobów), w przypadku sieci Pionier proponowana w niniejszej pracy metoda (jak i podejście referencyjne) również zatracą możliwość efektywnego doboru tras. Ze wzrostem obciążenia sieci równolegle wzrasta wartość współczynnika blokowania żądań i spada wartość współczynnika wykorzystania zasobów dla proponowanej metody (powód tego efektu jest taki sam, jak i w przypadku opisanym dla sieci USA i modelu ruchu sieciowego A0_12).

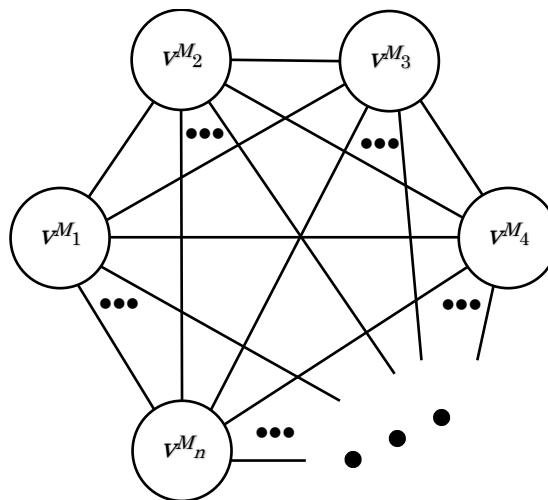


Aneks 6 – Analiza złożoności algorytmicznej PPCB

W niniejszym aneksie są prezentowane wyniki analizy złożoności algorytmicznej podejścia PPCB (Procedury Połączenia Cykli Bazowych) opisanej w rozdziale 4.

Prezentowana w podrozdziale 4.2 metoda wykonuje operacje na mapie komórkowej zamiast na grafie topologii sieci. Podstawowym elementem mapy komórkowej są cykle bazowe. Złożoność algorytmiczna procedury wyznaczenia cykli bazowych w grafie jest rzędu $O((|E|^2 \cdot |V|)/(\log |V|))$ [74]. Zbudowanie mapy komórkowej nie jest wymagane dla każdego uruchomienia PPCB. Wyznaczona jeden raz mapa komórkowa może być wykorzystana wielokrotnie dla tej samej topologii sieci.

Algorytm PPCB obejmuje pierwszą część iteracyjną (Krok 2), która z kolei zawiera w sobie drugą część iteracyjną (wiersze 8-16 algorytmu). Liczba iteracji w drugiej części iteracyjnej zależy generalnie od cyklu bazowego wybranego w pierwszej części iteracyjnej oraz od spełnienia warunku w wierszu 14 algorytmu PPCB. Z tego wynika, że dla mapy komórkowej o liczbie $n = |V^M|$ cykli bazowych uruchomienie PPCB na wybranym cyklu bazowym v^{M_i} będzie się wiązało z maksymalną liczbą iteracji w przypadku, kiedy liczba sąsiednich cykli bazowych względem v^{M_i} wynosi $|V^M| - 1$ (Rys 5.26).



Rys. 5.26 Przykład najgorszego przypadku mapy komórkowej pod względem liczby iteracji PPCB

W takim przypadku, jeżeli podczas ostatniej $|V^M| - 1$ iteracji zostanie spełniony warunek w wierszu 14, algorytm doda sąsiedni cykl bazowy do cyklu S_i i powtórzy drugą część iteracyjną dla cyklu bazowego v^M_i . W wyniku tego, po uruchomieniu $|V^M| - 1$ iteracji zostaną wykonane jeszcze $|V^M| - 2$ iteracje – o jedną iterację mniej niż poprzednim razem, ponieważ jeden sąsiedni cykl bazowy został dodany do cyklu S_i oraz do zbioru B (nie spełnia warunku w wierszu 8). W najgorszym przypadku w trakcie ostatniej $|V^M| - 2$ iteracji zostanie spełniony warunek w wierszu 14, co spowoduje kolejne powtórzenie Kroku 2 dla cyklu bazowego v^M_i . Tym razem część iteracyjna będzie miała $|V^M| - 3$ iteracji.

W najgorszym przypadku liczba wywołań w Kroku 2 jest równa liczbie cykli bazowych sąsiednich względem cyklu bazowego v^M_i . W każdym takim uruchomieniu druga część iteracyjna zawiera o jedną iterację mniej niż w poprzednim uruchomieniu (maksymalna liczba iteracji wynosi $|V^M| - 1$ dla pierwszego uruchomienia Kroku 2). Z tego wynika, że maksymalna liczba iteracji dla cyklu bazowego v^M_i wynosi:

$$N = (|V^M| - 1) + (|V^M| - 2) + (|V^M| - 3) + \dots + 1 = \frac{|V^M| (|V^M| - 1)}{2} \quad (5.43)$$

Następnie przeanalizujemy przypadek, kiedy cykl bazowy v^M_i nie ma wspólnych krawędzi z żadnym z pozostałych cykli bazowych $|V(v^M_j)| < |V^M|$. W takim scenariuszu maksymalna liczba iteracji PPCB na cyklu bazowym v^M_i wynosi:

$$N(v^M_i) = \frac{|V(v^M_i)| (|V(v^M_i)| - 1)}{2} \quad (5.44)$$

Jeżeli następny cykl bazowy v^M_j wybrany poprzez procedurę ma wspólne krawędzie ze wszystkimi pozostałymi cyklami bazowymi mapy komórkowej to maksymalna liczba iteracji dla PPCB dla takiej mapy komórkowej wynosi:

$$N = N(v^M_i) + N(v^M_j) \quad (5.45)$$

Maksymalna liczba iteracji na cyklu bazowym v^M_j w pierwszym uruchomieniu Kroku 2 jest równa $|V^M| - |V(v^M_j)|$, ponieważ wszystkie cykle sąsiednie cyklu

bazowego v^{M_i} już są obecne w S_j . W takim razie maksymalna liczba iteracji na cyklu bazowym v^{M_j} wynosi:

$$N(v^{M_j}) = \frac{(|V^M| - |V(v^{M_i})|)(|V^M| - |V(v^{M_i})| - 1)}{2} \quad (5.46)$$

Biorąc pod uwagę wyrażenia (5.44)–(5.46), możemy obliczyć maksymalną liczbę iteracji dla takiej mapy komórkowej jako:

$$\begin{aligned} N &= \frac{|V(v^{M_i})|(|V(v^{M_i})| - 1)}{2} + \frac{(|V^M| - |V(v^{M_i})|)(|V^M| - |V(v^{M_i})| - 1)}{2} = \\ &= \frac{|V^M|(|V^M| - 1)}{2} - |V(v^{M_i})|(|V^M| - |V(v^{M_i})|) \end{aligned} \quad (5.47)$$

Z tego wynika, że w scenariuszu kiedy pierwszy cykl bazowy v^{M_i} ma wspólne krawędzie tylko z częścią cykli bazowych mapy komórkowej, a jeden z jego cykli sąsiednich ma wspólne krawędzie z wszystkimi pozostałymi cyklami bazowymi, maksymalna liczba iteracji PPCB jest mniejsza o $|V(v^{M_i})| (|V^M| - |V(v^{M_i})|)$ niż w przypadku, kiedy cykl bazowy v^{M_i} ma wspólne krawędzie z wszystkimi pozostałymi cyklami bazowymi mapy komórkowej (5.43). W scenariuszu kiedy drugi wybrany przez PPCB cykl bazowy v^{M_j} dla uruchomienia Kroku 2 nie ma dostępu do wszystkich pozostałych cykli bazowych, maksymalna liczba iteracji może być opisana jako:

$$N = N(v^{M_i}) + N(v^{M_j}) + \dots + N(v^{M_z}) \quad (5.48)$$

Jak widać z powyżej opasanych przyczyn, maksymalna liczba iteracji będzie malała ze wzrostem liczby cykli bazowych, na których zostanie uruchomiony Krok 2 PPCB. Z tego wynika, że algorytmiczna złożoność PPCB najgorszego przypadku jest zdefiniowana wzorem (5.43).

Ze względu na to, że liczba cykli bazowych w grafie zgodnie z [75] jest określona poprzez zależność (5.49), formułę (5.43) można wyrazić w funkcji liczby elementów grafu topologii sieci poprzez wyrażenie (5.50).

$$|V^M| = |E| - |V| + 1 \quad (5.49)$$

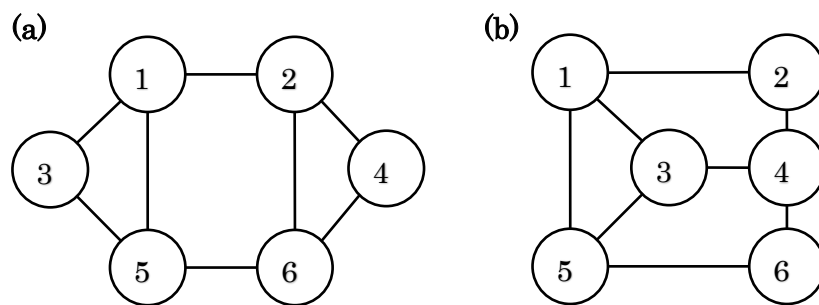


$$N = \frac{(|E| - |V| + 1) ((|E| - |V| + 1) - 1)}{2} = \frac{|E|(|E| + 1) + |V|(|V| - 1)}{2} - |V||E| \quad (5.50)$$

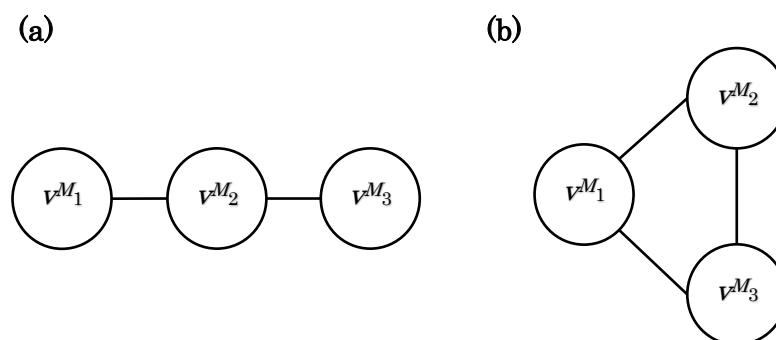
Złożoność algorytmiczna najgorszego przypadku nie jest bardzo przydatna w praktyce, gdyż w rzeczywistości trudno spotkać sieć, dla której jeden cykl bazowy ma wspólne krawędzie z pozostałymi cyklami bazowymi topologii sieci. Pomimo tego, warunek w wierszu 14 metody PPCB zostanie spełniony tylko w przypadku, kiedy sumaryczny koszt łączy sieci zawartych jednocześnie w cyklu bazowym v^M_j oraz cyklu S_i jest większy niż suma kosztów wszystkich pozostałych łączy cyklu bazowego v^M_j . W przypadku doboru tras ukierunkowanego na zrównoważenie obciążenia sieci możemy spróbować zignorować warunek w wierszu 14 metody PPCB dla uproszczonej analizy złożoności algorytmicznej, która jest bliższa rzeczywistej liczbie iteracji (pojedyncza iteracja dla każdej krawędzi mapy komórkowej):

$$N = |E^M| \quad (5.51)$$

Jednak nie jest możliwe wyrażenie liczby krawędzi mapy komórkowej poprzez elementy topologii sieci. Różne topologie sieci o takich samych liczbach węzłów i łączy sieci (Rys. 5.27) mogą mieć różną liczbę krawędzi mapy komórkowej (Rys. 5.28).



Rys. 5.27 Przykłady topologii sieci o równych liczbach elementów sieci



Rys. 5.28 Przykłady map komórkowych dla topologii sieci o takich samych liczbach elementów sieci

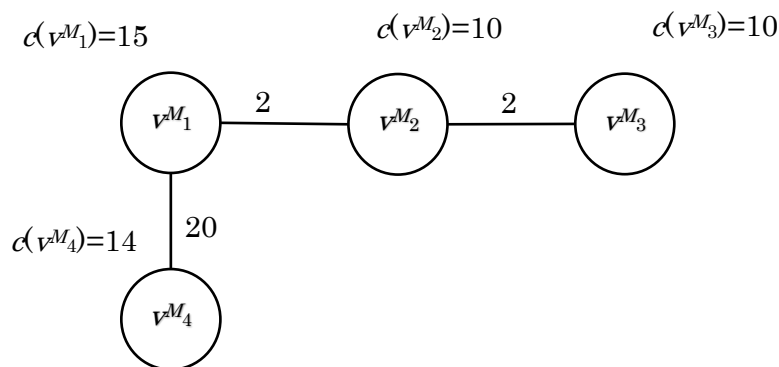
W takim razie złożoność algorytmiczna najgorszego przypadku dla PPCB można oszacować za pomocą wyrażenia (5.50) jako $O(|V|^4)$. Średnia oczekiwana liczba iteracji PPCB jest określona wzorem (5.51). Złożoność algorytmiczna najgorszego przypadku może być przeanalizowana bezpośrednio dla grafu topologii sieci. Z kolei, analiza średniej oczekiwanej liczby iteracji wymaga transformacji grafu topologii sieci do mapy komórkowej.



Aneks 7 – Warunek w wierszu 14 PPCB

W niniejszym aneksie zostały wyjaśnione zasady działania warunku w wierszu 14 algorytmu PPCB opisanego w podrozdziale 4.2.

Załóżmy, że mamy mapę komórkową zaprezentowaną na Rys. 5.29. Niech v^{M_1} jest źródłowym cyklem bazowym i v^{M_3} jest cyklem docelowym. W takim razie PPCB zdefiniuje cykl $S_1=(v^{M_1})$ i koszt $c(S_1)=c(v^{M_1})=15$. PPCB wykona po raz pierwszy Krok 2 o odniesieniu do cyklu bazowego v^{M_1} dla sąsiednich cykli bazowych: v^{M_2} i v^{M_4} .



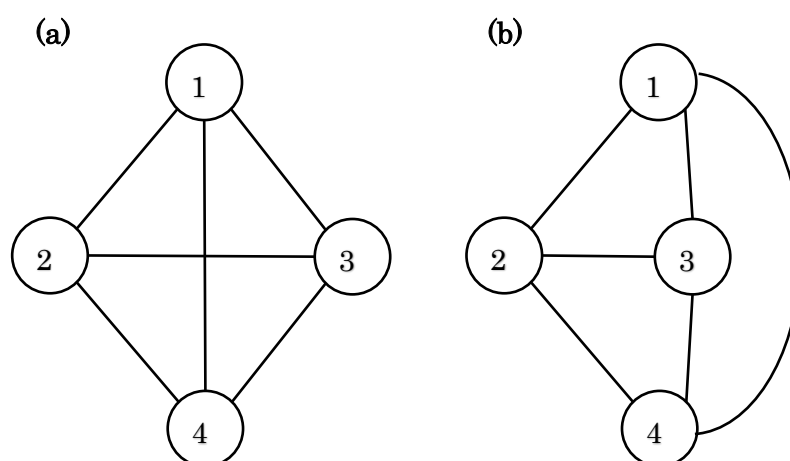
Rys. 5.29 Przykład mapy komórkowej

Nowy cykl S_2 o koszcie $c(S_2)=23$ zostanie obliczony dla cyklu bazowego v^{M_2} . Następnie zostanie wyznaczony cykl S_4 o koszcie $c(S_4)=9$ dla cyklu bazowego v^{M_4} . Jeżeli algorytm PPCB nie będzie zawierał warunku w wierszu 14, to koszt cyklu $S_3=(v^{M_1}, v^{M_2}, v^{M_3})$ dla cyklu bazowego v^{M_3} zostanie policzony jako $c(S_3)=c(S_2)+c(v^{M_3})-c(e^{M_2,3})=31$. Jednak, nie jest to cykl o najmniejszym koszcie. Jak możemy zauważyć, cyklem o najmniejszym koszcie będzie cykl $S_3=(v^{M_1}, v^{M_2}, v^{M_3}, v^{M_4})$ o koszcie $c(S_3)=25$. Warunek w wierszu 14 pozwala metodzie PPCB po wykryciu cyklu S_4 o koszcie $c(S_4)=9$ zaktualizować cykl S_1 jako: $S_1=(v^{M_1}, v^{M_4})$, $c(S_1)=9$ oraz wyznaczyć cykl S_1 na nowo. Tym razem cykl S_2 zostanie wyznaczony dla cyklu bazowego v^{M_2} jako: $S_2=(v^{M_1}, v^{M_2}, v^{M_4})$, $c(S_2)=c(S_1)+c(v^{M_2})-c(e^{M_1,2})=17$. Następnie zostanie wyznaczony cykl $S_3=(v^{M_1}, v^{M_2}, v^{M_3}, v^{M_4})$ o koszcie $c(S_3)=c(S_2)+c(v^{M_3})-c(e^{M_2,3})=25$.

W powyższy sposób warunek w wierszu 14 pozwala PPCB działać poprawnie w przypadku, kiedy sumaryczny koszt łączy cyklu bazowego v^M_j wspólnych z cyklami bazowymi w zbiorze S_i jest większy niż suma kosztu wszystkich pozostałych łączy cyklu bazowego v^M_j .

Aneks 8 – Modyfikacje PPCB

Prezentowany w podrozdziale 4.2 niniejszej pracy algorytm PPCB działa na grafach planarnych. Jednakże, nie znaczy to, że nie jest on przydatny wykorzystania w przypadku sieci o nieplanarnej topologii. Znaczną część grafów nieplanarnych można transformować do postaci planarnej (Rys. 5.30(a),(b)). Istnieje możliwość modyfikacji algorytmu, która pozwoli na wyznaczanie cykli w nieplanarnym grafie.

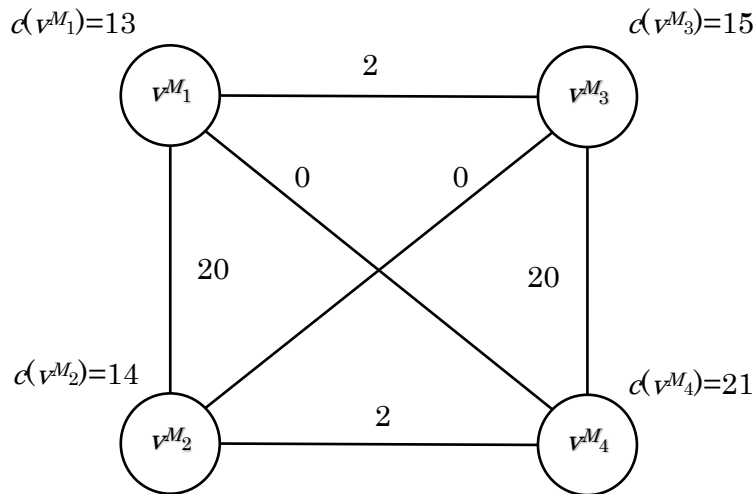


Rys. 5.30 Graf topologii sieci w wersji nieplanarnej i planarnej

Oryginalny algorytm PPCB nie może być wykorzystany w przypadku topologii, dla których nie jest możliwe przedzielenie łączy sieci tylko i wyłącznie do jednej krawędzi mapy komórkowej. Dla takich sieci metoda PPCB wymaga modyfikacji. Podczas sprawdzania warunku w wierszu 9 algorytmu PPCB powinny być sprawdzone również wszystkie łączy sieci zawarte w krawędziach mapy komórkowej wspólnych dla cykli bazowych zawartych w cyklu S_i i cyklu bazowym v_j . Takie krawędzie nie mogą należeć więcej niż do dwóch wymienionych powyżej cykli bazowych. Sprawdzanie tego warunku pozwala unikać połączenia cykli bazowych, które mają wspólne krawędzie, jednak nie tworzą cyklu poprzez usunięcie wspólnych łączy.

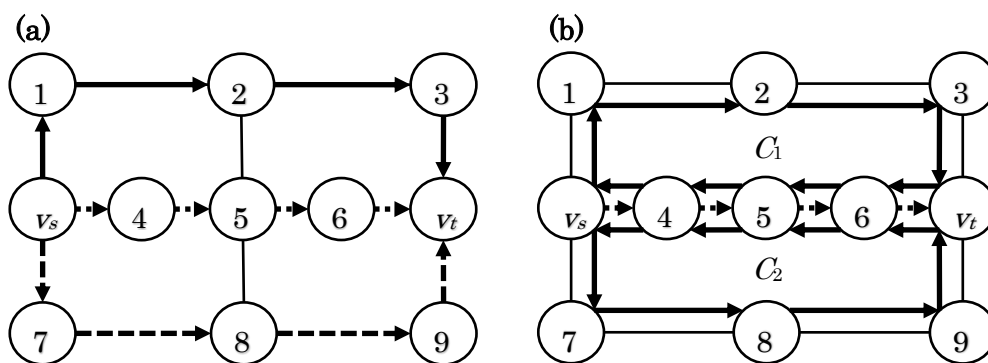
Oryginalny algorytm PPCB służy dla wykrycia pary węzłowo-rozłącznych tras. Jednak po modyfikacji mapy komórkowej istnieje możliwość wykrycia pary łączowo-rozłącznych tras. Taka modyfikacja wymaga wprowadzenia do mapy komórkowej dodatkowych krawędzi o zerowym koszcie pomiędzy cyklami bazowymi, które nie

mają wspólnych łączy sieci i mają wspólne węzły sieciowe. Przykład takiej mapy komórkowej dla topologii sieci z Rys. 4.16 jest zaprezentowany na Rys. 5.31:



Rys. 5.31 Przykład mapy komórkowej dla wykrycia łączowo-rozłącznych tras

W celu opracowania algorytmu doboru trzech węzłowo-rozłącznych tras warto obrać następujący kierunek. Faktycznie trzy pary węzłowo-rozłącznych tras pomiędzy węzłami v_s i v_t są równoważne parze cykli (każdy z których zawiera węzły v_s i v_t mających wspólne krawędzie, które składają trasę pomiędzy v_s i v_t (Rys. 5.32).



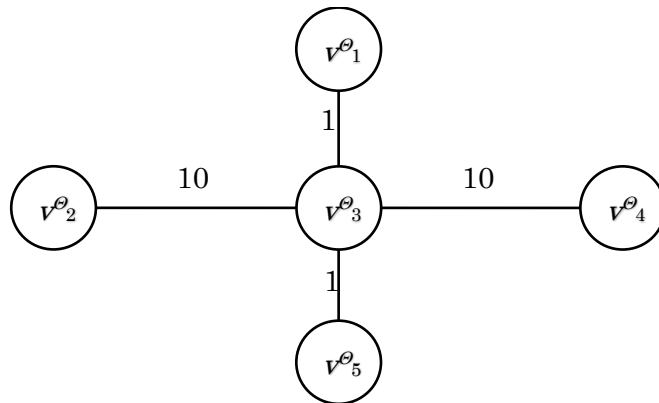
Rys. 5.32 Przykład trzech węzłowo-rozłącznych tras równoważnych dwóm cyklom

Suma kosztów trzech rozłącznych tras jest równa sumie kosztów dwóch takich cykli bez kosztu trasy pomiędzy nimi:

$$c(\Theta) = c(C_1) + c(C_2) + c(P) \quad (5.52)$$

Taki zbiór „motylkowy” Θ (dwa cykle i trasa pomiędzy nimi) może być wykorzystany dla wykrycia trzech węzłowo-rozłącznych tras. Zadanie wykrycia trójki węzłowo-rozłącznych tras pomiędzy węzłami v_s a v_t o minimalnym koszcie można transformować do zadania wykrycia zbioru motylkowego dla węzłów źródłowego v_s i docelowego v_t .

Mapa komórkowa zaprezentowana w rozdziale 4 nie wystarczy dla realizacji tego zadania. Dodatkową strukturą może być mapa motylkowa, która jest grafem $G^\Theta = (V^\Theta, E^\Theta)$ gdzie zbiór elementów V^Θ reprezentuje węzły sieci o stopniu 3 i wyższym, a zbiór E^Θ reprezentuje krawędzie mapy komórkowej łączące tamte węzły (Rys. 5.33).



Rys. 5.33 Przykład mapy motylkowej

Wykorzystanie mapy motylkowej pozwala wykryć trasy pomiędzy dwoma cyklami zbioru motylkowego. W tym samym momencie mapa komórkowa pozwala budować cykle na podstawie łączenia cykli bazowych.

W taki sposób opisana podejście (zbiór motylkowy oraz mapa motylkowa) jest perspektywnym kierunkiem badań dla projektowania algorytmów wykrywania trzech węzłowo-rozłącznych tras, gdzie PPCB teoretycznie może być wykorzystany w ramach takich algorytmów do wykrycia dwóch cykli zbioru motylkowego.

Powyższy opis kierunków rozwoju metod transformacyjnych miał na celu pokazanie, że zastosowanie metod transformacji grafu topologii sieci nie jest ograniczone tylko i wyłącznie do doboru par węzłowo-rozłącznych tras w grafach planarnych. Jednakże osiągnięcie innych celów wymaga modyfikacji metody PPCB, mapy komórkowej i dodatkowych badań.

Aneks 9 – Budowanie kwantów w metodzie PPCB

Istnieje możliwość wykorzystania metod opisanych w rozdziale 3 dla redukcji współczynnika blokowania żądań przy doborze par węzłowo-rozłącznych tras przez algorytm PPCB. Mimo tego, że w rozdziale 4 niniejszej pracy zostało opisane rozwiązanie ukierunkowane na redukcję współczynnika blokowania żądań poprzez redukcję czasu wyznaczenia par tras, metoda opisana w niniejszym aneksie jest przydatna dla ochrony ścieżki przed awariami pojedynczymi.

Metody opisane w rozdziale 3 wykorzystują trasy kwantowe pomiędzy parą węzłów sieci (kwanty). W celu ich wykorzystania w połączeniu z algorytmem PPCB musi istnieć metoda budowania cykli kwantowych. Aby wyznaczyć cykle kwantowe pomiędzy parą węzłów niezbędne jest wykrycie wszystkich istniejących cykli w grafie zawierających daną parę węzłów. Zbiór wszystkich cykli zawierających parę węzłów pozwala zbudować kwant dla tej pary w sposób analogiczny do opisanego dla tras pojedynczych w rozdziale 3, co z kolei pozwala zbudować wektor prognozowanych konfliktów i korzystać z tego wektora w odniesieniu do metryk kosztów łącza (3.2) i (3.32) dla algorytmu PPCB.

Każdy cykl w grafie, który możemy traktować jako oparty o zbiór cykli bazowych można opisać jako wektor \mathcal{E} , którego każdy element jest równy 1 w przypadku, gdy odpowiedni cykl bazowy mapy komórkowej jest zawarty w cyklu i jest równy 0 w przeciwnym przypadku. W takim razie możemy traktować wektor \mathcal{E} jako ciąg bitów i reprezentować cykl jako liczbę w systemie binarnym zwaną w niniejszej pracy numerem cyklu ψ . Wykorzystanie takich numerów cykli zamiast zbiorów A , B , S w algorytmie PPCB pozwoli zredukować czas doboru tras w porównaniu z wynikami opisanymi w podrozdziale 4.2, gdzie badania zostały prowadzone na zbiorach. Redukcja czasu obliczeń jest osiągnięta z powodu specyfiki działania procesorów i systemów komputerowych. Operacja wykrycia elementu w zbiorze jest bardziej czasochłonna w porównaniu do operacji bitowej wykrycia wartości wybranego bitu. Dotyczy to również operacji dodawania nowych elementów i usunięcia elementów ze zbiorów w porównaniu do ustawienia wartości wybranych bitów.

Warto zwrócić uwagę, że każdy cykl może być reprezentowany jako numer cyklu ψ . Jednak nie każdy numer cyklu ψ reprezentuje cykl. Jeżeli numer cyklu ψ

reprezentuje zbiór cykli bazowych, w którym cykle bazowe nie mają wspólnych krawędzi mapy komórkowej, to taki numer cyklu reprezentuje wiele cykli, a nie cykl pojedynczy. Istnienie cyklu S odpowiadającego numerowi cyklu ψ można sprawdzić poprzez uruchomienie na zawartym w S (reprezentowanym przez odpowiedni bit w numerze cyklu ψ) dowolnym cyklu bazowym mapy komórkowej algorytmu Dijkstry, BFS lub DFS. Jeżeli wszystkie pozostałe cykle bazowe w S (reprezentowane przez odpowiednie bity w numerze cyklu ψ) zostały osiągnięte przez algorytm, to taki numer cyklu ψ istnieje. W przypadku kiedy co najmniej jeden z cykli bazowych reprezentowany bitami numeru cyklu ψ nie został uzyskany przez algorytm, taki numer cyklu ψ nie istnieje (ψ reprezentuje wtedy dwa lub więcej cykli).

W celu budowania kwantów wymagane jest wykrycie wszystkich istniejących cykli w grafie. Maksymalna liczba istniejących cykli w grafie (maksymalna liczba istniejących numerów cykli) wynosi:

$$|S| = \psi_{max} = 2^{|V^M|} \quad (5.53)$$

W celu oszczędzania pamięci utworzymy ciąg elementów bitowych Φ o $2^{|V^M|}$ bitów (zwany w niniejszej pracy ciągiem numerów cykli), którego każdy element ϕ_i jest równy 1 jeżeli odpowiedni numer cyklu ψ_i istnieje i jest równy 0 jeżeli odpowiedni ψ_i nie istnieje. Dla obliczenia Φ sprawdzimy wszystkie numery cykli od 1 do $2^{|V^M|}$ pod względem ich istnienia w grafie (poprzez uruchomienie algorytmu Dijkstry, BFS lub DFS na dowolnym cyklu mapy komórkowej zawartym w cyklu S_i reprezentowanym przez numer cyklu ψ_i).

Mając zbudowany ciąg numerów cykli Φ , możemy zidentyfikować wszystkie cykle w grafie, które zawierają wybrane cykle bazowe v^{M_i} i v^{M_j} . Wystarczy w tym celu sprawdzić wartości wszystkich elementów ciągu numerów cykli Φ , których indeksy w systemie dwójkowym mają i -te i j -te bity równe 1. Jeżeli takie elementy ciągu Φ mają wartość 1, znaczy to, że takie cykle istnieją w grafie i możemy budować kwanty w sposób analogiczny do opisanego dla tras pojedynczych w podrozdziale 3.1. Pomimo tego możemy obliczyć koszt takich cykli za pomocą formuły (4.2). Pozwala to sortować na bieżąco istniejące cykle według ich kosztu przechowując w pamięci tylko k najkrótszych cykli. W taki sposób możliwe jest zbudowanie kwantu w sposób analogiczny do opisanego dla tras pojedynczych w podrozdziale 3.2.



Warto zwrócić uwagę, że zadanie sprawdzania wartości wszystkich elementów w Φ , których indeksy reprezentowane w postaci dwójkowej zawierają jedynki na wybranych pozycjach bitowych, może wydawać się czasochłonne i wymagające długich obliczeń. Jednakże dla systemów komputerowych pracujących na podstawie algebry Boole'a, zadanie to jest naturalne i nie wymaga dodatkowych obliczeń w przypadku kodu napisanego w assemblerze lub w systemowym języku programowania (C, C++, Rust, itp.).

Przechowywanie ciągu numerów cykli Φ w pamięci podczas działania algorytmu PPCB i po doborze tras jest przydatne w przypadku wielokrotnych awarii elementów sieci. Pozwala to szybko zbudować kwanty na nowo bez wielokrotnego uruchomienia algorytmu Dijkstry, DFS lub BFS ($2^{|\mathcal{V}^M|}$ razy) dla ponownego zbudowania ciągu numerów cykli Φ . Jednak, dla powtórnego wykorzystania Φ po awarii elementów sieci, niezbędne jest sprawdzanie każdego istniejącego cyklu na obecność elementów uległych awarii. Ten fakt wydłuża czas wymagany do wyznaczenia kwantów. Wpływ tego problemu można z kolei zredukować za pomocą opisanych w Aneksie 10 metod szybkiej aktualizacji mapy komórkowej w przypadku zmiany topologii sieci. Można nie sprawdzać cykli, które zawierają usunięte z mapy komórkowej cykle bazowe i traktować komórki, które powstały poprzez połączenie innych cykli bazowych w wyniku awarii, nadal jako zbiór tych cykli bazowych przed awarią.

Warto zwrócić uwagę, że przechowywanie ciągu numerów cykli Φ dla powtórnego zbudowania kwantów jest przydatne dla małych sieci o małej liczbie cykli bazowych. Pamiętając o liczbie numerów cykli określonej wyrażeniem (5.53), można dojść do wniosku, że dla takich sieci przechowywanie ciągu numerów cykli Φ nie wymaga dużej ilości pamięci. Na przykład przechowanie Φ dla sieci o 12 cyklach bazowych wymaga tylko 0.5 kB pamięci (wobec 1 GB dla sieci o 33 cyklach bazowych). Z kolei, aby sprawdzić obecność wszystkich możliwych cykli zawierających parę cykli bazowych w takiej sieci, wymagane są 1 024 odczyty elementów Φ (w porównaniu z 1 072 741 824 odczytami dla sieci o 33 cyklach bazowych).

Tak czy inaczej, czas wymagany do zbudowania kwantów po awarii zwiększa czas obliczenia nowych tras. Dlatego jako rozwiązanie została w niniejszej pracy wybrana mniej czasochłonna metoda o mniejszych wymaganiach pamięciowych (opisana w rozdziale 4). Z tego powodu metody opisane w niniejszym aneksie nie zostały dalej przebadane i nie zostały umieszczone w głównej części pracy. Jednakże, opisane powyżej założenie wskazuje na możliwość wykorzystania hybrydowego podejścia

(połączenia metod opisanych w rozdziałach 3 i 4) jako osobnej metody ochrony transmisji przed awariami pojedynczymi i dla ochrony sieci przed awariami wielokrotnymi w sieciach o małej liczbie cykli bazowych.

Aneks 10 – Sposoby szybkiej aktualizacji mapy komórkowej po zmianie topologii sieci

Pod pojęciem zmiany topologii sieci w niniejszym aneksie jest rozumiane dodawanie nowych elementów (węzłów/łączy) i połączenia ich z istniejącymi węzłami topologii sieci, usuwanie elementów z topologii sieci, niezdolność elementów sieci do zapewniania transmisji (dalej również zwana usunięciem elementów).

W odniesieniu do metody z rozdziału 4, szybka aktualizacja mapy komórkowej jest możliwa w następujących scenariuszach:

1. Usunięcie łącza sieci nienależącego do krawędzi mapy komórkowej.
2. Usunięcie węzła, który nie ma incydentnych łączy sieci należących do krawędzi mapy komórkowej.
3. Usunięcie węzła, który ma incydentne łącza sieci należące do krawędzi mapy komórkowej, jak również łącza należące do krawędzi mapy komórkowej.
4. Dodawanie nowych elementów łączących węzły należące do jednego cyklu bazowego.
5. Dodawanie nowych elementów łączących węzły należące do różnych cykli bazowych.

W scenariuszu 1 cykl bazowy, do którego należy takie łącze nie może nadal istnieć. Usuń cykl bazowy z mapy komórkowej. Należy usunąć z mapy komórkowej wszystkie krawędzie incydentne dla tego cyklu bazowego.

W scenariuszu 2 usunięcie takiego cyklu doprowadzi do niemożliwości występowania krawędzi zawierających taki węzeł. Należy usunąć wszystkie krawędzie mapy komórkowej, do których należy taki węzeł; połączyć w jeden cykl bazowy cykle bazowe, które są incydentne dla usuniętych krawędzi; obliczyć koszt nowego cyklu bazowego jako $\alpha(S)$ (gdzie S zawiera połączone cykle bazowe) poprzez wykorzystanie formuły (4.2).



W scenariuszu 3 cykle bazowe do których należy taki węzeł nie mogą istnieć. Należy usunąć takie cykle bazowe z mapy komórkowej oraz usunąć z mapy komórkowej wszystkie krawędzie incydentne względem takich cykli.

W scenariuszu 4 podziel każdy taki cykl bazowy na dwie części w węzłach, które zostały połączone nowymi elementami. Zbuduj na podstawie każdej części cyklu i nowych elementów dwa nowe cykle bazowe. Oblicz koszty nowych cykli bazowych. Dodaj wspólną dla nowych cykli bazowych krawędź mapy komórkowej zawierającą nowe elementy sieci. Ustaw koszt nowej krawędzi jako podwójną sumę kosztów wszystkich dodanych łączy sieci. W przypadku podziału istniejących krawędzi mapy komórkowej pomiędzy nowe cykle bazowe, dodaj do mapy komórkowej brakujące krawędzie i zaktualizuj ich koszty.

W scenariuszu 5 usuń z grafu topologii sieci wszystkie łącza sieci należące do wszystkich krawędzi mapy komórkowej i nowe dodane elementy. Na jednym węźle połączonym nowymi elementami uruchom algorytm Dijkstry. Jeżeli algorytm Dijkstry wykrył najkrótszą trasę (o najmniejszej liczbie przeskoków) do drugiego węzła połączonego nowymi elementami, zbuduj nowy cykl bazowy z nowych elementów i wykrytej trasy. Oblicz koszt nowego cyklu bazowego. Dodaj cykl bazowy do mapy komórkowej. Wyznacz nowe krawędzie mapy komórkowej wspólne dla nowego cyklu bazowego i jego sąsiadów (nowy cykl bazowy ma wspólne łącza z innymi cyklami bazowymi; wspólne łącza sieciowe są zawarte w wykrytej przez algorytm Dijkstry trasie). Dodaj nowe krawędzie do mapy komórkowej.

Warto zwrócić uwagę, że w przypadku niemożliwości wykrycia trasy przez algorytm Dijkstry, nie jest możliwa szybka aktualizacja mapy komórkowej. Wymaga to obliczenia mapy komórkowej na nowo. Obliczenie na nowo nie gwarantuje jednak możliwości działania PPCB na takiej mapie, ponieważ niemożliwość wykrycia trasy może świadczyć o tym, że dodanie nowych elementów do sieci zmodyfikowało graf topologii sieci czyniąc go grafem nieplanarnym. Oryginalny algorytm PPCB nie operuje na grafach nieplanarnych. Z kolei, w przypadku modyfikacji PPCB w celu umożliwienia jego działania na nieplanarnych grafach (opisane w Aneksie 8), w scenariuszu 5 nie jest wymagane usunięcie z grafu topologii sieci wszystkich łączy sieci należących do wszystkich krawędzi mapy komórkowej i nowych dodanych elementów przed uruchomieniem algorytmu Dijkstry.

Spis bibliografii

- [1] Coffman K., Odlyzco A.: “Optical Fiber Telecommunications IV-B: Systems and Impairments. Optics and Photonics”, Academic Press, 2015.
- [2] Grobe K., Eiselt M.: “Wavelength Division Multiplexing: A Practical Engineering Guide”, Wiley, 2013.
- [3] Laude J.P.: “Wavelength Division Multiplexing”, Prentice-Hall, 1993.
- [4] Chen R.T., Lome L.S.: “Wavelength Division Multiplexing”, Society of Photo-optical Instrumentation Engineers, 1999.
- [5] ITU-T Rec. G.707/Y.1322, “Network Node Interface for the Synchronous Digital Hierarchy (SDH)”. International Telecommunications Union, 2010.
- [6] Gumaste A., Antony T.: “DWDM Network Designs and Engineering Solutions”, Cisco Press, 2002.
- [7] Brackett C.A.: “Dense Wavelength Division Multiplexing Networks: Principles and Applications”, IEEE Journal on Selected Areas in Communications, vol. 8, no. 6, pp. 948-964, 1990.
- [8] ITU-T Rec. Y.1540, “Internet Protocol Data Communication Service – IP Packet Transfer and Availability Performance Parameters”. International Telecommunications Union, 2016.
- [9] ITU-T Rec. Y.1541, “Network Performance Objectives for IP-based Services”. International Telecommunications Union, 2003.
- [10] Szigeti T., Hattingh Ch.: “Quality of Service Design Overview, End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs”, Cisco Press, 2004.
- [11] Cacheda R.A., García D.C., Cuevas A., Pantò A.: “QoS Requirements for Multimedia Services, Resource Management in Satellite Networks”, In: Giambene G. (eds) Resource Management in Satellite Networks. Springer, Boston, Springer, 2007.
- [12] Chiesa M., Kamisiński A., Rak J., Rétvári G., Schmid S. “A Survey of Fast-Recovery Mechanisms in Packet-Switched Networks”, IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1253-1301, 2021.
- [13] Rak J., Hutchison D. (Eds.): “Guide to Disaster-Resilient Communication Networks”, Springer, 2020.
- [14] Gomes T., Tapolcai J., Esposito C., Hutchison D., Kuipers F., Rak J., de Sousa A., Iossifides A., Travanca R., André J., Jorge L., Martins L., Ugalde P. O., Pašić A., Pezaros D., Jouet S., Stefano S., Tornatore M. “A Survey of Strategies for Communication Networks to Protect against Large-scale Natural Disasters”, in Proc. 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), pp. 11-22, 2016.

- [15] Mauthe A., Hutchison D., Çetinkaya E. K., Ganchev I., Rak J., Sterbenz J.P.G., Gunkel M., Smith P., Gomes T. “Disaster-resilient Communication networks: Principles and Best Practices”, in Proc. 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), pp. 1-10, 2016.
- [16] Molisz W.: „Przeżywalność sieci teleinformatycznych i telekomunikacyjnych, Politechnika Gdańska”, 2002.
- [17] Mukherjee B.: “Optical WDM Networks”, Springer, 2006.
- [18] Zhou D., Subramaniam S.: “Survivability in Optical Networks”, IEEE Network. vol. 14, no. 6, pp. 16-23, 2000.
- [19] Rak J.: “Resilient Routing in Communication Networks”, Springer, 2015.
- [20] Diestel R. “Graph Theory, Electronic Edition”, NY: Springer-Verlag, 2005.
- [21] Sundararajan D.: “The Discrete Fourier Transform: Theory, Algorithms and Applications”, World Scientific, 2001.
- [22] Steiglitz K.: “A Digital Signal Processing Primer: With Applications to Digital Audio and Computer Music”, Dover Publications, 2020.
- [23] Leander G.: “On Linear Hulls, Statistical Saturation Attacks, PRESENT and a Cryptanalysis of PUFFIN”, in: Paterson K.G. (eds) Advances in Cryptology – EUROCRYPT 2011. Lecture Notes in Computer Science, vol. 6632, 2011.
- [24] Lyubashevsky V., Micciancio D., Peikert C., Rosen A.: “SWIFFT: A Modest Proposal for FFT Hashing”, Fast Software Encryption, vol. 5086, pp. 54-72, 2008.
- [25] Raid A.M., Khedr W.M., El-dosuky M.A., Wesam A.: “JPEG Image Compression Using Discrete Cosine Transform – A Survey”, International Journal of Computer Science & Engineering Survey. vol. 5, no. 2, pp. 39-47, 2014.
- [26] Bentley P.M., McDonnell J.T.E.: “Wavelet Transforms: An Introduction”, Electronics & Communication Engineering Journal. vol. 6, no. 4, pp. 175-186, 1994.
- [27] Burrus C.S., Gopinath R.A., Guo H.: “Introduction to Wavelets and Wavelet Transforms”, A Primer, Prentice Hall, 1998.
- [28] Mann S., Haykin S.: “The Chirplet Transform: Physical Considerations”, IEEE Transactions on Signal Processing. vol. 43, no. 11, pp. 2745-2761, 1995.
- [29] Perello J. *et al.*: “Experimental Demonstration of a GMPLS-enabled Impairment-aware Lightpath Restoration Scheme”, IEEE/OSA Journal of Optical Communications and Networking, vol. 4, no. 5, pp. 344-355, 2012.
- [30] Cholda P., Jajszczyk A.: “Recovery and Its Quality in Multilayer Networks”, IEEE/OSA Journal of Lightwave Technology, vol. 28, no. 4, pp. 372-389, 2010.
- [31] Rak J.: “Fast Service Recovery Under Shared Protection in WDM Networks”, Journal of Lightwave Technology, vol. 30, no. 1, pp. 84-95, 2011.



- [32] Vance K.: "The Time Division Multiplexing Handbook", Emereo Pty Limited, 2016.
- [33] Saleh A.A.M., Simmons J.M.: "All-Optical Networking – Evolution, Benefits, Challenges, and Future Vision", Proceedings of the IEEE, vol. 100, no. 5, pp. 1105-1117, 2012.
- [34] Ozdaglar A., Bertsekas D.: "Routing and Wavelength Assignment in Optical Networks", IEEE/ACM Transactions on Networking, vol. 11, no. 2, pp. 259-272, 2003.
- [35] Marković G.Z., Aćimović-Raspopović V.S.: "Solving the RWA Problem in WDM Optical Networks Using the BCO Meta-Heuristic", Telfor Journal, vol. 2, no. 1, pp. 43-48, 2010.
- [36] Nace D., Pióro M.: "Max-Min Fairness and Its Applications to Routing and Load-Balancing in Communication Networks", IEEE Communications Surveys & Tutorials, vol.10, no. 4, pp. 5-17, 2008.
- [37] Klinkowski M., Ruiz M., Velasco L., Careglio D., Lopez V., Comellas J.: "Elastic Spectrum Allocation for Time-Varying Traffic in FlexGrid Optical Networks", IEEE Journal on Selected Areas in Communications, vol. 31, no. 1, pp. 26-38, 2013.
- [38] Biernacka E., Lasoń A.: "Elastyczne sieci optyczne", Telekomunikacja I Techniki Informacyjne, vol. 1-2/2013, pp. 5-17, 2013.
- [39] Weinstein S.: "The History of Orthogonal Frequency-Division Multiplexing", IEEE Communications Magazine, vol. 47, no. 11, pp. 26-35, 2009.
- [40] Geiregat P., Houtepen A., Van Thourhout D., Hens Z.: "All-Optical Wavelength Conversion by Picosecond Burst Absorption in Colloidal PbS Quantum Dots", ACS NANO, Optoelectronics Materials Section, vol. 10, no. 1, pp. 1265-1272, 2016.
- [41] Zhao X., Lou C., Pan S.: "All-Optical Wavelength Conversion of Nonreturn-to-zero Signals Based on a Nonlinear Polarization Switch and Delayed Interferometer", Optical Engineering, vol. 49, no. 11, pp. 110502-1–110502-3, 2016.
- [42] Kreutz D., Ramos F.M.V., Veríssimo P., Rothenberg C., Azodolmolky S., Uhlig S.: "Software-Defined Networking: A Comprehensive Survey", Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, 2014.
- [43] Grežo R., Nagy M.: "Network Traffic Measurement and Management in Software Defined Networks", in Proc. 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 541-546, 2017.
- [44] Nadar S., Chaudhari S.: "Proactive-routing Path Update in Software Defined Networks (SDN)", in Proc. 2017 International Conference on Intelligent Computing and Control (I2C2), pp.1-3, 2017.
- [45] Neghabi A., Navimipour N., Hosseinzadeh M., Rezaee A.: "Load Balancing Mechanisms in the Software Defined Networks": A Systematic and Comprehensive Review of the Literature, IEEE Access, vol. 6, pp. 14159-14178, 2018.

- [46] Hu T., Guo Z., Yi P., Baker T., Lan J.: "Multi-controller Based Software-Defined Networking: A Survey", IEEE Access. vol. 6, pp. 15980-15996, 2018.
- [47] Buckman L.A., Wu M.S., Giaretta G., Li G.S., Pepeljugoski P.K., Goodman J.W., Varma A., Lau K.Y., Chang-Hasnain C.J.: "A novel all-optical self-routed wavelength-addressable network (SWANET)", IEEE Photonics Technology Letters. vol. 7, no. 9, pp. 1066-1068, 1995.
- [48] Myslitski K., Rak J.: "A New Quantum-Inspired Approach to Reduce the Blocking Probability of Demands in Resource-constrained Path Computation Scenarios", Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne, nr. 8-9, pp. 728-734, 2016.
- [49] Liu W., Tao Y., Liu L.: "Load-balancing Routing Algorithm Based on Segment Routing for Traffic Return in LEO Satellite Networks" IEEE Access, vol. 7 p.112044-112053, 2019.
- [50] Dijkstra E.W.: "A Note on Two Problems in Connexion with Graphs", Numerische Mathematik, Springer Science+Business Media, vol. 1, pp. 269-271, 1959.
- [51] Pióro M., Medhi D.: "Routing, Flow, and Capacity Design in Communication and Computer Networks", Elsevier, 2004.
- [52] Phillips A.: "Introduction to Quantum Mechanics", John Willey & Sons, 2013.
- [53] Binney J., Skinner D.: "The Physics of Quantum Mechanics", Oxford University Press, 2008.
- [54] Von Neumann J.: "Mathematical Foundations of Quantum Mechanics", Princeton University Press, 1955.
- [55] Giacosa F.: "On Unitary Evolution and Collapse in Quantum Mechanics", Quanta, vol. 3, no. 1, pp. 156-170, 2014.
- [56] Even S.: "Graph Algorithms", 2nd Edition, Cambridge University Press, 2011.
- [57] Myslitski K., Rak J., Kuszner Ł.: "Network Graph Transformation Providing Fast Calculation of Paths for Resilient Routing", in Proc. 2016 8th International Workshop on Resilient Network Design and Modeling (RNDM), pp. 238-244, 2016.
- [58] Myslitski K.: "Metoda redukcji prawdopodobieństwa blokowania żądań w sieci szkieletowej oparta o mechanizm prognozowania konfliktów w zasobach przepustowości", Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne, vol. 8-9, pp. 715-718, 2017.
- [59] Ullrich A., Forst Ch.V.: " k -PathA: k -shortest Path Algorithm", in Proc. 2009 International Workshop on High Performance Computational Systems Biology, pp. 22-30, 2009.
- [60] Kusetogullari H., Leeson M.S., Ren W., Hines E.L.: " K -shortest Path Network Problem Solution with a Hybrid Genetic Algorithm: Particle Swarm Optimization Algorithm", in Proc. 13th International Conference on Transparent Optical Networks (ICTON 2011), pp. 1-4, 2011.

- [61] Scano G., Huguet M.-J., Ngueveu S.U.: “Adaptations of k -shortest Path Algorithms for Transportation Networks”, in Proc. International Conference on Industrial Engineering and Systems Management, pp. 663-669, 2016.
- [62] Kwon J., Lee J., Yu M., Park H.: “Automatic Classification of Network Traffic Data based on Deep Learning in ONOS Platform”, in Proc. 2020 International Conference on Information and Communication Technology Convergence (ICTC 2020), pp. 1028-1030, 2020.
- [63] Kwon J., Jung D., Park H.: “Traffic Data Classification Using Machine Learning Algorithms in SDN Networks”, in Proc. 2020 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1031-1033, 2020.
- [64] Holanda Filho R., Fontenelle do Carmo M.F., Maia J.E.B., Paulino Siqueira G.: “An Internet Traffic Classification Methodology Based on Statistical Discriminators”, in Proc. 2008 IEEE Network Operations and Management Symposium (NOMS), pp. 907-910, 2008.
- [65] Suurballe J., Tarjan R.: “A Quick Method for Finding Shortest Pairs of Disjoint Paths”, *Networks*, vol. 14, no. 2, pp. 325-336, 1984.
- [67] Bhandari R.: “Survivable Networks: Algorithms for Diverse Routing”, Kluwer Academic Publishers, 1999.
- [68] Cormen T., Leiserson C., Rivest R., Steain C.: “Introduction to Algorithms”, 3rd Edition, The MIT Press, 2009.
- [69] Myslitski K., Rak J.: “Metoda szybkiego wyznaczania par węzłoworozłącznych tras dla ochrony transmisji unicast”, *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne*, vol. 8-9, pp. 717-723, 2015.
- [70] Myslitski K., Rak J., Kuszner Ł.: “Toward Fast Calculation of Communication Paths for Resilient Routing”, *Networks*, vol. 70, no. 4, pp. 308-326, 2017.
- [71] Myslitski K., Rak J.: “An Approach to Improve the Time Efficiency of Disjoint Paths Calculation”, in Proc. 2015 17th International Conference on Transparent Optical Networks (ICTON), pp. 1-4, 2015.
- [72] Myslitski K., Rak J.: “Evaluation of Time-efficiency of Disjoint Paths Calculation Schemes”, in Proc. 2015 17th International Conference on Transparent Optical Networks (ICTON), pp. 1-4, 2015.
- [73] Bona M.: “Combinatorics of Permutations”, Chapman and Hall/CRC, 2012.
- [74] Amaldi E., Iuliano C., Rizzi R.: “Efficient Deterministic Algorithms for Finding a Minimum Cycle Basis in Undirected Graphs”, *Integer Programming and Combinatorial Optimization*, Springer, vol. 6080, pp. 397-410, 2010.
- [75] Berge C.: “The Theory of Graphs”, Courier Dover Publications, New York, pp. 27-30, 2001.