

Sprawozdanie z drugiej listy zadań na laboratoria

Technologie Sieciowe, semestr letni 2017/2018

Mikołaj Pietrek

do kursu mgr. inż. Ilyi Hradovicha

1 Szacowanie niezawodności

1.1 Metodyka

Rozwiązanie wszystkich podpunktów stanowi program napisany w języku Python. Wykorzystuje metodę Monte Carlo i Mersenne Twister. Pozwala na szacowanie niezawodności jednej ze zdefiniowanych topologii. Wybór jest dokonywany po uruchomieniu programu. Alternatywnie można wczytać listę krawędzi z pliku. Pozostałe dane wejściowe to ilość powtórzeń oraz czas odświeżania.

1.1.1 Algorytm

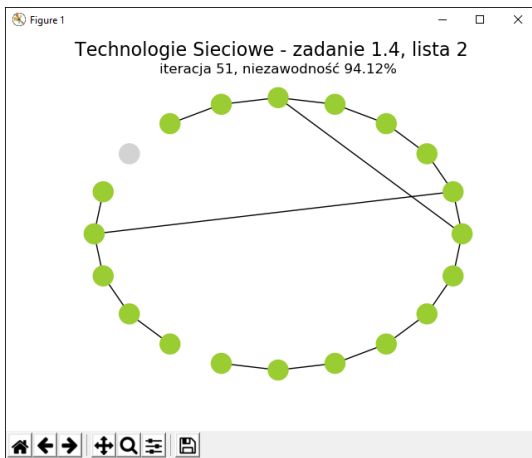
1. Dla każdej krawędzi wylosuj liczbę z zakresu $[0, 1]$, jeśli jest większa od niezawodności, usuń krawędź.
2. Jeśli graf pozostał spójny, zwiększ licznik poprawnych iteracji.
3. Zwiększ licznik wszystkich iteracji, przywróć usunięte krawędzie.
4. Powtórz punkty 1-3.

1.1.2 Tryb graficzny

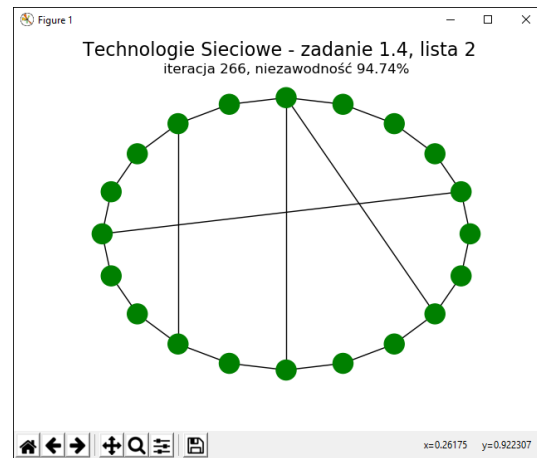
Określenie czasu z zakresu $[0.1 - 5.0]$ sekund powoduje włączenie wizualizacji. Wyświetlane są: numer listy i zadania, licznik iteracji, częściowa niezawodność, bieżący schemat sieci. Nieprawne krawędzie zostają ukryte. W przypadku uszkodzenia grafu, kolorem jasnozielonym jest oznaczany największy spójny podgraf. Zachowanie spójności sygnalizuje kolor ciemnozielony. Przykład uruchomienia:

```
D:\ts\l2\sieciowe_2_1.py
Podaj ilosc powtorzen (1-1000000000): 10000
Podaj czas odswiezania (0.0-5.0): 0.1
Wczytac dane z pliku? (T/N) N
Podaj numer podpunktu (1-4): 4

Prosze czekac, trwa losowanie...
```



(a) Graf rozspójniony



(b) Graf spójny

Rysunek 1: Przykładowa wizualizacja

1.1.3 Tryb tekstowy

Wybór zerowego czasu odświeżania przełącza program w tryb tekstowy. W trakcie obliczeń nie są wyświetlane wtedy żadne informacje. Przy taktowaniu 3.2 Ghz, średni czas iteracji wynosił 0,00025s. Natomiast dla symulacji graficznej minimalna wartość to 0,1s. Szybsze generowanie staje się bezcelowe, ze względu na możliwości oka ludzkiego. Wyłączenie wizualizacji skutkuje około 400 razy krótszym czasem uzyskania żądanej dokładności. Przykład:

```
D:\ts\12\sieciowe_2_1.py
Podaj ilosc powtorzen (1-1000000000): 10000
Podaj czas odswiezania (0.0-5.0): 0
Wczytac dane z pliku? (T/N) N
Podaj numer podpunktu (1-4): 1

Prosze czekac, trwa losowanie...

iteracje: 10000, czas: 2.1s, niezawodnosc: 38.37%
```

1.2 Rezultaty

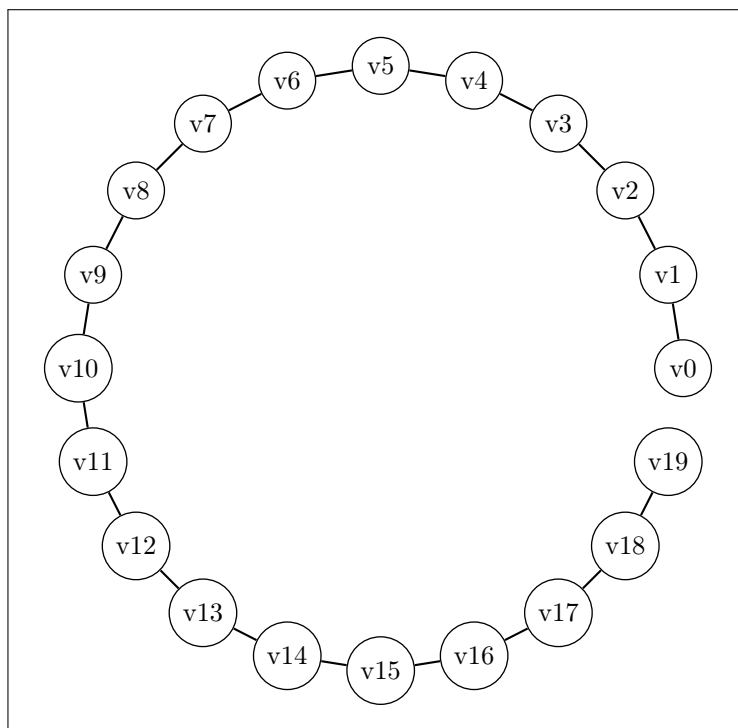
Wszystkie poniższe wyniki zostały otrzymane dla 1000000 iteracji.

1.2.1 Graf liniowy

Rozważana topologia powstaje w wyniku połączenia wierzchołków szeregowo przy niezawodności jednostkowej 95%.

```
iteracje: 1000000, czas: 227.9s, niezawodnosc: 37.73%
```

Otrzymany wynik 37% wyklucza praktyczne zastosowanie przedstawionej topologii. Z definicji, usunięcie dowolnej krawędzi prowadzi do rozspójnienia. Najważniejszą zaletą jest niski koszt.



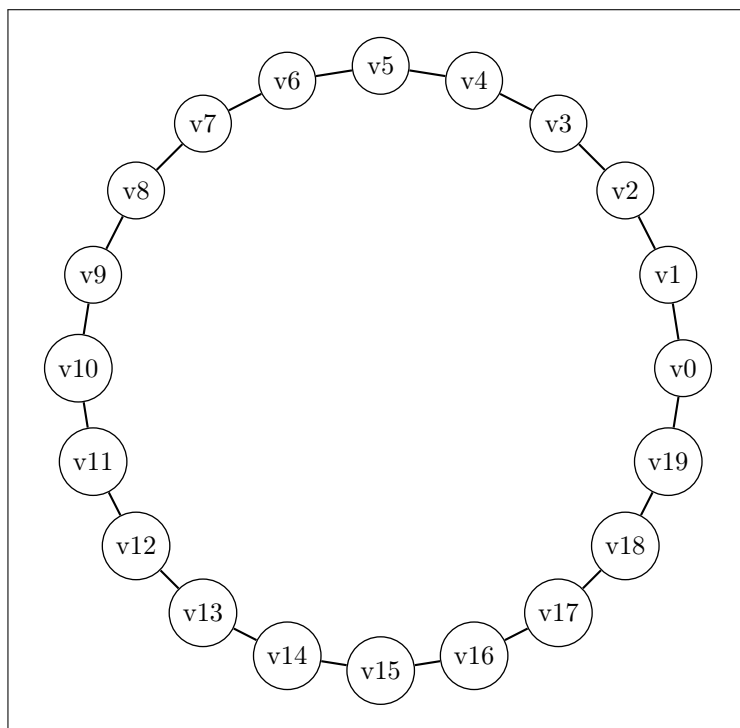
Rysunek 2: Graf liniowy $|V| = 20, |E| = 19$.

1.2.2 Graf cykliczny

Złączenie skrajnych wierzchołów grafu liniowego powoduje powstanie cyklu.

iteracje: 1000000, czas: 227.7s, niezawodność: 73.63%

Pojedyncza awaria rozrywa cykl. Warunkiem koniecznym rozspójnienia są co najmniej dwa uszkodzenia. Dodatkowa krawędź zwiększyła niezawodność do około 74%.



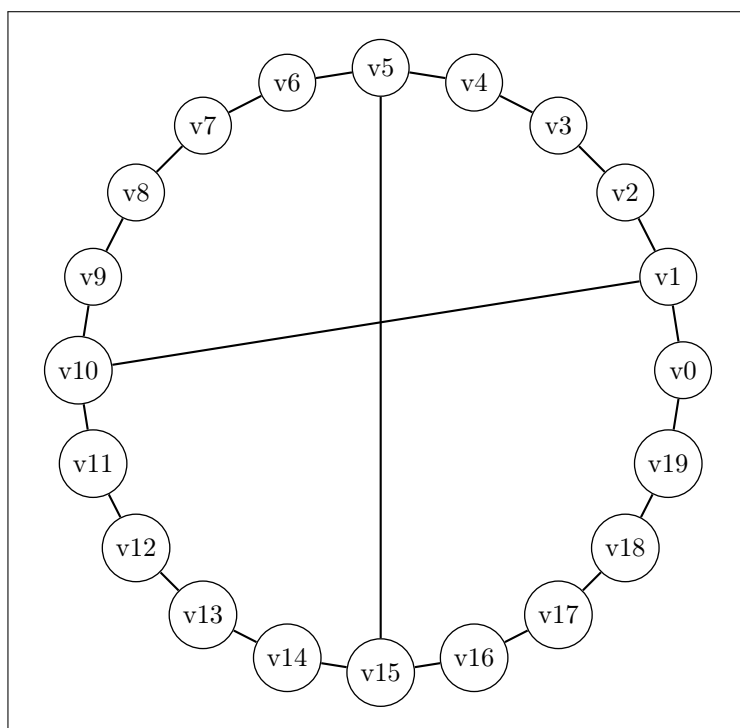
Rysunek 3: Graf cykliczny $|V| = 20, |E| = 20$.

1.2.3 Graf cykliczny z dwoma dodatkowymi krawędziami

Połączenie naprzeciwległych wierzchołków dwiema krawędziami o niezawodnościach 0.8 i 0.7 tworzy dodatkowe (nadmiarowe) kanały komunikacyjne.

iteracje: 1000000, czas: 244.5s, niezawodnosc: 87.07%

W przypadku pesymistycznym do rozspójnienia grafu nadal wystarczą dwie awarie. Wariant optymistyczny stanowi przerwanie sieci po czwartym uszkodzeniu. Zgodny z tym jest rezultat 87%.



Rysunek 4: Graf cykliczny rozszerzony o dwie krawędzie; $|V| = 20, |E| = 22$.

1.2.4 Graf cykliczny z sześcioma dodatkowymi krawędziami (przykład)

Graf z poprzedniego podpunktu rozszerzony o cztery losowe krawędzie, przy czym niezawodność nowych połączeń wynosi 40%. Treść zadania nie zawiera szczegółowych informacji na temat sposobu losowania. W celu przetestowania konkretnego zestawu można wykorzystać wczytywanie topologii z danych zewnętrznych. Przykład: (3, 8), (8, 13), (13, 18), (3, 18). Pomocniczy skrypt w języku Python, generujący zgodną listę:

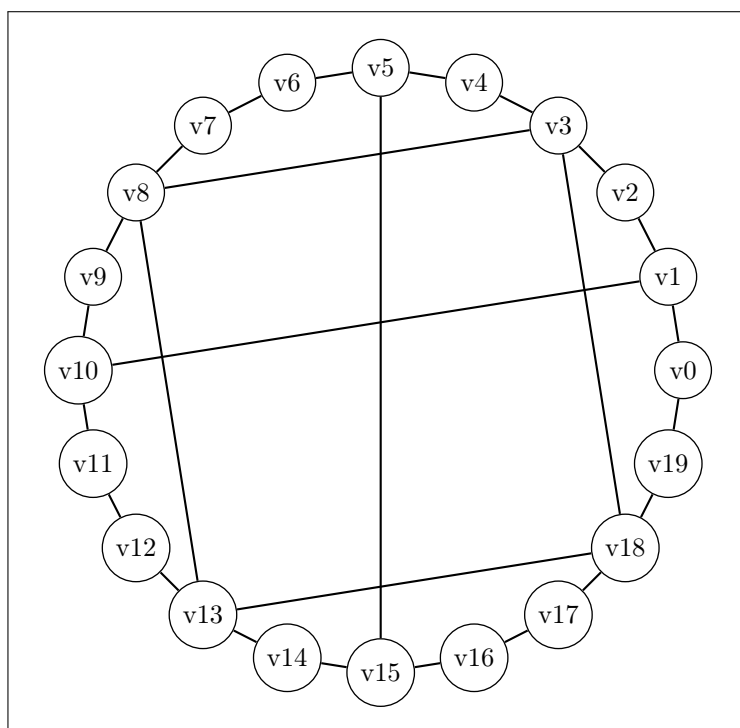
```
import networkx
import random
G = networkx.cycle_graph(20)
networkx.set_edge_attributes(G, 0.95, 'rel')
G.add_edge(1, 10, rel=0.8)
G.add_edge(5, 15, rel=0.7)
G.add_edge(3, 8, rel=0.4)
G.add_edge(8, 3, rel=0.4)
G.add_edge(13, 8, rel=0.4)
G.add_edge(18, 13, rel=0.4)
networkx.write_edgelist(G, "2_1_4_a.adjlist", data=True)
```

Początek wywołania jest identyczny jak dla zdefiniowanych. Po zatwierdzeniu opcji „wczytania danych” należy podać nazwę lub pełną ścieżkę pliku, zamiast numeru podpunktu.

```
Podaj ilość powtorzeń (1-1000000000): 1000000
Podaj czas odswiezania (0.0-5.0): 0
Wczytac dane z pliku? (T/N) T
Podaj nazwe pliku: 2_1_4_a.adjlist

Prosze czekac, trwa losowanie...
iteracje: 1000000, czas: 279.0s, niezawodnosc: 90.89%
```

Wynik 91% potwierdza, że nawet wysokoawaryjne połączenia dodatkowe mogą zauważalnie zwiększyć niezawodność.



Rysunek 5: Przykładowa realizacja grafu z krawędziami losowymi $|V| = 20, |E| = 26$.

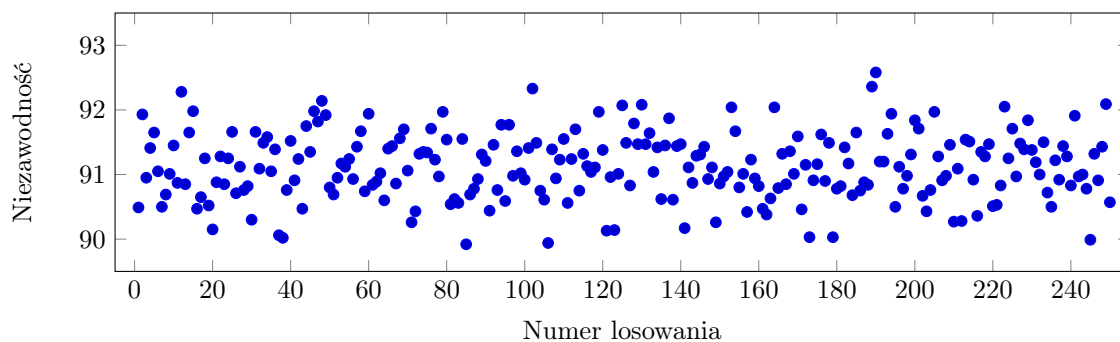
1.2.5 Graf cykliczny z sześcioma dodatkowymi krawędziami (losowanie wielokrotne)

Ograniczając rozważania do grafów prostych, podane wartości to jeden z około $4 \cdot 10^7$ wariantów. Każdy może mieć inną niezawodność sumaryczną. Z tego powodu dla podpunktu czwartego został zaimplementowany wariant ogólny. Program porównujący dokonuje ponownego losowania krawędzi przy każdej iteracji. Prowadzi to do wyznaczenia wartości uśrednionych.

```
Wczytać dane z pliku? (T/N) N
Podaj numer podpunktu (1-4): 4

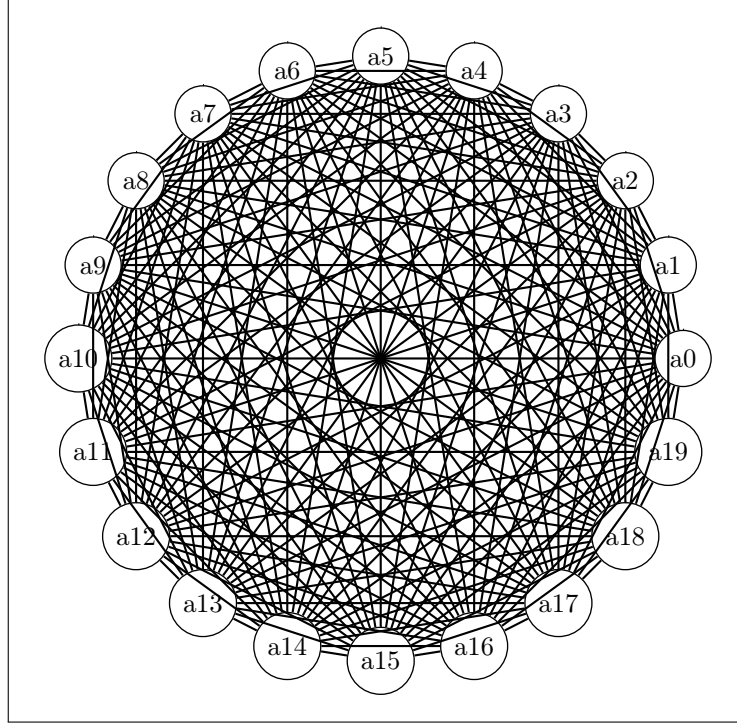
Proszę czekać, trwa losowanie...
iteracje: 1000000, czas: 286.3s, niezawodność: 90.95
```

Różnica rezultatów dla wcześniej rozważanej topologii oraz dla wielokrotnego losowania wynosi poniżej 0,1%. W celu oszacowania, jak znacząca jest ta różnica, wykonano testy 250 innych zestawów losowych.



Rysunek 6: Sumaryczna niezawodność różnych zestawów krawędzi losowych

Rysunek 6 przedstawia podsumowanie wielokrotnych losowań. Zakres od 90% do 93% pozwala uznać topologię różniącą się o 0.1% jako dobre przybliżenie wariantu teoretycznego.



Rysunek 7: Graf pełny, nałożone wszystkie możliwe realizacje krawędzi losowych $|V| = 20, |E| = 190$.

1.3 Niezawodność teoretyczna

Prawdopodobieństwo nierozspójnienia prostych topologii sieciowych można również wyznaczyć matematycznie.

1.3.1 Graf liniowy

Jeśli graf nie jest odporny na usunięcie usunięcie jakiegokolwiek krawędzi, niezawodność grafu to iloczyn poszczególnych niezawodności:

$$p^n = (0,95)^{19} \approx 0,37735360253$$

1.3.2 Graf cykliczny

Tolerancję pojedynczego uszkodzenia można podzielić na dwa osobne przypadki. Wariant braku uszkodzeń analogiczny do bezawaryjności liniowego. Awaria dokładnie jednej krawędzi wymaga rozważenia dla każdego wierzchołka prawdopodobieństwa jego uszkodzenia oraz braku uszkodzeń wszystkich pozostałych. Sumując:

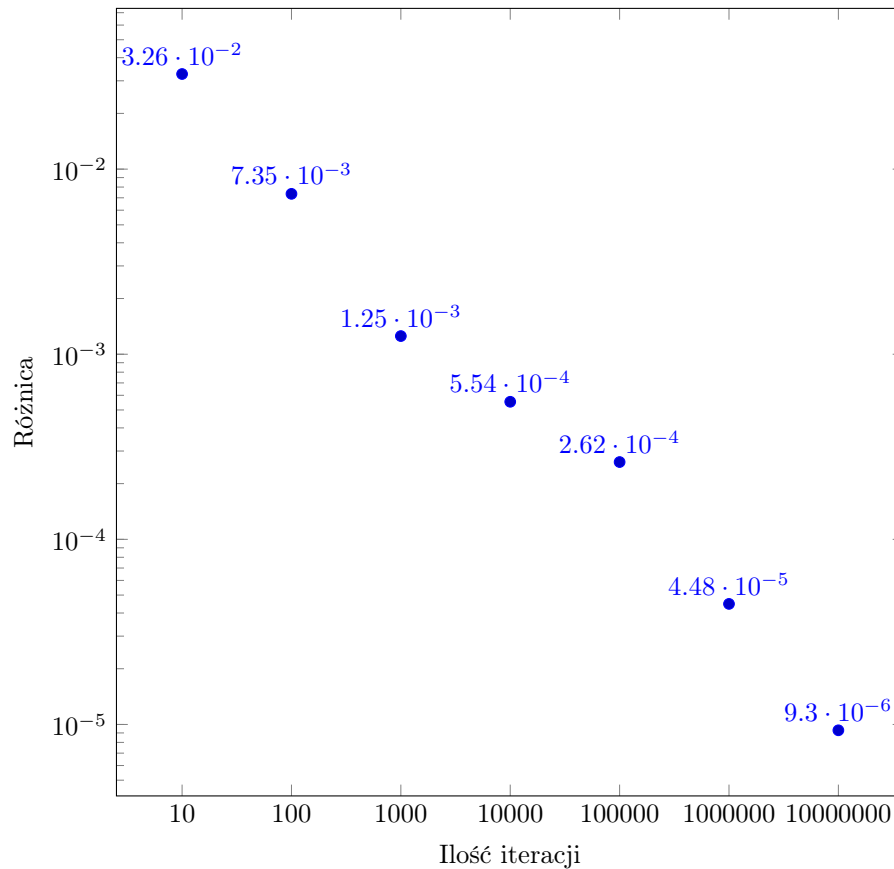
$$p^n + n * (1 - p) * p^{n-1} = 0,95^{20} + 20 * 0,05 * 0,95^{19} \approx 0,73583952494$$

1.3.3 Rozszerzenia grafu cyklicznego

Dodanie dodatkowych krawędzi o różnej niezawodności utrudnia zastosowanie analogicznych reguł. Przykładowo dla dwóch połączeń nadmiarowych, ozspójnienie może nastąpić po drugiej, trzeciej lub czwartej awarii. Trywialna metoda wymagałaby analizy $\sum_{k=2}^4 \binom{20}{k} = 6175$ wariantów.

1.4 Dokładność

Porównanie wyników teoretycznych z metodą Monte Carlo pozwala oszacować zależność dokładności od liczby iteracji. Jako przykład posłużył podpunkt pierwszy z grafem liniowym, o niezawodności teoretycznej 0,37735360253%. Dla $n \in [1, 7]$ wykonano po 10 wywołań 10^n iteracji. Uśrednione wartości błędu przedstawia wykres (Rysunek 8).



Rysunek 8: Zależność wielkości błędu od ilości powtórzeń

Dla uprzednio przyjętej wartości 1000000 iteracji średni błąd wynosił $4,48 \cdot 10^{-5}$, co można uznać za wystarczającą dokładność przy analizie opartej na trzech miejscach znaczących.

1.5 Wnioski

1. Zmiana grafu liniowego na cykliczny pozwala znacząco zwiększyć niezawodność poprzez dodanie jednej krawędzi. W praktycznych zastosowaniach należy jednak uwzględnić ograniczenia technologiczne i geograficzne.
2. Przy teoretycznym modelowaniu nadmiarowe połączenia mogą zauważalnie poprawić bezwaryjność pomimo wysokiej wadliwości jednostkowej.
3. Metoda Monte Carlo pozwala szybko szacować parametry skomplikowanych topologii sieciowych, natomiast dokładność jest ograniczona przez ilość iteracji oraz jakość generatora liczb losowych.

2 Szacowanie opóźnień

2.1 Metodyka

Program przyjmuje pięć parametrów wejściowych: ilość iteracji, niezawodność, nazwę pliku z macierzą, nazwę pliku z krawędziami grafu oraz opcjonalnie T_{max} . Jeśli T_{max} nie zostanie podane, jest wyznaczane jako dwukrotność T w stanie początkowym. Z uwagi na ilość danych generowanych w każdej iteracji, dostępny jest wyłącznie tryb tekstowy.

2.2 Algorytm wariantu początkowego

Średnie opóźnienie wariantu początkowego jest liczone bezpośrednio ze wzoru:

$$T = \frac{1}{G} * \sum_e \frac{a(e)}{c(e) - a(e)}$$

- G : suma elementów macierzy natężeń
- $a(e)$: przepływ dla danej krawędzi
- $c(e)$: przepustowość danej krawędzi, określona jako trzykrotność początkowego maksymalnego przepływu

2.3 Algorytm dla dowolnego przedziału czasowego

1. Wyznaczenie przepływu, przepustowości i średniego opóźnienia w grafie nieuszkodzonym (jak wyżej).
2. Określenie niezawodności, w tym przypadku 95% dla każdego połączenia.
3. Usunięcie krawędzi, dla których wylosowana wartość $[0,1]$ przekroczyła niezawodność.
4. Kontrola spójności grafu.
5. Obliczenie aktualnych przepływów.
6. Wyznaczenie ze wzoru średniego opóźnienia T i porównanie z T_{max} .
7. Sprawdzenie, czy dla każdej krawędzi przepływ jest mniejszy od przepustowości.
8. Jeśli graf pozostał spójny, średnie opóźnienie nie przekracza maksymalnego, przepływ nie przekroczył przepustowości - powtórzenie kroków 3-7.

Po każdej iteracji aktualizowane są statystyki przyczyn zakończenia: rozspójnienie, przeciążenie, przekroczenie T_{max} . Sumaryczny raport zostaje wyświetlony po zakończeniu pracy. W pliku `result.txt` dodatkowo zapisywane są poszczególne wartości opóźnień.

2.3.1 Przykład uruchomienia

```
Ilosc powtorzen (1-1000000000): 100
Niezawodnosc (0-99): 98
Nazwa pliku z macierza natezen: mrand.csv
Nazwa pliku z lista krawedzi: gpetersen.adjlist
Maksymalne srednie opoznienie (T_max): 0

Rozspojnione: 17
Przeciazone: 13
Przekroczone TMax: 70

Usredniony czas prawidlowego funkcjonowania: 14.41
```

2.4 Wyniki

Pierwsza macierz (**muni**) symuluje identyczny przesył między każdymi dwoma wierzchołkami, druga (**mrnd**) zawiera wartości losowe. Tylko wyniki uzyskane dla tej samej macierzy mogą być bezpośrednio porównywane. T_{max} określone domyślnie jako dwukrotność początkowego średniego opóźnienia.

$$M1 = \begin{bmatrix} 0 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 0 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 0 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 0 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 0 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 0 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 0 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 0 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 0 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 0 \end{bmatrix} \quad M2 = \begin{bmatrix} 0 & 4 & 5 & 5 & 1 & 6 & 2 & 8 & 9 & 0 \\ 2 & 0 & 2 & 2 & 5 & 3 & 9 & 6 & 5 & 5 \\ 0 & 5 & 0 & 0 & 7 & 7 & 6 & 8 & 4 & 3 \\ 3 & 8 & 7 & 0 & 0 & 6 & 7 & 6 & 5 & 5 \\ 5 & 6 & 1 & 1 & 0 & 5 & 6 & 1 & 3 & 6 \\ 0 & 0 & 5 & 8 & 3 & 0 & 8 & 8 & 4 & 2 \\ 6 & 7 & 9 & 3 & 7 & 1 & 0 & 5 & 3 & 1 \\ 9 & 9 & 1 & 5 & 7 & 6 & 7 & 0 & 5 & 6 \\ 1 & 3 & 5 & 8 & 5 & 6 & 2 & 2 & 0 & 9 \\ 2 & 9 & 7 & 4 & 6 & 0 & 5 & 9 & 3 & 0 \end{bmatrix}$$

Wszystkie testy zostały wykonane dla 10000 iteracji. Zmniejszona liczba względem poprzedniego zadania jest konsekwencją dłuższego czasu trwania pojedynczego przejścia. Teoretyczne wyznaczenie dokładności znacząco przekraczałoby zakres niniejszej pracy, z uwagi na poziom skomplikowania algorytmu. Natomiast analogicznie do wcześniejszych rozważań na temat metody Monte Carlo, można określić precyzję jako wystarczającą do porównań poszczególnych grafów, pod warunkiem wystąpienia różnic co najmniej na czwartej cyfrze znaczącej.

2.4.1 Graf cykliczny

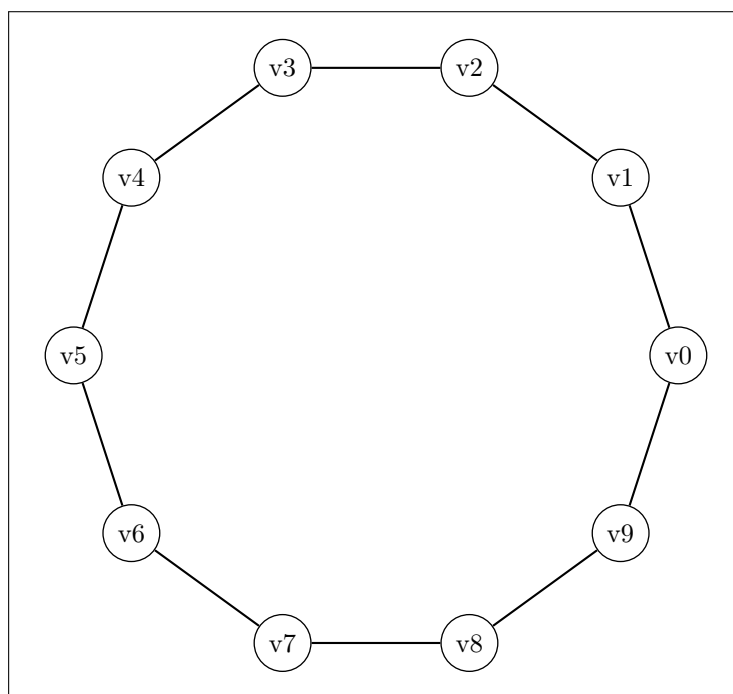
Topologia o niskim stopniu odporności na uszkodzenia.

```
Nazwa pliku z macierza natezen: muni.csv  
  
Rozspojnione: 10000  
Przeciazone: 0  
Przekroczone TMax: 0  
  
Usredniony czas prawidlowego funkcjonowania: 3.6341
```

Wynik dla macierzy identycznych elementów jest zgodny z przewidywaniami teoretycznymi. Po wystąpieniu pojedynczej awarii, pakiety są przesyłane jak w liniowym. Druga awaria prowadzi do rozspójnienia.

```
Nazwa pliku z macierza natezen: mrnd.csv  
  
Rozspojnione: 1363  
Przeciazone: 1704  
Przekroczone TMax: 6933  
  
Usredniony czas prawidlowego funkcjonowania: 4.9516
```

W przypadku macierzy losowej nadmiarowy ruch z uszkodzonej krawędzi nie rozkłada się równomiernie, co prowadzi do nadmiernego wzrostu średnich opóźnień. W kolejnych przykładach macierz losowa zostanie pominięta, ze względu na brak równie istotnych rezultatów.



Rysunek 9: Graf cykliczny $|V| = 10, |E| = 10$.

2.4.2 Graf Petersena

Silnie regularny, trójspójny graf stopnia 3 znacząco różni się właściwościami od cyklu.

Nazwa pliku z macierza natezen: muni.csv

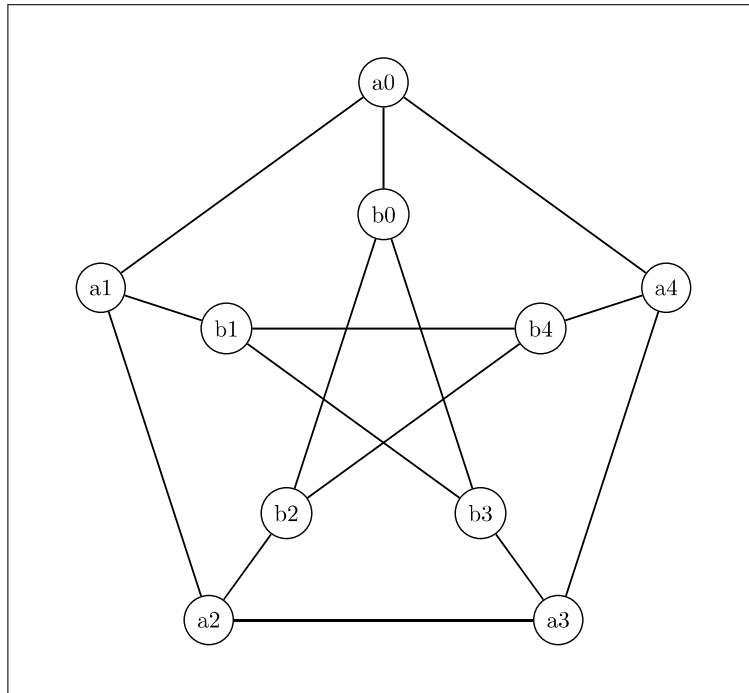
Rozspojnione: 616

Przeciazone: 3536

Przekroczone TMax: 5848

Usredniony czas prawidłowego funkcjonowania: 3.8798

Niższy odsetek iteracji zakończonych z powodu powstania wierzchołków izolowanych wynika z trójspójności grafu. Czas prawidłowego funkcjonowania wzrósł o około 7% względem grafu cyklicznego.



Rysunek 10: Graf Petersena $|V| = 10, |E| = 15$.

2.4.3 Graf kołowy

Wierzchołek w centrum jest połączony ze wszystkimi pozostałymi, co pozwala efektywnie przekazywać ruch pomimo pojedynczych uszkodzonych krawędzi.

Nazwa pliku z macierza natezen: muni.csv

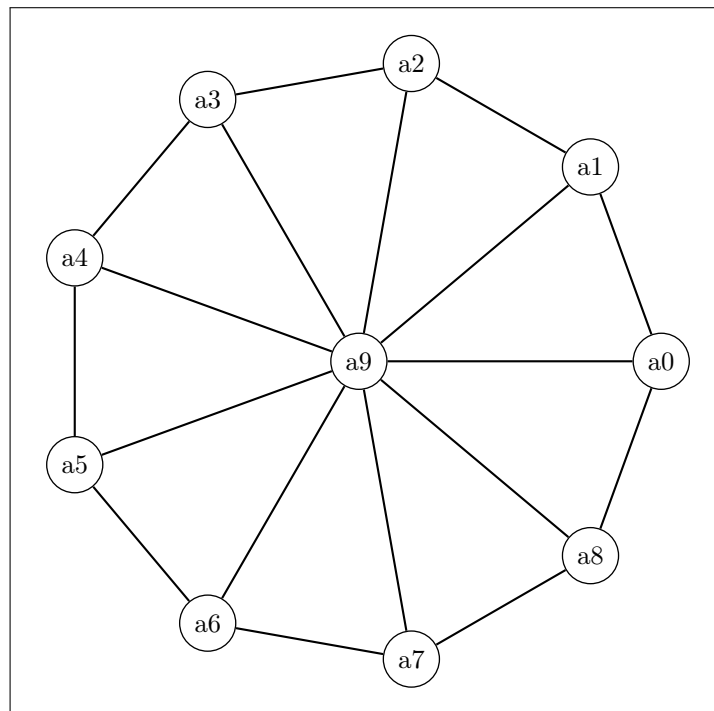
Rozspojnione: 3126

Przeciążone: 1739

Przekroczone TMax: 5135

Usredniony czas prawidłowego funkcjonowania: 6.6546

Test potwierdza własności teoretyczne. Czas poprawnego działania ponad 80% większy, niż dla grafu cyklicznego.



Rysunek 11: Graf kołowy $|V| = 10, |E| = 10$.

2.5 Wnioski

1. Prawdopodobieństwo rozspójnienia nie jest jedynym czynnikiem wpływającym na wybór właściwego układu.
2. Połączenia posiadają określoną maksymalną przepustowość, co uniemożliwia przejęcie dowolnie dużego ruchu z uszkodzonych ścieżek.
3. Graf kołowy charakteryzuje się pożądanymi parametrami, jednak konieczność bezpośredniego połączenia jednego wierzchołka ze wszystkimi pozostałymi stanowiłaby istotne utrudnienie przy realizacji praktycznej.