0.1 Narzędzia wykorzystane do realizacji zadania

Do realizacji zadania pierwszego użyłem biblioteki Javy: JGraphT, która służy do tworzenia i manipulowania struktury grafów.
Do szacowania niezawodności (spójności) posłużyłem się metodą Monte Carlo.

0.2 Wyniki na zad 1

Prawdopodobienstwo spojnosci w a: 0.89343
Prawdopodobienstwo spojnosci w b: 0.89377
Prawdopodobienstwo spojnosci w c: 0.96863
Prawdopodobienstwo spojnosci w d: 0.96832

0.3 Wyniki na zad2

Mam nastepujaca topologie, od v1 do v10 wierzcholki (v(i), v(i + 1)), ponadto polaczylem v4 z v7 i v8 oraz v1 z v5 i v8, kazdy wierzcholek ma prawdopodobienstwo 0.95 i tablica natezen okresla przesyl 10 z kazdego wierzcholka do dowolnego innego, Wyliczone funkcje
Edgevertex1='v3', vertex2='v4', reliability=0.95, a=180.0, c=720.0
Edgevertex1='v4', vertex2='v5', reliability=0.95, a=170.0, c=680.0
Edgevertex1='v5', vertex2='v6', reliability=0.95, a=110.0, c=440.0
Edgevertex1='v6', vertex2='v7', reliability=0.95, a=90.0, c=360.0
Edgevertex1='v7', vertex2='v8', reliability=0.95, a=140.0, c=560.0
Edgevertex1='v8', vertex2='v9', reliability=0.95, a=320.0, c=1280.0
Edgevertex1='v9', vertex2='v10', reliability=0.95, a=180.0, c=720.0
Edgevertex1='v4', vertex2='v7', reliability=0.95, a=70.0, c=280.0
Edgevertex1='v4', vertex2='v8', reliability=0.95, a=180.0, c=720.0
Edgevertex1='v1', vertex2='v5', reliability=0.95, a=100.0, c=400.0
Edgevertex1='v1', vertex2='v8', reliability=0.95, a=140.0, c=560.0

Przyjalem srednia wielkosc pakietu w bitach 'm' = 1, dla Tmax = 0.005
Sredni czas opoznien mi wychodzil: 0.00324675090868328
A szacowane prawdopodobienstwo w ostatnim podpunkcie: 0.876

Dla Tmax = 0.0035 juz wyniki symulacji ukazuja: 0.576

0.3.1 Wnioski

Możemy zaobserwować, że dla większych p grafy są bardziej niezawodne, tzn. prawdopodobieństwo rozspójnienia sieci jest mniejsze. Zwiększając Tmax przy ustalonym p również otrzymujemy lepszą niezawodność.