Problem niezawodności sieci

Kamil Kasperek

18.01.2023 r.

Zadanie polega na zaproponowaniu struktury sieci komputerowej dla zadanej liczby węzłów (komputerów) aby koszt jej budowy był minimalny ale gwarantujący założony poziom niezawodności. Niezawodność sieci należy rozumieć jako maksymalną liczbę awarii, których wystąpienie nie spowoduje awarii całej sieci (nadal będzie istniała możliwość przesłania informacji pomiędzy dowolnymi węzłami).

Program uruchamiany jest z linii poleceń z wykorzystaniem następujących przełączników:

-i plik wejściowy

-o plik wyjściowy

-n poziom niezawodności (wartość całkowita)

Przykładowy plik z danymi:

Komputer1 Komputer2 220.0

Komputer1 Komputer3 100.0

Komputer1 Komputer4 150.0

Komputer2 Komputer3 175.0

Komputer2 Komputer4 125.0

Komputer3 Komputer4 160.0

Plik wynikowy dla n = 1:

Komputer1 Komputer3

Komputer1 Komputer4

Komputer2 Komputer3

Komputer2 Komputer4

Wczytane dane zapisywane są do dwóch struktur. Pierwsza (Data) to wektor, gdzie dane zapisywane są w analogiczny sposób do pliku wejściowego.

Druga (Graph) to mapa w której klucze to kolejne wierzchołki, a wartościami są zestawy wierzchołków sąsiadujących z kluczem.

Algorytm wykonujący zadanie bazuje na odwrotnym algorytmie Kruskala.

Wczytane dane są sortowane malejąco względem wag krawędzi.

Aby zapewnić najtańszą możliwą budowę sieci algorytm wykonuje się dla każdego wierzchołka.

W pierwszym kroku usuwane są najdroższe krawędzie wychodzące z wierzchołka startowego, dopóki będzie ich minimalna ilość.

Następnie usuwane są najdroższe krawędzie wedle wcześniej posortowanych danych, jednocześnie dbając o zapewnienie minimalnej ilości krawędzi wychodzących z wierzchołków. Kosz budowy tak stworzonego grafu porównywany jest z kosztem budowy obecnie najkorzystniejszego, jeśli jest tańszy, staje się nowym wynikowym grafem. Po przejściu wszystkich wierzchołków zwracany jest najtańszy graf spełniający poziom niezawodności.

Szczególnym przypadkiem jest sytuacja, gdy poziom niezawodności wynosi 0.

Należy wtedy zadbać o spójność grafu, dlatego w tym wypadku po każdym usunięciu krawędzi sprawdzana jest spójność grafu, jeżeli usunięcie krawędzi spowoduje niespójność usunięta krawędź jest przywracana. Nie jest też konieczne wykonywanie algorytmu dla każdego wierzchołka z osobna, ponieważ od razu budowana jest możliwa najtańsza sieć.

Działanie programu krok po kroku dla danych wejściowych:

Komputer1 Komputer2 400.0

Komputer1 Komputer3 300.0

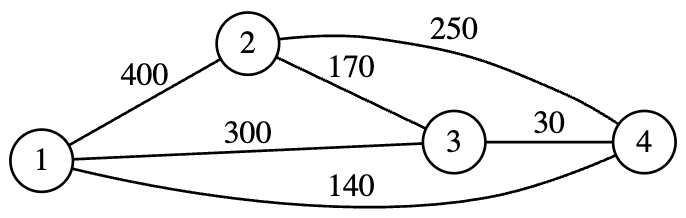
Komputer1 Komputer4 140.0

Komputer2 Komputer3 170.0

Komputer2 Komputer4 250.0

Komputer3 Komputer4 30.0

n = 1 (poziom niezawodności)

Graficzne przedstawienie danych w postaci grafu:

1. Wczytanie danych do struktur Data i Graph
2. Posortowanie danych w strukturze Data wg wag krawędzi malejąco:

Komputer1 Komputer2 400.0

Komputer1 Komputer3 300.0

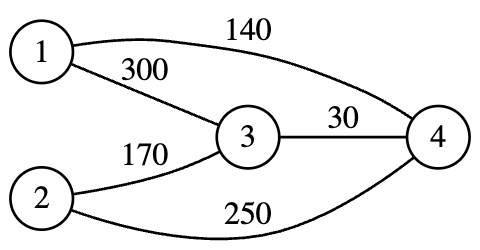
Komputer2 Komputer4 250.0

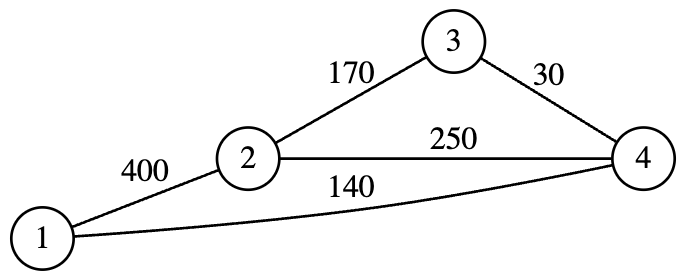
Komputer2 Komputer3 170.0

Komputer1 Komputer4 140.0

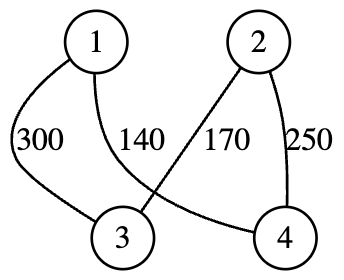
Komputer3 Komputer4 30.0

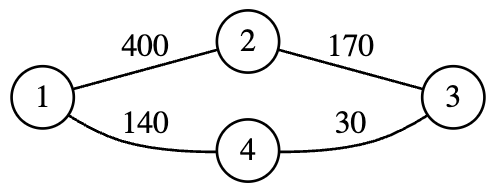
3. Usunięcie najdroższych krawędzi w wybranym wierzchołku startowym z zachowaniem minimalnej ich ilości zapewniającej żądaną niezawodność.

Dla wierzchołka 1:

Dla wierzchołka 3:

4. Iteracja po danych i usunięcie krawędzi, jeżeli nie spowoduje ono nie zapewnienia żądanej niezawodności.

Dla wierzchołka 1: (koszt budowy 860)

Dla wierzchołka 3: (koszt budowy 740)

5. Porównanie kosztu budowy analizowanego grafu z obecnie najtańszym, jeżeli analizowany tańszy ustawienie go jako najtańszy.

6. Wybranie nowego wierzchołka startowego i skok do punktu 3., jeżeli brak nowych wierzchołków do analizy skok do punktu 7.

7. Zapisanie struktury najtańszej w budowie sieci do pliku wyjściowego w formacie:

Komputer1 Komputer2

Komputer1 Komputer4

Komputer2 Komputer3

Komputer3 Komputer4