POLITECHNIKA
GDAŃSKA

# Analiza algorytmów dla dominowania rzymskiego słabo spójnego

## Analysis of algorithms for weakly connected Roman domination

inż. Paulina Brzęcka

30 czerwca 2025

# Definicja problemu

## Definicja

Funkcję dominującą rzymską słabo spójną (WCRDF) na grafie $G$ definiuje się jako taką funkcję dominującą rzymską $f : V(G) \to \{0, 1, 2\}$, dla której zbiór wierzchołków

$$\{u \in V(G) : f(u) \in \{1, 2\}\}$$

stanowi jednocześnie słabo spójny zbiór dominujący.

Wagę funkcji $f$ definiuje się jako:

$$f(V) = \sum_{u \in V} f(u)$$

Liczbą dominowania rzymskiego słabo spójnego grafu $G$ nazywamy najmniejszą możliwą wagę funkcji $f$ spełniającej powyższe warunki i oznaczamy ją symbolem:

$$\gamma_R^{\mathrm{wc}}(G)$$

# Cel pracy

- analiza algorytmów dla dominowania rzymskiego słabo spójnego,
- opisanie już istniejących rozwiązań i opracowanie własnych,
- analiza i porównanie ich skuteczności,
- znalezienie możliwych praktycznych zastosowań.

# Pytania badawcze

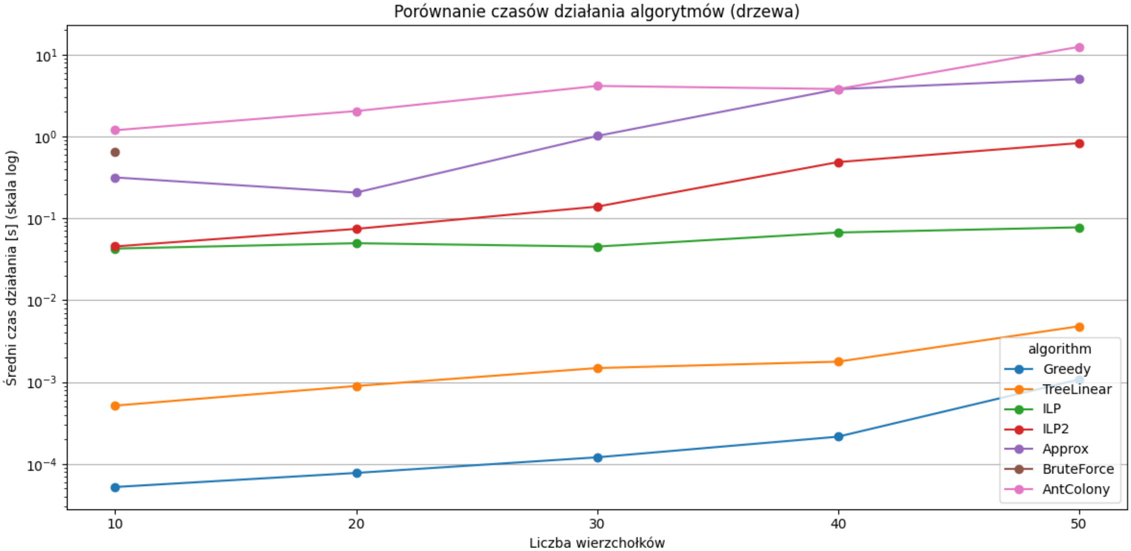
- Jakie algorytmy są w stanie znaleźć WCRDF? Które z nich są w stanie znaleźć dodatkowo najmniejszą sumę wag WCRDF?
- Czy czas i jakość działania algorytmów będzie uzależniony od klasy grafów?
- Czy i jakie algorytmy heurystyczne mogą skutecznie przybliżyć wartość liczby dominowania rzymskiego słabo spójnego w czasie krótszym niż dokładne algorytmy?
- Czy hiperparametry algorytmu mrówkowego można dostroić w taki sposób, aby ten algorytm znajdował liczbę dominowania rzymskiego słabo spójnego bliską optymalnej?

# Algorytmy

| Algorytm | Złożoność czasowa |
|---|---|
| brute familyorce | $O(3^n \cdot n^2)$ |
| liniowy dla drzew | $O(n)$ |
| programowania liniowego 1 | wykładnicza |
| programowania liniowego 2 | wykładnicza |
| mrówkowy | $O(num\_iterations \cdot num\_ants \cdot n^2)$ |
| aproksymacyjny | wykładnicza |
| zachłanny | $O(n^2)$ - grafy rzadkie, $O(n^3)$ - grafy gęste |

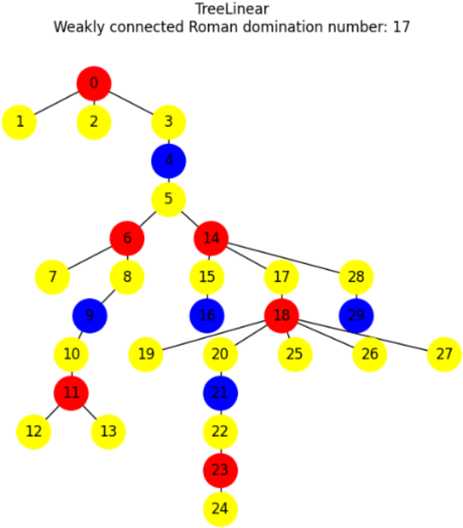
Tabela: Porównanie złożoności czasowej różnych klas algorytmów

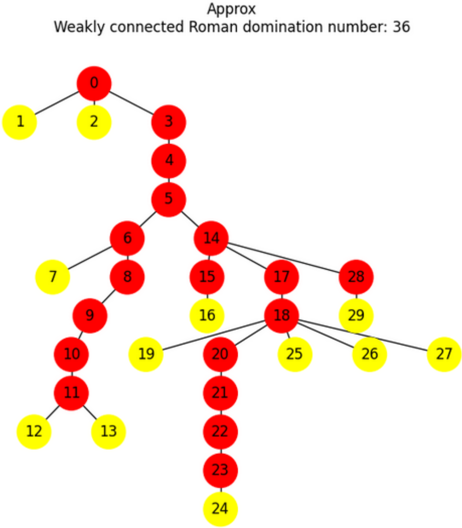
# Wyniki



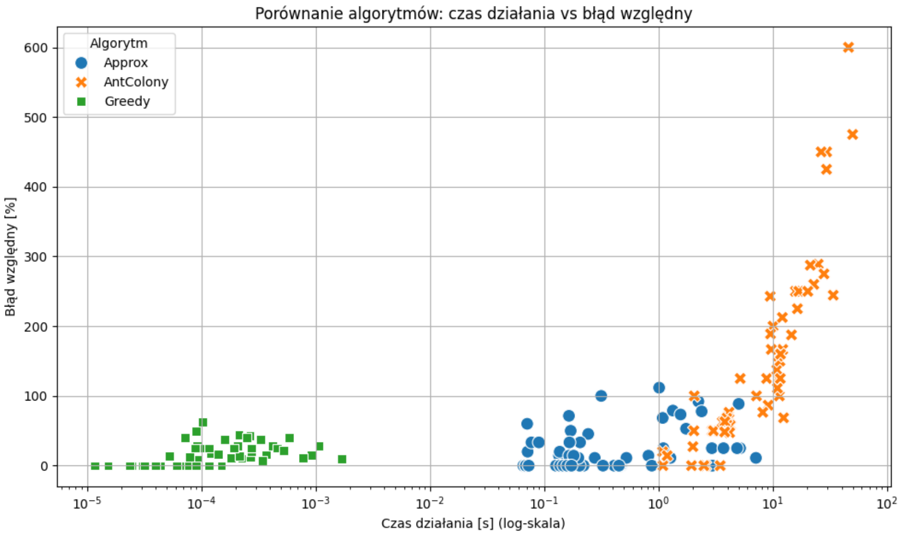
Porównanie czasów działania algorytmów (drzewa)
Średni czas działania [s] (skala log)
$10^1$
$10^0$
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
10
20
30
40
50
Liczba wierzchołków
algorithm
Greedy
TreeLinear
ILP
ILP2
Approx
BruteForce
AntColony

# Wyniki


TreeLinear
Weakly connected Roman domination number: 17
0
1
2
3
4
5
6
14
7
8
15
17
28
9
16
18
29
10
19
20
25
26
27
11
21
12
13
22
23
24


Approx
Weakly connected Roman domination number: 36
0
1
2
3
4
5
6
14
7
8
15
17
28
9
16
18
29
10
19
20
25
26
27
11
21
12
13
22
23
24

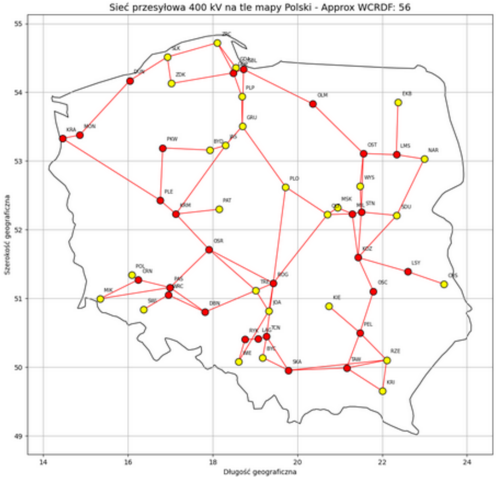
# Wyniki

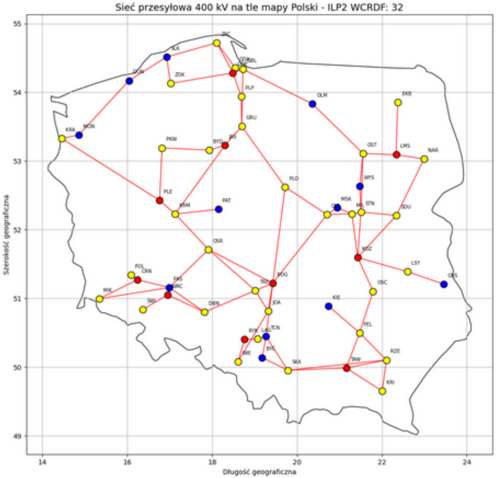


Porównanie algorytmów: czas działania vs błąd względny
Algorytm
Approx
AntColony
Greedy
Błąd względny [%]
600
500
400
300
200
100
0
$10^{-5}$
$10^{-4}$
$10^{-3}$
$10^{-2}$
$10^{-1}$
$10^{0}$
$10^{1}$
$10^{2}$
Czas działania [s] (log-skala)

POLITECHNIKA GDAŃSKA

# Zastosowania praktyczne




Rysunek: Rozmieszczenie zabezpieczeń sieci energetycznych.

# Wnioski

- udało się odpowiedzieć na pytania badawcze,
- konieczność doboru właściwego algorytmu do grafu lub klasy grafu,
- słaba jakość rozwiązań algorytmu mrówkowego,
- pokazanie potencjalnych rozwiązań praktycznych w ujęciu teoretycznym.

# Dalsze kierunki badań

- implementację i testy potencjalnych ulepszeń dla algorytmu zachłannego,
- rozszerzenie testów na inne klasy, jak i na inne wielkości grafów,
- podniesienie jakości wyników algorytmu mrówkowego, między innymi poprzez inną implementację heurystyki lokalnej oraz strategii feromonowej
- opracowanie algorytmów dokładnych, rozwiązywalnych w czasie wielomianowym dla innych klas grafów
- weryfikację i przełożenie teoretycznych rozważań na temat praktycznych zastosowań tego problemu na praktyczną analizę i realizację

POLITECHNIKA
GDAŃSKA