

POLITECHNIKA WROCŁAWSKA

PROJEKT

ZARZĄDZANIE W SYSTEMACH I SIECIACH KOMPUTEROWYCH

---

**Analiza wpływu liczby procesów na czas  
działania algorytmów dla problemu  
komiwojażera**

---

*Authors:*

Rafał PIENIAŻEK  
Jakub POMYKAŁA

*Supervisor:*

Dr inż. Robert WÓJCIK

7 kwietnia 2016

# Spis treści

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Wstęp</b>  | <b>2</b> |
| 1.1      | Cel projektu . . . . .  | 2        |
| 1.2      | Zakres projektu . . . . .                                       | 2        |
| <b>2</b> | <b>Sformułowanie problemu</b>                                   | <b>2</b> |
| 2.1      | Opis wariantów problemu . . . . .                               | 2        |
| 2.1.1    | Przegląd zupełny . . . . .                                      | 2        |
| 2.1.2    | Branch&Bound . . . . .  | 2        |
| <b>3</b> | <b>Projekt aplikacji</b>  | <b>3</b> |
| 3.1      | Wybrane klasy . . . . .   | 3        |
| 3.1.1    | Matrix . . . . .  | 3        |
| 3.1.2    | Edge . . . . .  | 3        |
| 3.1.3    | Node . . . . .  | 3        |
| 3.2      | Realizacja algorytmów wyznaczania rozwiązań . . . . .           | 3        |
| 3.2.1    | Przegląd zupełny . . . . .                                      | 3        |
| 3.2.2    | Branch&Bound . . . . .  | 3        |
| 3.3      | Odczyt i zapis danych . . . . .                                 | 3        |
| 3.3.1    | Wczytywanie z plików .tsp . . . . .                             | 3        |
| 3.3.2    | Wczytywanie macierzy . . . . .                                  | 3        |
| 3.3.3    | Generowanie danych losowych . . . . .                           | 3        |
| 3.3.4    | Generowanie plików cvs . . . . .                                | 3        |
| <b>4</b> | <b>Testowanie wydajności</b>                                    | <b>3</b> |
| 4.1      | Czasy wykonania algorytmów . . . . .                            | 3        |
| 4.1.1    | Czas działania algorytmu metodą podziału i ograniczeń . . . . . | 3        |
| 4.1.2    | Czas działania algorytmu dla przeglądu zupełnego . . . . .      | 3        |
| 4.1.3    | Porównanie czasów wykonywania algorytmu . . . . .               | 4        |
| 4.2      | Analiza i ocena jakości . . . . .                               | 4        |
| <b>5</b> | <b>Podsumowanie</b>   | <b>4</b> |

# 1 Wstęp

## 1.1 Cel projektu

## 1.2 Zakres projektu

# 2 Sformułowanie problemu

Problem komiwojażera jest problemem optymalizacyjnym, polegającym na znalezieniu ścieżki pomiędzy ustalonymi miastami dla następujących warunków:

- wszystkie miasta są odwiedzone dokładnie jeden raz
- rozpoczynamy i kończymy w tym samym mieście.
- koszt (suma wag krawędzi) jest najmniejszy z wszystkich możliwych

Oznacza to, że należy znaleźć taki cykl Hamiltona dla grafu reprezentującego zbiór miast, dla którego suma wag wybranych krawędzi jest najmniejsza.

## 2.1 Opis wariantów problemu

### 2.1.1 Przegląd zupełny

### 2.1.2 Branch&Bound

W rozwiązaniu powyższego problemu zastosowano metodę podziału i ograniczeń. Metoda ta jest metodą optymalizacji dyskretnej, opierając się na podejściu *dziel i zwyciężaj*. W każdym kroku algorytmu przeglądane jest drzewo potencjalnych rozwiązań. Jeżeli natrafimy na węzeł, który jest liściem, czyli można określić dla niego długość drogi komiwojażera, sprawdzamy, czy nowo znaleziona wartość nie jest lepsza od aktualnie zapisanej. Jeżeli tak jest, to zapamiętujemy nowe rozwiązanie. W przypadku, gdy dany węzeł nie jest liściem, tworzymy dla niego podproblemy. W tym celu odwiedzamy kolejne miasto starając się oszacować dolne ograniczenie kosztów całej trasy. W tym projekcie wykorzystano najprostszy sposób szacowania. Na początku z macierzy sąsiedztwa wycinana jest przekątna, następnie kolumny i wiersze miejsc już odwiedzonych. Następnie wartości z tak przygotowanej macierzy są sortowane w kolejności niemalejącej. Szacowanie polega na dodaniu do siebie tylu kolejnych wartości z listy, ile zostało miast do odwiedzenia. W niniejszym projekcie do przetrzymywania drzewa rozwiązań wykorzystano listę jednokierunkową. Wybór wynika z faktu, iż i tak należy przejrzeć wszystkie możliwe węzły, aby sprawdzić, czy ich ograniczenie nie jest większe niż aktualnie znalezione.

## 3 Projekt aplikacji

### 3.1 Wybrane klasy

#### 3.1.1 Matrix

#### 3.1.2 Edge

#### 3.1.3 Node

### 3.2 Realizacja algorytmów wyznaczania rozwiązań

#### 3.2.1 Przegląd zupełny

#### 3.2.2 Branch&Bound

### 3.3 Odczyt i zapis danych

#### 3.3.1 Wczytywanie z plików .tsp

#### 3.3.2 Wczytywanie macierzy

#### 3.3.3 Generowanie danych losowych

#### 3.3.4 Generowanie plików cvs

## 4 Testowanie wydajności

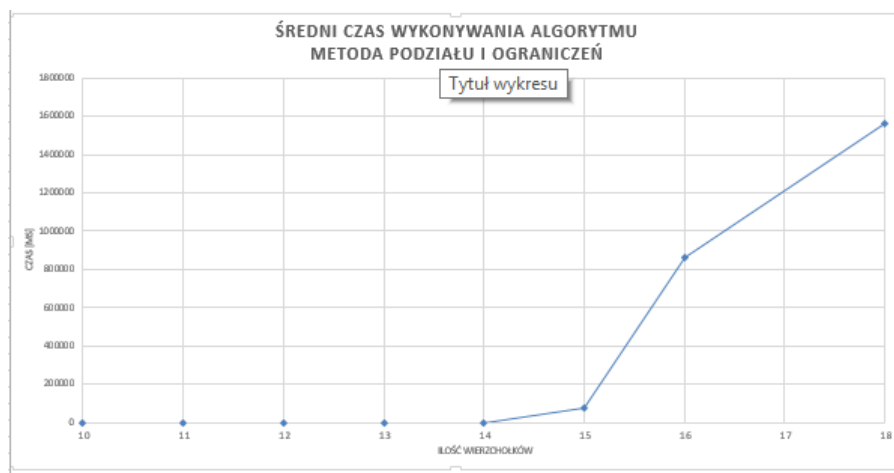
### 4.1 Czasy wykonania algorytmów

#### 4.1.1 Czas działania algorytmu metodą podziału i ograniczeń

Tabela 1: Uśrednione wyniki dla 18 różnych ilości wierzchołków

| Ilość miast | czas działania algorytmu [ms] |
|-------------|-------------------------------|
| 10          | 11                            |
| 11          | 6                             |
| 12          | 36                            |
| 13          | 100                           |
| 14          | 1535                          |
| 15          | 74434                         |
| 16          | 864463                        |
| 17          | 36992                         |
| 18          | 1560456                       |

Poniższy wykres przedstawia czas działania algorytmu dla metody podziału i ograniczeń. Wykres ma charakter wykładniczy. Znaczące różnice czasu można zaobserwować już pomiędzy 15 a 16 wierzchołkiem.

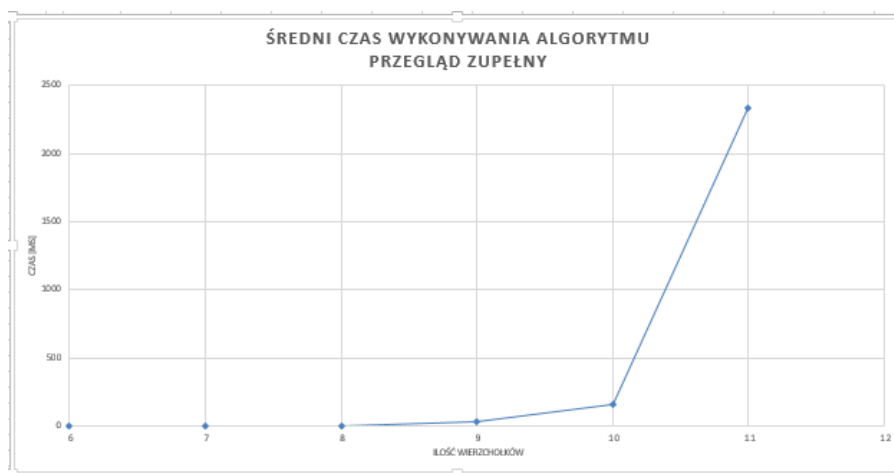


### 4.1.2 Czas działania algorytmu dla przeglądu zupełnego

Tabela 2: Uśrednione wyniki dla 6 różnych ilości wierzchołków

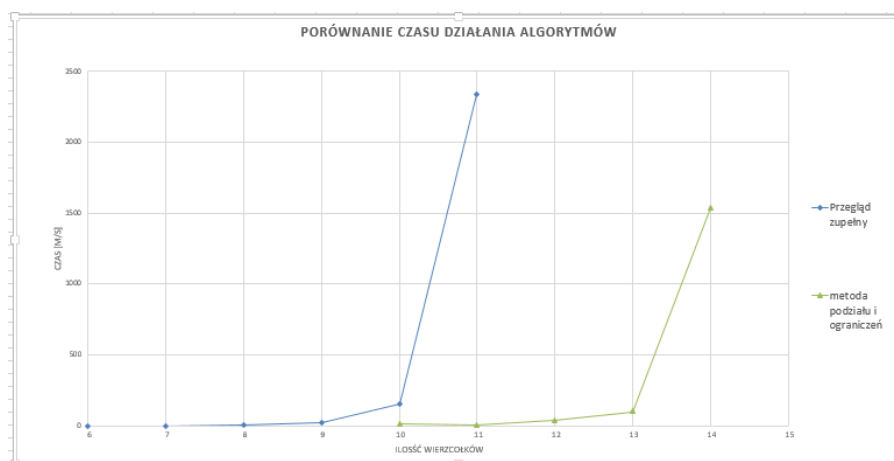
| Ilość miast | czas działania algorytmu [ms] |
|-------------|-------------------------------|
| 6           | 0,08                          |
| 7           | 0,5                           |
| 8           | 3,2                           |
| 9           | 25,2                          |
| 10          | 252,5                         |
| 11          | 2340                          |

Czas działania algorytmu dla przeglądu zupełnego ma charakter wykładniczy. Jest to metoda z najgorszą możliwą złożonością, jednakże z pewnością daje dobre wyniki.



### 4.1.3 Porównanie czasów wykonywania algorytmu

Wykres przedstawiony poniżej pozwala zaobserwować różnice pomiędzy czasem wykonywania algorytmu zarówno dla metody podziału i ograniczeń jak i przeglądu zupełnego. Nie jest zaskoczeniem, że czas sprawdzenia wszystkich możliwości jest dużo większy niż algorytm wykluczający pewne rozwiązania ze zbioru możliwych permutacji.



## 4.2 Analiza i ocena jakości

## 5 Podsumowanie