

# Projektowanie Efektywnych Algorytmów

Implementacja i analiza efektywności metody Tabu Search dla problemu komiwojażera

*Autorzy:*

Sebastian Łągiewski 226173

Łukasz Zatorski 226172

Prowadzący : mgr inż. Radosław Idzikowski

Wydział Elektroniki  
III rok

15 grudnia 2017

# Spis treści

|          |                                                                                                                   |          |
|----------|-------------------------------------------------------------------------------------------------------------------|----------|
| <b>1</b> | <b>Opis algorytmu</b>                                                                                             | <b>2</b> |
| <b>2</b> | <b>Plan eksperymentu</b>                                                                                          | <b>3</b> |
| <b>3</b> | <b>Wyniki</b>                                                                                                     | <b>4</b> |
| 3.1      | Wpływ strategii dywersyfikacji na działanie algorytmu . . . . .                                                   | 4        |
| 3.2      | Zależność jakości rozwiązania od czasu(ilości iteracji) znajdowania się wybranych ruchów na liście tabu . . . . . | 5        |
| 3.3      | Zależność jakości rozwiązania od długości wykonywania się algorytmu . . . . .                                     | 6        |
| 3.3.1    | rat783 . . . . .                                                                                                  | 6        |
| 3.3.2    | pr1002 . . . . .                                                                                                  | 7        |
| 3.3.3    | d2103 . . . . .                                                                                                   | 8        |
| <b>4</b> | <b>Podsumowanie</b>                                                                                               | <b>9</b> |

# 1 Opis algorytmu

Tabu Search jest metaheurystyką do rozwiązywania problemów optymalizacyjnych, opartą na iteracyjnym przeszukiwaniu przestrzeni rozwiązań, wykorzystującą ich sąsiedztwo, zapamiętującą niektóre ruchy oraz odległość czasową ich wystąpienia, w celu unikania minimów lokalnych i poszukiwania rozwiązań globalnie optymalnych w rozsądnym czasie.

Algorytm Tabu Search dla problemu komiwojażera polega na iteracyjnym przeszukiwaniu sąsiedztwa obecnie rozpatrywanego rozwiązania. Początkowe rozwiązanie generowane jest przy pomocy algorytmu zachłannego, dla losowo wybranego miasta początkowego. Następnie przez określoną liczbę iteracji, wykonywane są kolejne kroki algorytmu:

1. Przeszukiwanie sąsiedztwa obecnie rozpatrywanego rozwiązania poprzez wielokrotną zamianę kolejności odwiedzenia dwóch losowo wybranych miast, a następnie wybranie tej zamiany, która nie należy do tabu oraz dla której droga jest najkrótsza.
2. Dodanie do listy tabu zamiany prowadzącej do otrzymania najlepszego lokalnie rozwiązania.
3. Porównanie najlepszego lokalnie rozwiązania dla danej iteracji z rozwiązaniem najlepszym globalnie. Jeżeli rozwiązanie lokalne jest lepsze od globalnego, nowym rozwiązaniem globalnym staje się obecne rozwiązanie lokalne.

## Elementy charakterystyczne metody

- Lista ruchów będących *tabu*.
- *Zdarzenie krytyczne* - wywołanie określonej czynności, jeśli przez określony odstęp czasu rozwiązanie nie uległo poprawie.
- *Strategia dywersyfikacji* - wygenerowanie nowego rozwiązania początkowego, na którym będą dokonywane zmiany. Jest wywoływana przez zdarzenie krytyczne.
- *Kryterium aspiracji* - ignorowanie ruchu znajdującego się na liście tabu, jeśli ma on szansę na znaczące poprawienie wyniku.

## 2 Plan eksperymentu

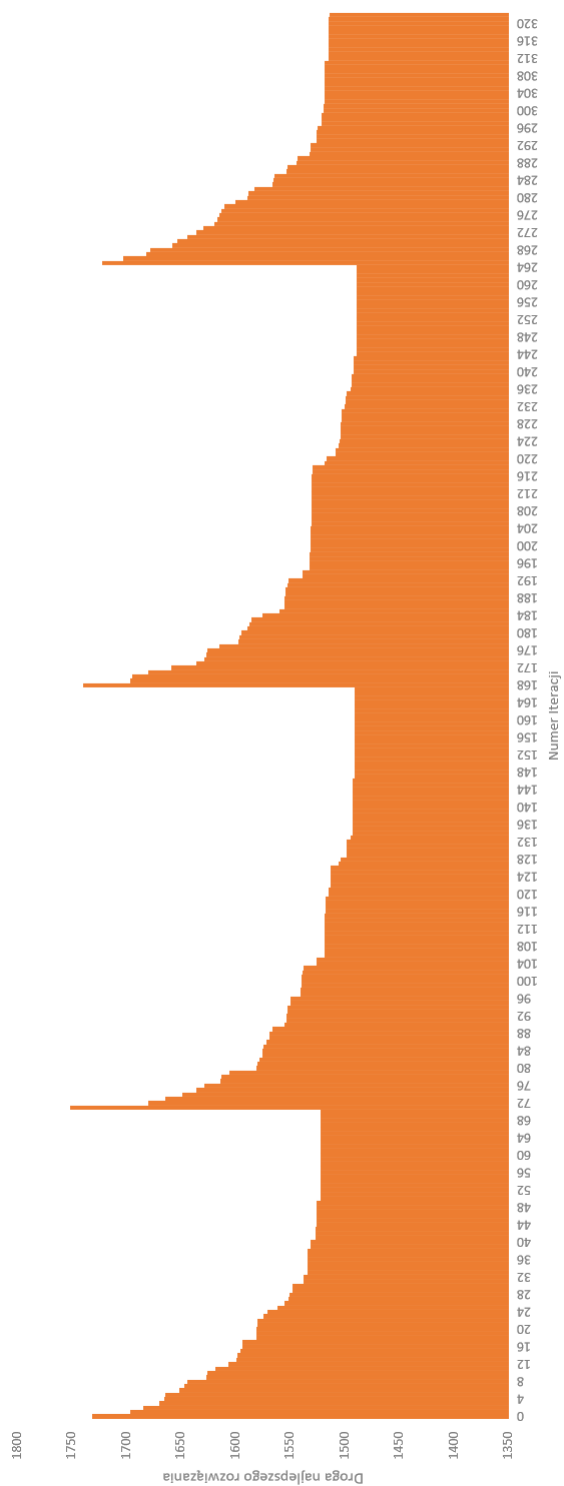
- Pomiar czasu został wykonany za pomocą klasy `StopWatch(QueryPerformanceCounter)`.
- Do reprezentacji odległości między miastami użyto macierzy sąsiedztwa.
- Dla każdego z typów pomiaru algorytm wykonywany był 30 razy, wyniki uśredniono.

### Metoda Tabu Search

- Do przechowywania listy tabu użyto macierzy  $N \times N$ , gdzie  $N$  - liczba miast rozpatrywanego przypadku. Współrzędne  $[i, j]$  oznaczają zamianę kolejności elementu  $i$  z elementem  $j$ .
- Długość listy tabu (ilość iteracji, przez którą ruch będzie na niej obecny) wyliczana jest według wzoru  $\text{ceil}(N/10)$ , gdzie  $N$  to liczba miast dla danej instancji problemu.
- Zdarzenie krytyczne polega na wywołaniu dywersyfikacji, jeśli przez 20 ostatnich iteracji głównej pętli nie nastąpiła poprawa najlepszego rozwiązania.
- Dywersyfikacja polega na obraniu nowego rozwiązania początkowego, na którym będą dokonywane zmiany. Rozwiązanie to jest generowane przy użyciu algorytmu zachłannego, dla losowo wybranego miasta początkowego.
- Kryterium aspiracji anuluje zakaz ruchu, jeśli spełniony jest warunek dla danego rozwiązania lokalnego:  $x * 1.05 < y$ , gdzie  $x$  - waga obecnie rozpatrywanego rozwiązania lokalnego,  $y$  - waga obecnie najlepszego rozwiązania lokalnego.

### 3 Wyniki

#### 3.1 Wpływ strategii dywersyfikacji na działanie algorytmu



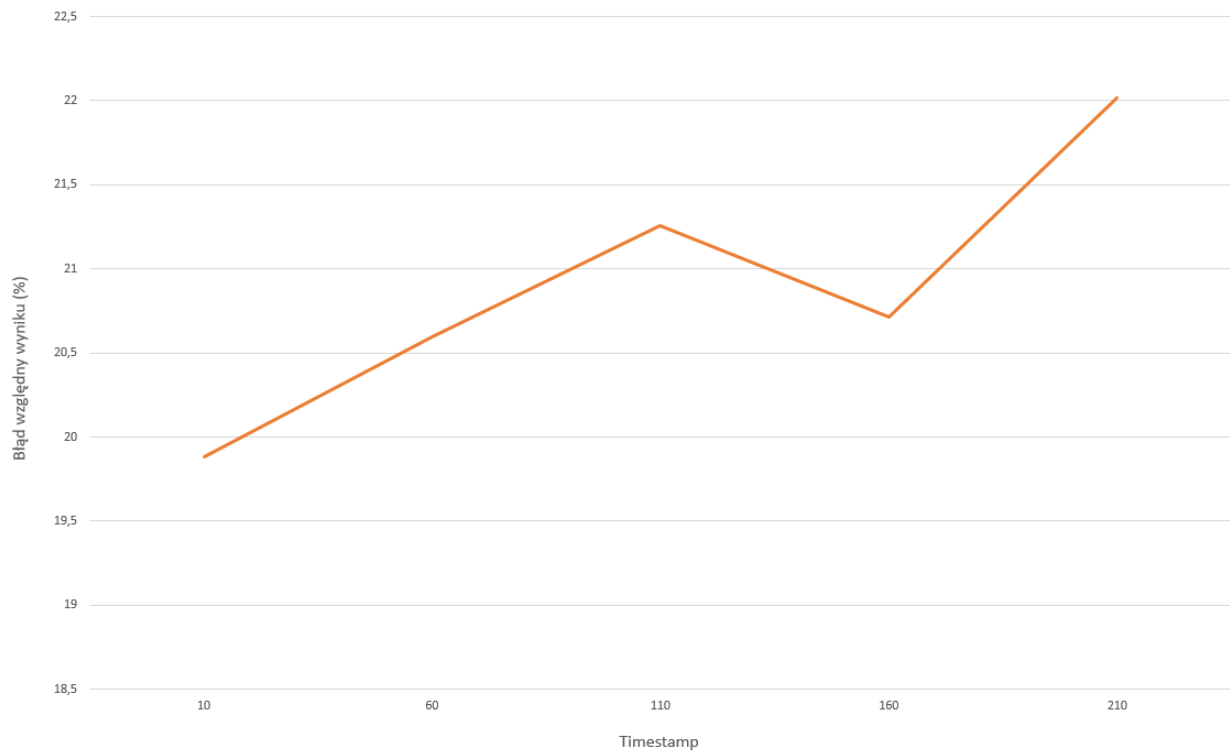
Rysunek 1: Wpływ strategii dywersyfikacji na działanie algorytmu.

W celu zilustrowania wpływu strategii dywersyfikacji na działanie napisanego algorytmu, przyjęto, iż po zdarzeniu krytycznym nowym najlepszym rozwiązaniem zostanie nowo wygenerowane rozwiązanie startowe(za pomocą algorytmu zachłannego).

### 3.2 Zależność jakości rozwiązania od czasu(ilości iteracji) znajdowania się wybranych ruchów na liście tabu

| Timestamp | Błąd względny wyniku (%) |
|-----------|--------------------------|
| 10        | 19,88                    |
| 60        | 20,59                    |
| 110       | 21,15                    |
| 160       | 20,71                    |
| 210       | 22,07                    |

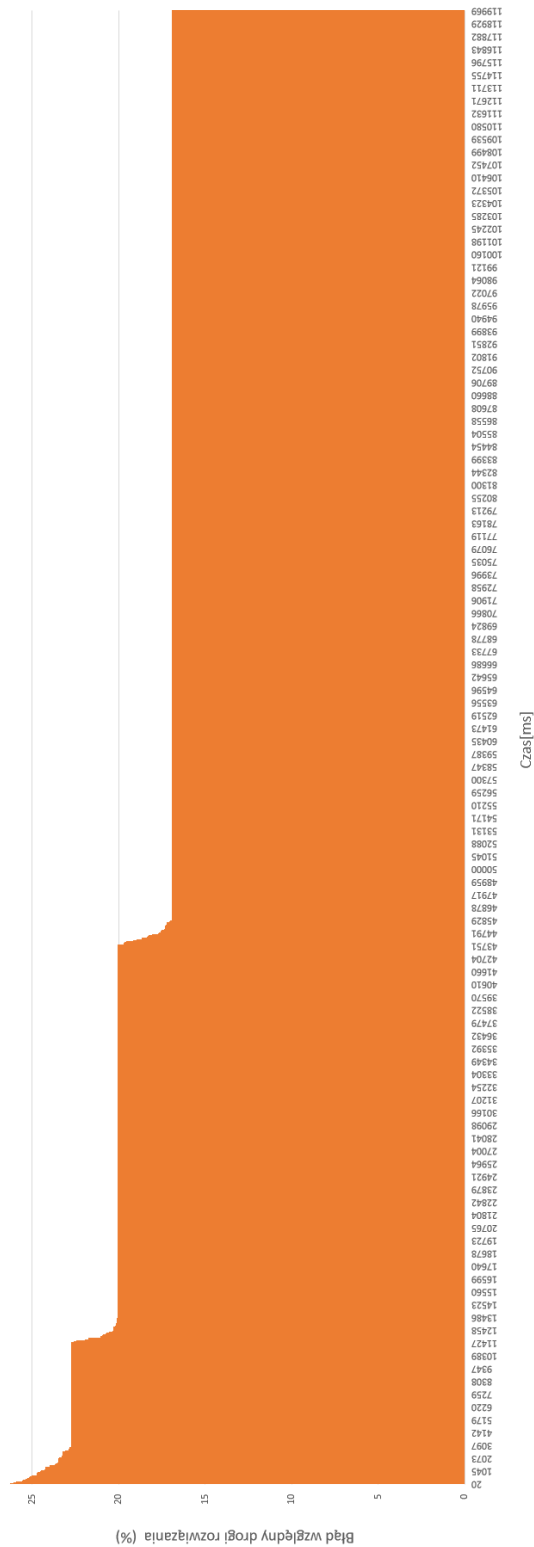
Tablica 1: Zależność jakości rozwiązania od czasu(ilości iteracji) znajdowania się wybranych ruchów na liście tabu, dla instancji rbg323



Rysunek 2: Wpływ jakości rozwiązania od czasu(ilości iteracji) obecności wybranych ruchów na liście tabu, dla instancji rbg323.

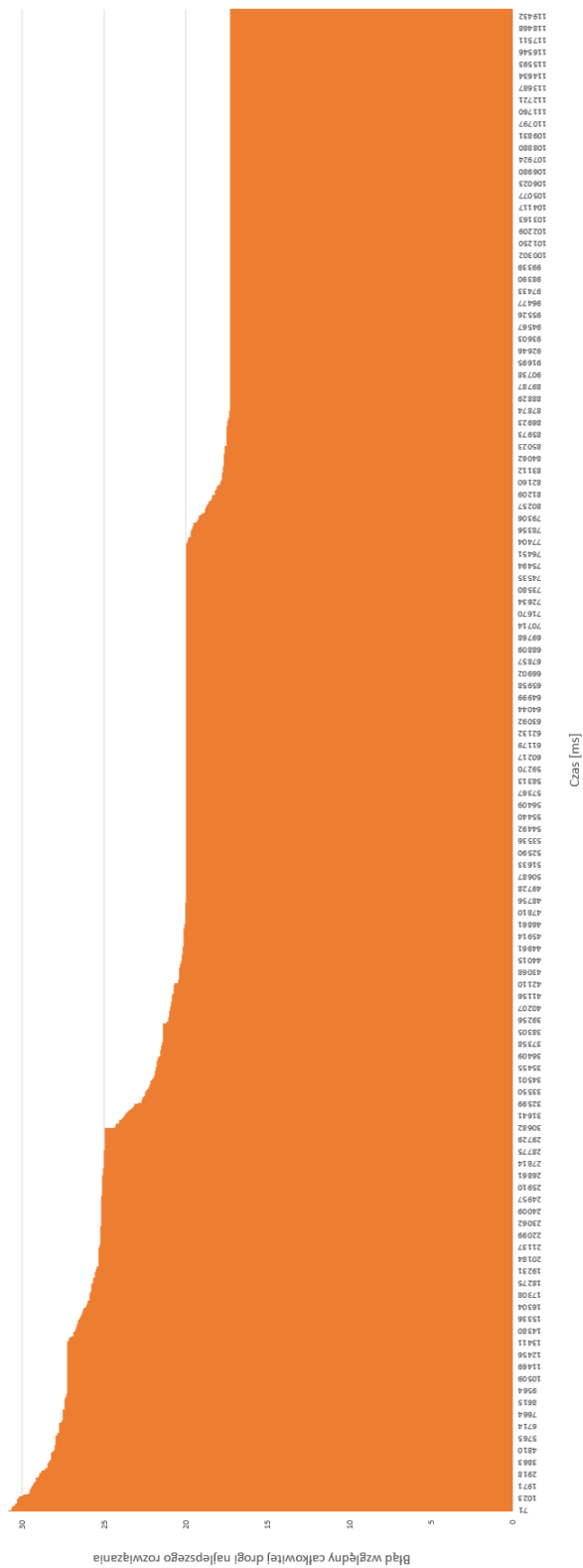
3.3 Zależność jakości rozwiązania od długości wykonywania się algorytmu

3.3.1 rat783

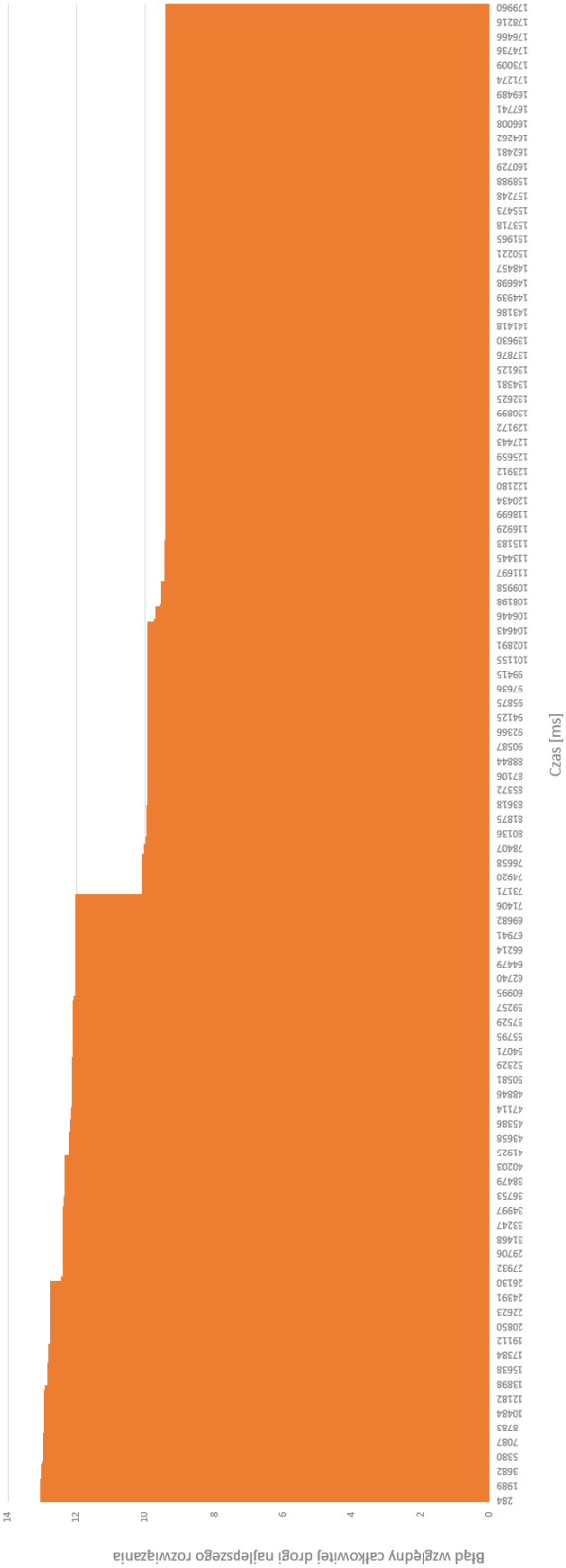


Rysunek 3: Zależność błędu względnego rozwiązania od długości wykonywania algorytmu(ilości iteracji). Czas ograniczono do 2 minut.

3.3.2 pr1002







## 4 Podsumowanie

Na podstawie uzyskanych wyników, można stwierdzić, iż metodę Tabu Search z powodzeniem można wykorzystać jako modyfikację ulepszającą działanie innych algorytmów. W tym przypadku, z powodzeniem został ulepszony algorytm zachłanny. Wyniki uzyskane za jego pomocą zależą od parametrów(3.2), jakie zostały dobrane dla danej instancji problemu. Odpowiednie ich wyselekcjonowanie doprowadziło do uzyskania rezultatów poniżej poziomu 20% błędu względnego, a dla pewnych instancji było to około 10% przy rozsądnym czasie wykonywania. Niestety, nie udało się uzyskać lepszych rezultatów, jednak możliwe, iż algorytm, który zostanie zaimplementowany w ramach następnego zadania projektowego umożliwi poprawienie rezultatów.

Zaimplementowane przez ulepszenia algorytmu pomyślnie wpłynęły na uzyskane wyniki. Na rysunku 1. można zauważyć jak strategia dywersyfikacji pozwala przenieść poszukiwanie najlepszego rozwiązania na nowy obszar, co w zaprezentowanym przykładzie poprawiło rezultat (3. generacja nowego rozwiązania). Dodatkowo, kryterium aspiracji pozwala na złamanie tabu, jeśli potencjalny ruch może prowadzić do znacznie lepszego rozwiązania niż obecne.

Napisany podczas tego projektu algorytm, pozwala - w odróżnieniu od metody *podziału i ograniczeń* - na pewną elastyczność. To od użytkownika zależy, czy algorytm ma charakteryzować się skutecznością znajdowania lepszych rozwiązań, czy krótkim czasem działania. Pozwala on tym samym na uzyskanie wyników dla różnej wielkości instancji - metoda *branch and bound* mogła nie zapewnić rozwiązania w akceptowalnym czasie już dla 17 miast, gdzie *tabu search* z powodzeniem uzyskuje zadowalające wyniki(10% różnicy względem najlepszego rozwiązania) dla instancji o rozmiarze 2100 miast.