

Zadanie projektowe 2

Sprawozdanie

Kurs: Projektowanie efektywnych algorytmów

Prowadzący: mgr inż. Antoni Sterna

Grupa: E01-60h (środa 13:15-15:00)

Autor: Mirosław Kuźniar (nr indeksu: 248870)

1 Wstęp teoretyczny

1.1 Problem optymalizacyjny

Rozważanym problemem optymalizacyjnym jest problem komiwojażera (ang. Travelling Salesman Problem). Zgodnie z definicją polega on na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym. W praktyce często opisuje się go jako problem komiwojażera/wędrownego sprzedawcy który wyruszając z miasta początkowego musi odwiedzić każde z pozostałych miast tylko raz i wrócić do miasta początkowego. Ponadto trasa, którą przebędzie musi być minimalna pod względem przyjętego kosztu (np. odległości, czasu, kosztów ekonomicznych itp.). Warto także wspomnieć, że zgodnie z definicją grafu pełnego musi istnieć połączenie z danego miasta do wszystkich pozostałych.

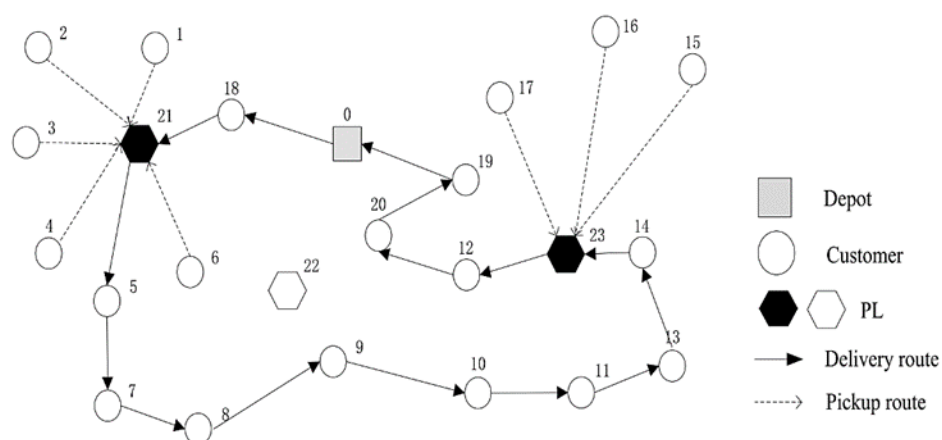
Problem komiwojażera należy do klasy problemów NP – trudnych. Problemy z tej klasy opisywane są jako takie dla których (najprawdopodobniej) nie można skonstruować algorytmów wielomianowych, czyli algorytmów efektywnych obliczeniowo.

Wyróżniamy dwa rodzaje problemu komiwojażera:

- Symetryczny problem komiwojażera (STSP), w którym waga krawędzi przyjmuje jedną wartość bez względu na kierunek poruszania się, tzn. odległość z miasta A do miasta B jest taka sama jak z miasta B do miasta A,
- Asymetryczny problem komiwojażera (ATSP), w którym waga krawędzi może przyjmować różne wartości w zależności od kierunku przemieszczania się, tzn. odległość z miasta A do miasta B może być różna niż z miasta B do miasta A.

Rozważanym dalej rodzajem będzie asymetryczny problem komiwojażera (ATSP).

Pomimo swojej złożoności problem komiwojażera nie jest tylko tematem dysput akademickich ale także znajduje zastosowanie w praktycznych aplikacjach. Dla przykładu, wydajne rozwiązania TSP są często wykorzystywane w tzw. dostawach ostatniej mili (ang. last mile delivery), gdzie towar przemieszczany jest z węzła transportowego, takiego jak skład lub magazyn do klienta. Dostawy te stanowią główne źródło różnic kosztów w całym łańcuchu dostaw. Dlatego wiele firm z sektora logistyki dąży do zminimalizowania kosztów dostawy ostatniej mili wykorzystując wydajne algorytmy bazujące na problemie komiwojażera.



1.2 Sąsiedztwo

Sąsiedztwo to zbiór rozwiązań bliskich rozwiązaniu x . Zazwyczaj oznaczane jest przez $N(x)$. W przypadku problemów dyskretnych, jakim jest m.in. problem komiwożacza, definiuje się je przez wszystkie możliwe transformacje x . Transformację stanowi ruch m ze zbioru możliwych ruchów $M(x)$, który prowadzi do uzyskania nowego rozwiązania x' będącego w sąsiedztwie x .

1.2.1 Rodzaje sąsiedztwa:

- Sąsiedztwo typu Insert
Ruch typu Insert zdefiniowany jest przez parę liczb $(i, j), i = 1, \dots, n, j = 1, \dots, n, i \neq j$ i powoduje usunięcie elementu z pozycji i -tej i wstawienie go na pozycję j -tą.
- Sąsiedztwo typu Swap
Ruch typu Swap zdefiniowany jest przez parę liczb $(i, j), i = 1, \dots, n, j = 1, \dots, n, i \neq j$ i powoduje zamianę elementu z pozycji i -tej z elementem z pozycji j -tej.
- Sąsiedztwo typu Invert
Ruch typu Invert zdefiniowany jest przez parę liczb $(i, j), i = 1, \dots, n, j = 1, \dots, n, i \neq j$ i powoduje odwrócenie w permutacji ciągu od elementu z pozycji i -tej do elementu z pozycji j -tej.

1.3 Algorytmy heurystyczne

Algorytmy heurystyczne są to algorytmy, które w ogólnym przypadku, nie dają gwarancji znalezienia rozwiązania optymalnego. Umożliwiają jednak znalezienie rozwiązania dość dobrego w rozsądnym czasie. Algorytmy tego typu używane są w problemach optymalizacyjnych, gdzie znalezienie rozwiązania optymalnego ma zbyt dużą złożoność obliczeniową lub w ogóle nie jest możliwe. W szczególności są to problemy NP-trudne.

1.3.1 Symulowane wyżarzanie

W metalurgii wyżarzaniem określa się powolne i stopniowe ochładzanie materiału dzięki czemu cząsteczki, oddając energię, tworzą równomierne struktury. Celem obróbki jest przybliżenie stanu materiału do warunków równowagi.

Algorytm symulowanego wyżarzania powstał poprzez analogię do ww. procesu fizycznego. Jest także rozwinięciem metod iteracyjnych, polegających na ciągłym ulepszaniu istniejącego rozwiązania. Główną wadą metod iteracyjnych jest możliwość zatrzymania w rozwiązaniu pseudo-optymalnym będącym jedynie ekstremum lokalnym.

W celu uniknięcia ww. wady algorytm symulowanego wyżarzania wprowadza możliwość wyboru gorszego rozwiązania z pewnym prawdopodobieństwem. Dzięki temu, może w określonych warunkach wyjść ze znalezionej minimum lokalnego i dalej podążać w kierunku rozwiązania optymalnego

Ogólny schemat algorytmu symulowanego wyżarzania prezentuje się następująco:

```
BEGIN
  R := R0 {rozwiązanie początkowe};
  T := T0 {temperatura początkowa > 0};
  wybierz funkcję redukcji temperatury  $\alpha$ ;
  WHILE (niespełniony warunek zatrzymania) DO
    BEGIN
      WHILE (nieosiągnięty stan równowagi) DO
        BEGIN
          R' := f(R) {nowe, sąsiednie rozwiązanie,
            generowane losowo};
          dE := E(R') - E(R);
          IF (dE < 0) THEN R := R'
          ELSE IF (RANDOM < exp(-dE/T)) THEN R := R'
        END;
      T :=  $\alpha$ (T);
    END;
  END;
```

Elementy algorytmu symulowanego wyżarzania:

- Temperatura początkowa – jest parametrem, który ma wpływ na prawdopodobieństwo wyboru gorszego rozwiązania. W fazie eksploracji temperatura jest wysoka, dzięki czemu prawdopodobieństwo wyboru gorszego rozwiązania jest wysokie. Wraz z przechodzeniem do fazy eksploatacji temperatura maleje przez co częściej wybierane są rozwiązania lepsze. Pod koniec pracy algorytmu, temperatura jest na tyle niska, że prawdopodobieństwo wyboru gorszego rozwiązania jest bliskie zeru. Algorytm zachowuje się wówczas, jak typowy algorytm iteracyjny i stara się maksymalnie ulepszyć rozwiązanie.
- Długość epoki (stan równowagi) – ma bezpośredni wpływ na ilość dokonywanych zmian w rozwiązaniu przy tej samej wartości temperatury, tym samym zwiększając szansę na znalezienie lepszych rozwiązań. Dobór długości epoki ma wpływ na dokładność i czas działania algorytmu. W ogólnym przypadku dłuższa epoka oznacza dokładniejsze działanie, ale też dłuższy czas pracy.
- Funkcja redukcji temperatury – istnieje wiele schematów chłodzenia, m.in. logarytmiczny czy liniowy (Cauchy'ego). Jednakże najczęściej wybieranym jest schemat geometryczny sprowadzający redukcję temperatury do operacji $T = \alpha T$. Wartości α , w praktycznych zastosowaniach, nie powinny być mniejsze niż 0,85.
- Kryterium zatrzymania – jest parametrem, który w największym stopniu podlega modyfikacjom w konkretnych implementacjach algorytmu symulowanego wyżarzania. Kryterium zatrzymania przyjętym w tej implementacji jest czas pracy algorytmu zadany przez użytkownika.

1.3.2 Poszukiwanie z zakazami

Poszukiwanie z zakazami (ang. Tabu Search) zawdzięcza swoją nazwę inspiracją zjawiskami społecznymi. W religiach ludów pierwotnych Tabu oznacza zakaz podejmowania różnych działań w stosunku do osób, miejsc, zwierząt, rzeczy bądź stanów rzeczy lub też wypowiedzania pewnych słów.

W ujęciu algorytmicznym Tabu Search oznacza wykorzystanie zakazów/ograniczeń w celu przeszukiwania przestrzeni dopuszczalnych rozwiązań ze zmniejszonym ryzykiem zatrzymania w ekstremum lokalnym.

Algorytm Tabu Search opisywany jest przez twórcę jako metoda dynamicznej zmiany sąsiedztwa danego rozwiązania. Oznacza to, że sąsiedztwo danego rozwiązania nie jest z góry ustalone dla każdego rozwiązania lecz może zmieniać się w zależności od informacji zebranych w poprzednich etapach przeszukiwania.

Ogólny schemat algorytmu poszukiwania z zakazami prezentuje się następująco:

```
wybierz lub wylosuj punkt startowy  $x_0 \in X$ 
 $x_{opt} \leftarrow x_0$ 
 $tabu\_list \leftarrow \emptyset$ 
repeat
    znajdź  $x \in N(x_0)$ , dla którego  $M_{val}(x_0, x)$  jest najlepsza
     $x_0 \leftarrow x$ 
    if  $f(x_0) < f(x_{opt})$  then
         $x_{opt} \leftarrow x_0$ 
    zweryfikuj  $tabu\_list$ 
     $\forall element \in tabu\_list$  do
        --kadencjai
        if  $kadencja_i = 0$  then
            usuń  $element(atrybut_i, kadencja_i)$  z  $tabu\_list$ 
until warunek zakończenia
```

Elementy algorytmu poszukiwania z zakazami:

- Lista ruchów zakazanych – zawiera zbiór ruchów zakazanych oraz liczbę iteracji przez którą dany ruch nie może zostać wykonany – kadencję. Użycie listy ruchów zakazanych zmniejsza prawdopodobieństwo wpadnięcia w cykl.
- Kadencja – określa przez jaką ilość iteracji dany ruch jest zakazany. Kadencja krótka zwiększa dokładność przeszukiwania kosztem zwiększonego ryzyka wpadnięcia w cykl. Kadencja długa poszerza zakres przeszukiwana kosztem pogorszonej dokładności przeszukiwania sąsiedztwa.
- Kryterium aspiracji – pozwala, pod pewnymi warunkami, uniknąć zakazu bardzo dobrych ruchów. Jest pewnym warunkiem narzuconym na rozwiązanie, które pozwala na wykonanie ruchu nawet jeżeli jest zakazany.

2 Najważniejsze aspekty implementacji

2.1 Symulowane wyżarzanie

2.1.1 Generowanie rozwiązania początkowego

Przyjętym sposobem na wytworzenie rozwiązania początkowego jest generowanie go w sposób losowy.

Aby uniknąć problemów związanych z uzyskiwaniem rozwiązań niepoprawnych tzn. takich, w których w permutacji miasto występuje więcej niż raz użyto funkcji `std::random_shuffle`.

Umożliwiła ona losowe przetasowanie wektora zawierającego każde miasto wyłącznie raz.

2.1.2 Dobór temperatury początkowej

W przypadku algorytmu symulowanego wyżarzania sposób doboru temperatury początkowej może zostać wykonany na wiele sposobów.

W początkowej wersji implementacji temperatura została dobrana w sposób eksperymentalny i nie zależny od badanej instancji. Była to stała wartość, zazwyczaj z zakresu 100 – 1000.

W późniejszej wersji dobór temperatury został wykonany w taki sposób, by prawdopodobieństwo wyboru gorszego rozwiązania malało w równomierny sposób od 1 (faza eksploracji) do 0 (faza eksploatacji), wraz z przebiegiem algorytmu. Aby poprawnie określić temperaturę początkową posłużono się wzorem:

$$0,99 = e^{\frac{-\Delta}{t_0}}$$

Ponadto przeprowadzono kilkunastokrotne badanie wartości Δ oraz ustalenie średniej.

Użycie takiego sposobu spowodowało zmniejszenie ogólnego błędu względnego, a także mniejszy wpływ badanej instancji na uzyskiwany błąd względny.

2.2 Poszukiwanie z zakazami

2.2.1 Dobór kadencji

W implementacji wielkość kadencji uzależniona została od wielkości instancji. Użyto współczynnika λ przez który mnożona jest wielkość instancji i na tej podstawie wyznaczana jest wartość kadencji. Wartość współczynnika została dobrana eksperymentalnie.

2.2.2 Kryterium aspiracji

Kryterium aspiracji zaimplementowano w taki sposób, by danych ruch znajdujący się na Liście Tabu mógł zostać wykonany tylko gdy prowadzi on do uzyskania rozwiązania lepszego niż najlepsze do tej pory.

$$f(y) < f(x_{opt})$$

3 Implementacja algorytmów

Wybrany językiem programowania jest C++. Algorytmy operują na zmiennych dziesiętnych `int` lub zmiennoprzecinkowych `float`. Do przechowywania danych wykorzystano strukturę danych z biblioteki STL – `vector`.

4 Plan eksperymentu

4.1 Rozmiar problemu

Zarówno w przypadku algorytmu symulowanego wyżarzania jak i poszukiwania z zakazami testy efektywności przeprowadzono dla trzech przykładów wyraźnie różniących się rozmiarem. Wykorzystano instancje o rozmiarze 17, 43 i 403 miast.

4.2 Dane testowe

Przykłady testowe zaczerpnięto ze strony http://staff.iiar.pwr.wroc.pl/antoni.sterna/pea/PEA_testy.htm. Wykorzystano pliki br17 o rozwiązaniu optymalnym 39, p43 o rozwiązaniu optymalnym 5620 oraz rbg403 o rozwiązaniu optymalnym 2465.

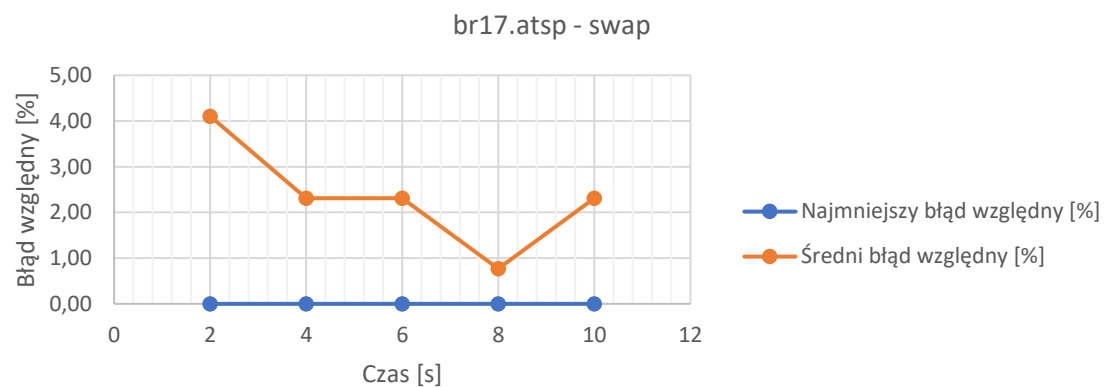
4.3 Pomiar czasu

Do pomiaru czasu użyta została rozbudowana wersja funkcji `read_QPC()`, zawierająca funkcję `QueryPerformanceFrequency()`. Zastosowanie jej umożliwia pomiar czasu w rozdzielczości mikrosekundowej.

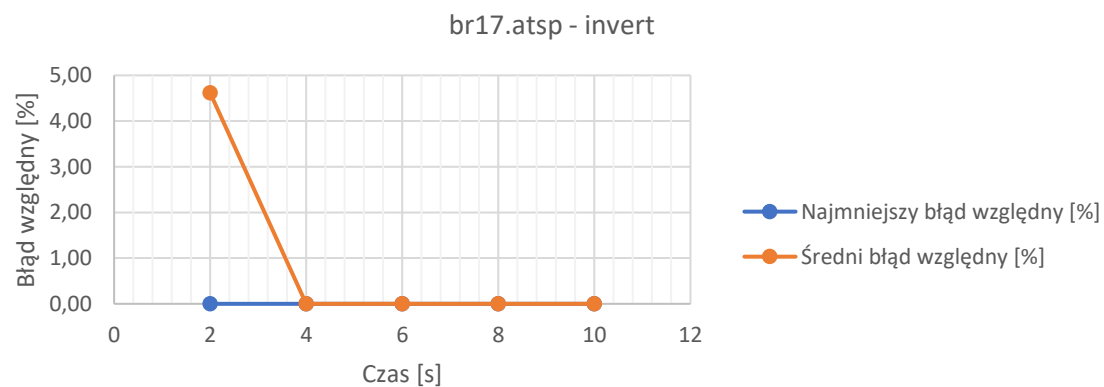
5 Wyniki pomiarów

5.1 Symulowanie wyżarzanie

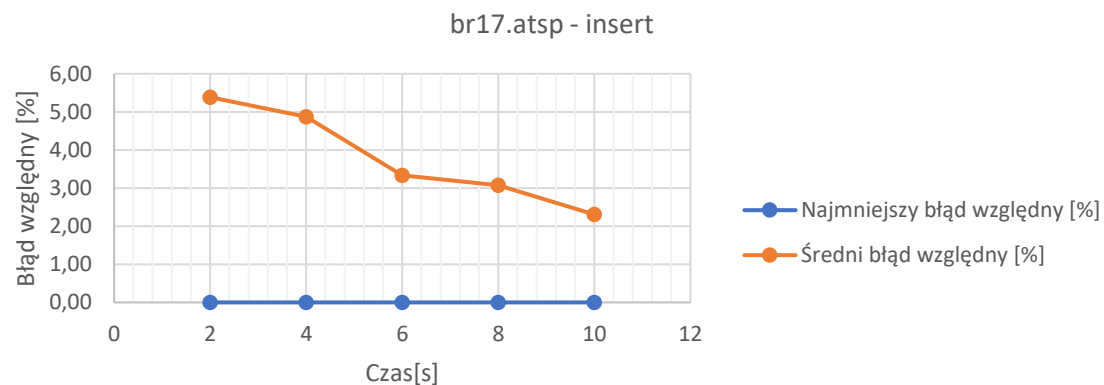
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 39 | 0,00 | 40,6 | 4,10 |
| 4 | 39 | 0,00 | 39,9 | 2,31 |
| 6 | 39 | 0,00 | 39,9 | 2,31 |
| 8 | 39 | 0,00 | 39,3 | 0,77 |
| 10 | 39 | 0,00 | 39,9 | 2,31 |



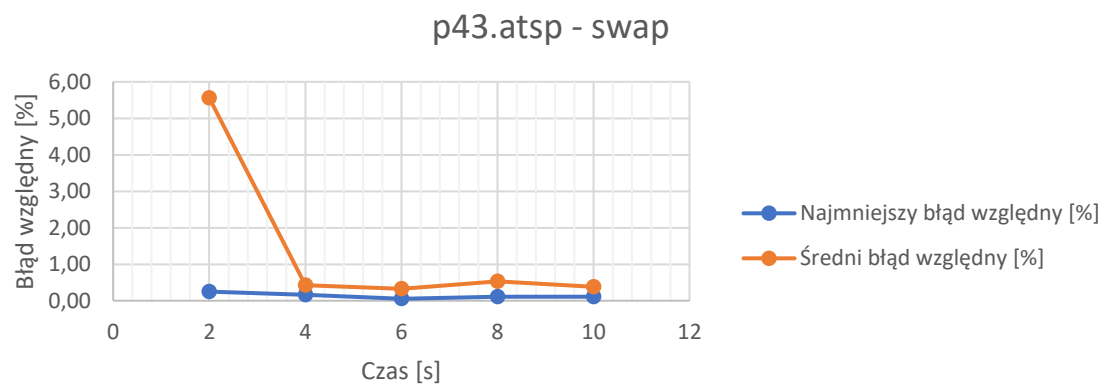
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 39 | 0,00 | 40,8 | 4,62 |
| 4 | 39 | 0,00 | 39 | 0,00 |
| 6 | 39 | 0,00 | 39 | 0,00 |
| 8 | 39 | 0,00 | 39 | 0,00 |
| 10 | 39 | 0,00 | 39 | 0,00 |



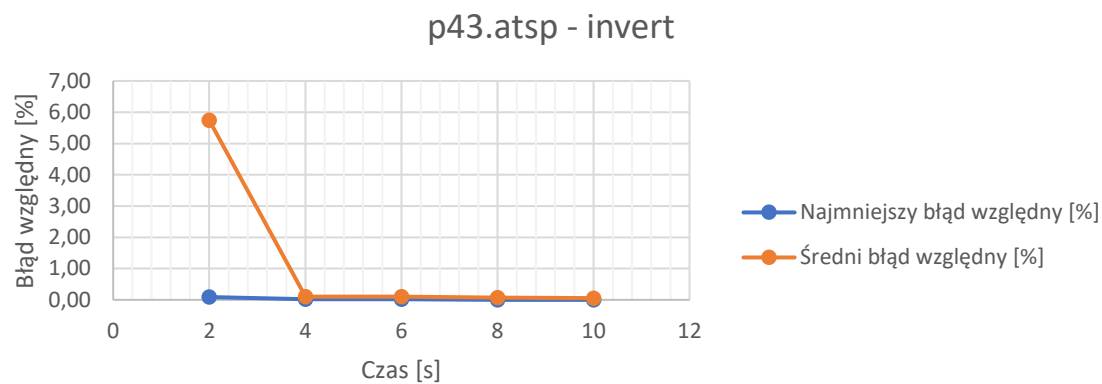
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 39 | 0,00 | 41,1 | 5,38 |
| 4 | 39 | 0,00 | 40,9 | 4,87 |
| 6 | 39 | 0,00 | 40,3 | 3,33 |
| 8 | 39 | 0,00 | 40,2 | 3,08 |
| 10 | 39 | 0,00 | 39,9 | 2,31 |



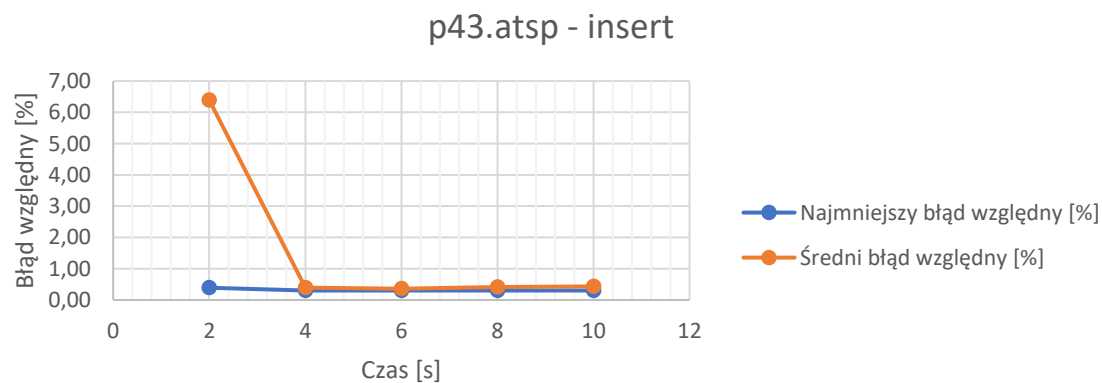
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|--|------------------------------|-----------------------------------|
| 2 | 5642 | 0,39 | 5979,1 | 6,39 |
| 4 | 5637 | 0,30 | 5642,3 | 0,40 |
| 6 | 5637 | 0,30 | 5640,2 | 0,36 |
| 8 | 5637 | 0,30 | 5643 | 0,41 |
| 10 | 5637 | 0,30 | 5644,2 | 0,43 |



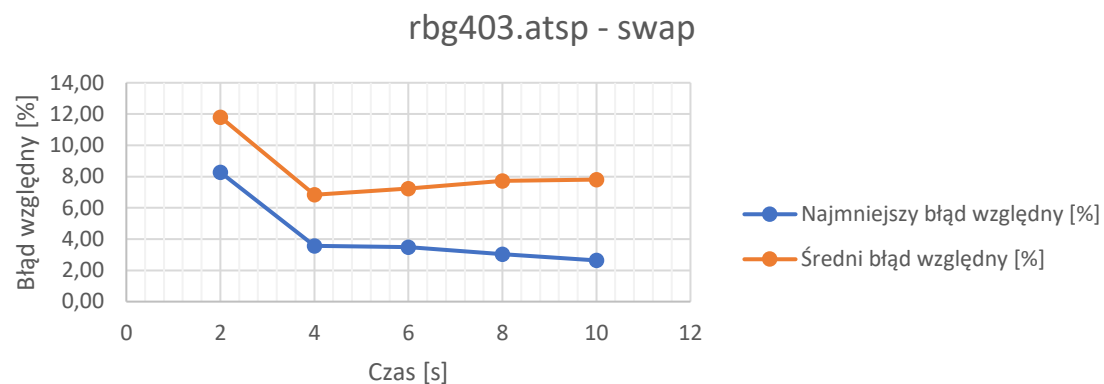
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|--|------------------------------|-----------------------------------|
| 2 | 5625 | 0,09 | 5942,7 | 5,74 |
| 4 | 5621 | 0,02 | 5625,6 | 0,10 |
| 6 | 5621 | 0,02 | 5626 | 0,11 |
| 8 | 5620 | 0,00 | 5624,2 | 0,07 |
| 10 | 5620 | 0,00 | 5623,1 | 0,06 |



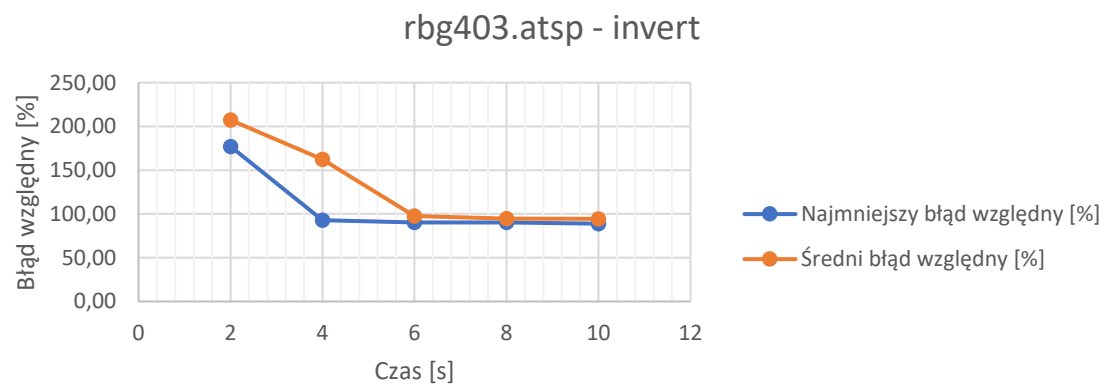
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|--|------------------------------|-----------------------------------|
| 2 | 5634 | 0,25 | 5932,4 | 5,56 |
| 4 | 5629 | 0,16 | 5644,1 | 0,43 |
| 6 | 5623 | 0,05 | 5638,3 | 0,33 |
| 8 | 5626 | 0,11 | 5649,7 | 0,53 |
| 10 | 5626 | 0,11 | 5641,6 | 0,38 |



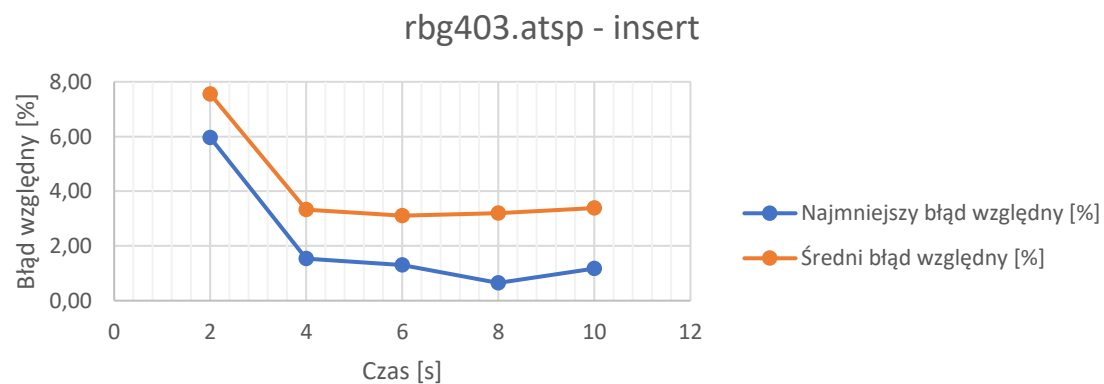
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 2669 | 8,28 | 2756,1 | 11,81 |
| 4 | 2553 | 3,57 | 2633,5 | 6,84 |
| 6 | 2551 | 3,49 | 2643,2 | 7,23 |
| 8 | 2540 | 3,04 | 2655,6 | 7,73 |
| 10 | 2530 | 2,64 | 2657,7 | 7,82 |



| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 6831 | 177,12 | 7575,3 | 207,31 |
| 4 | 4754 | 92,86 | 6466,9 | 162,35 |
| 6 | 4686 | 90,10 | 4871,6 | 97,63 |
| 8 | 4689 | 90,22 | 4798,3 | 94,66 |
| 10 | 4656 | 88,88 | 4792,8 | 94,43 |

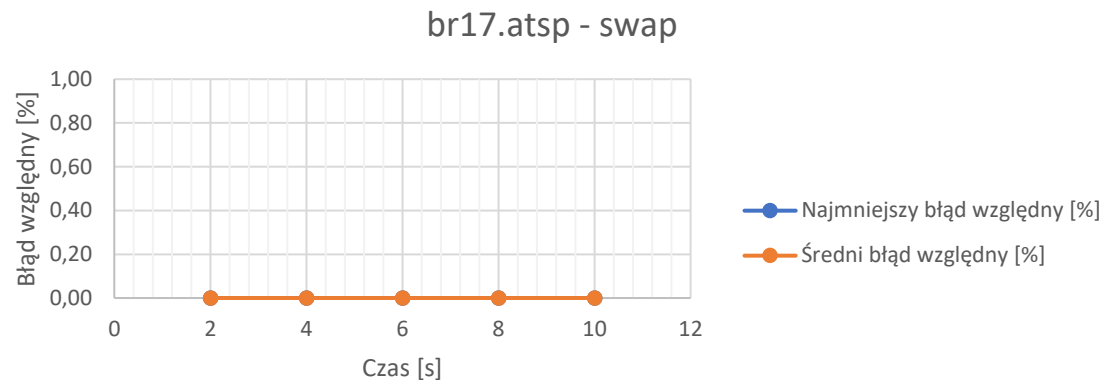


| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 2612 | 5,96 | 2651,2 | 7,55 |
| 4 | 2503 | 1,54 | 2547,1 | 3,33 |
| 6 | 2497 | 1,30 | 2541,6 | 3,11 |
| 8 | 2481 | 0,65 | 2543,7 | 3,19 |
| 10 | 2494 | 1,18 | 2548,4 | 3,38 |

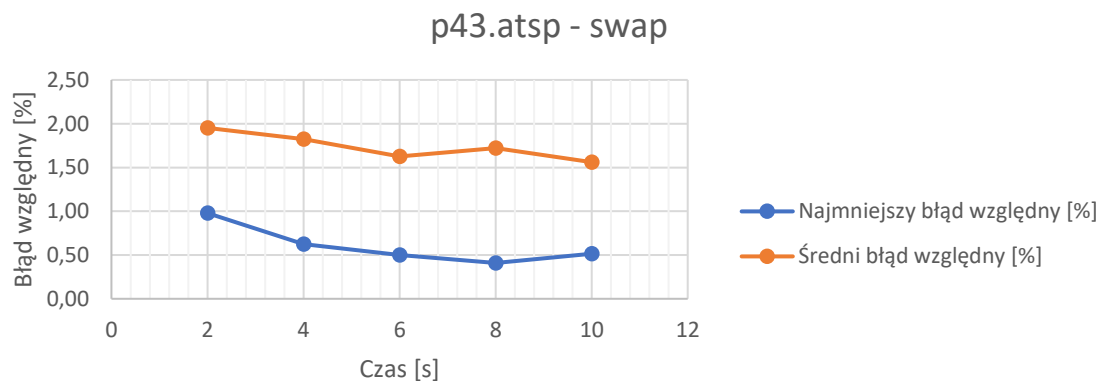


5.2 Poszukiwanie z zakazami

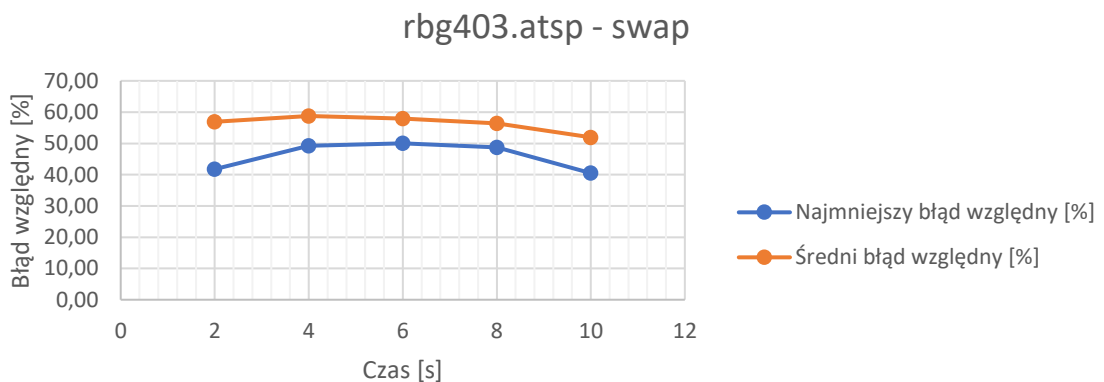
| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 39 | 0,00 | 39 | 0,00 |
| 4 | 39 | 0,00 | 39 | 0,00 |
| 6 | 39 | 0,00 | 39 | 0,00 |
| 8 | 39 | 0,00 | 39 | 0,00 |
| 10 | 39 | 0,00 | 39 | 0,00 |



| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 5675 | 0,98 | 5729,7 | 1,95 |
| 4 | 5655 | 0,62 | 5722,4 | 1,82 |
| 6 | 5648 | 0,50 | 5711,3 | 1,62 |
| 8 | 5643 | 0,41 | 5716,8 | 1,72 |
| 10 | 5649 | 0,52 | 5707,7 | 1,56 |



| Czas[s] | Najmniejszy wynik otrzymany | Najmniejszy błąd względny [%] | Średni wynik otrzymany | Średni błąd względny [%] |
|---------|-----------------------------------|-------------------------------------|------------------------------|-----------------------------------|
| 2 | 3495 | 41,78 | 3867,1 | 56,88 |
| 4 | 3678 | 49,21 | 3913,5 | 58,76 |
| 6 | 3698 | 50,02 | 3892,1 | 57,89 |
| 8 | 3666 | 48,72 | 3856,3 | 56,44 |
| 10 | 3463 | 40,49 | 3745,5 | 51,95 |



6 Dyskusja wyników

6.1 Symulowane wyżarzanie

W przypadku algorytmu symulowanego wyżarzania przeanalizowany zostanie wpływ doboru sąsiedztwa na uzyskiwany błąd względny.

W przypadku najmniejszej instancji – br17 użycie wszystkich trzech sąsiedztw prowadzi do uzyskania rozwiązania optymalnego. Jednak biorąc pod uwagę średnią wartość rozwiązania znacząco wyróżnia się sąsiedztwo typu Invert – tylko w przypadku czasu 2 sekund średnia jest różna od rozwiązania optymalnego. W przypadku sąsiedztw Insert i Swap uzyskiwane średnie są nieco gorsze niż w przypadku Invert ale także zadowalające. W najgorszym przypadku błąd względny nie przekracza 10%.

W przypadku instancji p43 różnice pomiędzy sąsiedztwami wydają się być mniej widoczne. Warte uwagi jest to, że mimo większego rozmiaru, użycie sąsiedztwa Invert prowadzi do uzyskania rozwiązania optymalnego. Pozostałe sąsiedztwa nie osiągają rozwiązania optymalnego, ale zbliżają się do niego na tyle blisko, że błąd względny nie przekracza 1%. W przypadku błędu średniego, nawet przy krótkim czasie 2 sekund nie przekracza 7%. Zwiększenie czasu powoduje zmniejszenie średniego błędu względnego do wartości poniżej 1%.

W przypadku największej instancji rbg403 różnice pomiędzy sąsiedztwami stają się bardziej widoczne. Tak duży rozmiar problemu sprawia, że sąsiedztwo Invert znacząco traci na jakości. Wyróżniają się natomiast sąsiedztwa Insert oraz Swap. Obie pozwalają na uzyskanie błędu względnego poniżej 15%. Jednak nieco na przód wysuwa się sąsiedztwo Insert dla którego najmniejszy błąd względny oscyluje w okolicach 1%.

6.2 Poszukiwanie z zakazami

W przypadku poszukiwania z zakazami przeanalizowany zostanie wpływ rozmiaru instancji na uzyskiwany błąd względny.

W przypadku najmniejszej instancji br17 zauważyć można, że niezależnie od skracania czasu wykonania algorytmu pozwala on na uzyskanie rozwiązania optymalnego zawsze. Sprawia to że zarówno najmniejszy jak i średni błąd względny osiąga wartość 0%.

W przypadku instancji p43 uzyskiwanie wyniki nie przyjmują wartości rozwiązania optymalnego, ale zbliżają się do niego w znaczący sposób. Nawet dla krótkich czasów pracy algorytmu najbardziej pesymistyczna wartość błędu względnego nie przekracza 3%.

W przypadku największej instancji – rbg403 zauważyć można ograniczenia czasowe i ich wpływ na jakość uzyskiwanych rozwiązań. Przy tak dużej instancji przeglądanie sąsiedztwa jest znacznie bardziej czasochłonne co uwydatnia się przez większą wartość błędu względnego. Jednak przy najdłuższym badanym czasie pracy wyniki można uznać za zadowalające gdyż oscylują w granicach 50% błędu względnego.

7 Wnioski

Podczas formułowania wniosków należy pamiętać, że testowaniu podlegały algorytmy heurystyczne. Jedną z ich cech jest to, że nie dają gwarancji uzyskania rozwiązania optymalnego. W ogólności jednak są w stanie uzyskać rozwiązanie zbliżone do optymalnego w akceptowalnym czasie. Zarówno wyniki badania algorytmu symulowanego wyżarzania jak i poszukiwania z zakazami wpisują się w te cechy. Przy ograniczonym czasie są w stanie uzyskać dobre wyniki. Warto także zauważyć, że zwiększenie czasu pracy algorytmu wpływa pozytywnie na jakość zwracanych przez niego rozwiązań.

W przypadku symulowanego wyżarzania implementacja różnych rodzajów sąsiedztwa pozwalała na zauważenie wpływu wyboru sąsiedztwa na jakość wyników. W ogólności sąsiedztwo Invert cechowało się najlepszą jakością rozwiązań ale tylko przy niewielkich rozmiarach problemu. Sąsiedztwo Insert przewyższało pozostałe dla średnich i dużych rozmiarów problemu, a sąsiedztwo Swap charakteryzowało się przyzwoitą jakością wyników niezależnie od rozmiaru.

Wyniki działania algorytmu poszukiwania z zakazami pozwalają zobrazować jak wielkość instancji ma wpływ na jakość rozwiązań przy ograniczonym czasie. Algorytm cechował się bardzo dobrą jakością zwracanych rozwiązań przy niewielkich rozmiarach problemu, która niestety malała wraz z wzrostem rozmiaru instancji.

Porównując zaimplementowane algorytmy ze sobą należy pamiętać, że różnią się one stopniem rozbudowania. Algorytm symulowanego wyżarzania dysponując trzema rodzajami sąsiedztwa pozwala na lepsze dopasowanie do problemu. Przy testowaniu innych niż wymienione w tym sprawozdaniu instancje zaleca się użycie właśnie jego z wyborem sąsiedztwa typu Insert. Algorytm Poszukiwania z zakazami dysponując tylko sąsiedztwem Swap nie ma możliwości tak uniwersalnego działania bez odpowiedniej manipulacji jego parametrami. Jednak zauważyć można potencjał tego algorytmu, szczególnie jeśli chodzi o czas pracy oraz wartości błędu względnego przy odpowiednio dobranych parametrach. Można więc podejrzewać, że przy podobnym stopniu rozbudowania co symulowane wyżarzanie wskazane by było użycie algorytmu poszukiwania z zakazami.