



Projektowanie Efektywnych Algorytmów	
Kierunek <i>Informatyka</i>	Termin <i>Czwartek 19:05</i>
Temat <i>Algorytmy poszukiwania lokalnego</i>	Problem <i>(A)TSP</i>
Skład grupy <i>241284 Jakub Płona</i>	Nr grupy <i>-</i>
Prowadzący <i>Mgr inż. Radosław Idzikowski</i>	data <i>19 grudnia 2019</i>

Spis treści

1	Opis problemu	3
2	Metody i algorytmy rozwiązywania problemu	3
2.1	Symulowane wyżarzanie	3
2.2	Poszukiwanie z zakazami	4
3	Eksperymenty obliczeniowe	5
3.1	Analiza wpływu parametrów algorytmu na jakość rozwiązań	6
3.1.1	Symulowane wyżarzanie	7
3.1.2	Poszukiwanie z zakazami	10
3.2	Testy czasowe i jakościowe	13
3.2.1	Tabela - zestawienie danych pomiarowych	13
3.2.2	Analiza wyników przeprowadzonych testów	14
4	Wnioski	15

1 Opis problemu

Jak można przeczytać na Wikipedii,

problem komiwojażera (ang. travelling salesman problem, TSP) to zagadnienie optymalizacyjne, polegające na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym.

Problem ten można przedstawić bez wykorzystania terminologii teorii grafów. Problem komiwojażera jest problemem optymalizacyjnym z danymi wejściowymi w postaci zbioru n miast oraz danymi kosztami przejścia między dowolnie wybraną parą miast, za wyjątkiem przejścia z danego miasta do tego samego miasta (przejście to nie istnieje). W problemie tym poszukuje się drogi o najmniejszym koszcie przejścia, przechodzącej przez każde miasto dokładnie raz, wracającej do miasta początkowego. Jakość rozwiązania ocenia się za pomocą tzw. funkcji celu, która jest definiowana jako suma kosztów przejścia między kolejnymi miastami na drodze będącej rozwiązaniem problemu wraz z przejściem łączącym ostatnie odwiedzone (unikalne) miasto z miastem startowym. Rozwiązanie, dla którego funkcja ta przyjmuje wartość minimalną nazywa się rozwiązaniem optymalnym.

W zależności od tego, czy przejście z miasta A do miasta B ma taki sam koszt, co przejście z miasta B do miasta A dla dowolnej pary miast (miasta muszą być różne od siebie) mamy do czynienia z symetrycznym problemem komiwojażera, jeżeli koszty te są równe, zaś w przeciwnym przypadku mówi się o asymetrycznym problemie komiwojażera. Tematem projektu jest asymetryczny problem komiwojażera. Warto dodać, iż każdy symetryczny problem komiwojażera można potraktować tak, jak asymetryczny, uprzednio dublując koszty przejścia pomiędzy każdą parą miast, tak aby przejście było możliwe w obie strony.

2 Metody i algorytmy rozwiązywania problemu

2.1 Symulowane wyżarzanie

Symulowane wyżarzanie jest metaheurystyką z kategorii metod poszukiwania lokalnego. Jest to metoda iteracyjna. Polega na zdefiniowanym poruszaniu się po przestrzeni rozwiązań mając do dyspozycji ruchy z sąsiedztwa, w nadziei na znalezienie rozwiązania globalnie optymalnego. Element sąsiedztwa jest dobierany losowo i oceniany. Następnie dokonuje się decyzji binarnej o wykonaniu ruchu do wyznaczonego punktu przestrzeni lub jego niewykonaniu. Prawdopodobieństwo wykonania ruchu jest tym większe, im większy jest przyrost wartości kryterium dla nowo wyznaczonego punktu przestrzeni rozwiązań, względem aktualnie posiadanego rozwiązania. Zachowanie to nosi cechy eksploatacji, gdyż podczas tego kroku następuje próba przejścia w kierunku ekstremum lokalnego. Dodatkowo dopuszcza się wykonanie ruchu do punktu z mniejszą od aktualnego rozwiązania wartością kryterium. Prawdopodobieństwo wykonania takiego ruchu jest tym mniejsze im mniejsza jest wartość kryterium dla nowego punktu. Takie zachowanie pozwala na eksplorację przestrzeni. Przeszukiwane są różne jej fragmenty po to, by znaleźć rozwiązanie globalnie optymalne. Bez tego zabiegu metoda ta znajdowałaby przeważnie rozwiązania lokalnie optymalne.

Esencją symulowanego wyżarzania jest bilans pomiędzy eksploracją i eksploatacją. Jest on utrzymywany za pomocą wyżej opisanych sposobów poruszania się po przestrzeni. Na prawdopodobieństwo podejmowanej decyzji wpływa dodatkowy parametr zwany temperaturą. Wpływ ten jest taki, iż przy większych wartościach parametru pojawia się tendencja do wykonywania ruchów eksploracyjnych, zaś przy mniejszych wartościach liczba ruchów eksploracyjnych maleje na rzecz ruchów eksploatacyjnych. Zazwyczaj temperatura jest na początku względnie dużą liczbą dodatnią, która maleje wraz upływem czasu. Symuluje to proces przechodzenia systemu ze stanu wzbudzonego do stanu równowagi. Przypomina to wyżarzanie materiału; od procesu tego pochodzi nazwa metody.

Zaimplementowany algorytm działający zgodnie z tą metodą miał kilka dodatkowych elementów. Umożliwiono wykonywanie ruchów w przestrzeni dla stałej temperatury. Wzbogacono algorytm o możliwość zapisywania najlepszego dotychczasowego rozwiązania. Zdefiniowano 3 typy sąsiedztwa: swap, insert, invert oraz zaimplementowano szybsze liczenie wartości funkcji celu dla nowego punktu przestrzeni. Akceleracja obliczeń była wykonywana na podstawie typu sąsiedztwa oraz aktualnego rozwiązania; modyfikowano wartości jedynie tych

krawędzi, które różniły rozważane rozwiązania. Zmodyfikowano sposób podejmowania decyzji o wyborze rozwiązania z sąsiedztwa jako kolejne tak, że w przypadku decyzji dotyczącej lepszego rozwiązania jest ono wybierane z prawdopodobieństwem równym 1. Funkcję wykorzystaną do obliczania prawdopodobieństwa wyboru rozwiązania gorszego zdefiniowano jako

$$P(s_c, s_n, T) = \frac{2}{1 + e^{-(s_c - s_n)/T}},$$

gdzie s_c oznacza wartość funkcji celu rozwiązania aktualnego, s_n oznacza wartość funkcji celu wyznaczonego elementu sąsiedztwa, zaś T oznacza obecną temperaturę. Zdefiniowano 3 typy rozwiązania początkowego, tj. permutacja naturalna, rozwiązanie losowe, rozwiązanie po algorytmie zachłannym oraz 3 schematy chłodzenia (liniowy, geometryczny, logarytmiczny) modyfikowane jednym parametrem. Wprowadzanymi parametrami algorytmu były:

- liczba iteracji,
- liczba iteracji dla stałej temperatury,
- temperatura początkowa,
- schemat chłodzenia,
- parametr schematu chłodzenia,
- typ rozwiązania początkowego,
- typ sąsiedztwa.

2.2 Poszukiwanie z zakazami

Poszukiwanie z zakazami jest metaheurystyką z kategorii metod poszukiwania lokalnego. Jest to metoda iteracyjna. Jej idea polega na poruszaniu się po przestrzeni rozwiązań z nałożonymi w czasie szukania rozwiązania ograniczeniami. Kolejny ruch z aktualnego punktu przestrzeni następuje po zbadaniu wszystkich elementów sąsiedztwa. Z pośród nich wybierany jest element „najlepszy”, na który nie zostały nałożone ograniczenia (ruch nie został zakazany). Możliwy jest wybór rozwiązania zabronionego, które spełnia zadane warunki, tak zwane kryterium aspiracji. Po wyborze kolejnego punktu przestrzeni następuje ruch (pomijając przypadek, w którym żaden element sąsiedztwa nie jest elementem dozwolonym; wtedy ruch nie następuje), po którym atrybut identyfikujący wykonane przemieszczenie jest zapisywany na liście ograniczeń. Ruchy o atrybutach z listy ograniczeń są ruchami zabronionymi. Swój status utrzymują przez czas zwany kadencją; po jego upływie ruch na powrót staje się dozwolony.

Cechy eksploatacji można zauważyć podczas wyboru ruchu „najlepszego” z aktualnego sąsiedztwa. Dzięki temu, że ruch ten jest „najlepszy” kolejne rozwiązania będą zbliżały się do ekstremum lokalnego. Aby zwiększyć szanse na znalezienie rozwiązania globalnie optymalnego metody poszukiwania lokalnego muszą posiadać cechy eksploracji. W poszukiwaniu z zakazami efekt ten uzyskuje się dzięki ustanawianiu ograniczeń; sprawia się, że ruchy do siebie podobne w sensie ich atrybutów nie są wykonywane natychmiast po sobie; wymuszane jest badanie większej części przestrzeni rozwiązań. Cech eksploracji metoda ta nabiera również ze względu na fakt, iż zawsze wybierane jest rozwiązanie z sąsiedztwa (o ile jest dozwolone), bez względu na to czy jest ono lepsze, czy gorsze od rozwiązania aktualnego.

Zaimplementowany algorytm działał zgodnie z wyżej opisaną metodą, przy czym wprowadzono kilka elementów dodatkowych. Pierwszym z nich była zdolność algorytmu do wylosowania nowego punktu przestrzeni jako rozwiązania aktualnego w przypadku, gdy poprawa najlepszego dotąd rozwiązania nie zachodziła przez określoną liczbę iteracji. Dodano strukturę danych przechowującą kilka rozwiązań, do których ruch był niedawno wykonany. Podczas badania sąsiedztwa brano tę strukturę pod uwagę i zabraniano ruchu do elementów, które był przechowywane w tej strukturze. Tak jak w przypadku symulowanego wyżarzania przechowywano osobno najlepsze znalezione dotychczas rozwiązanie. Algorytm mógł pracować z sąsiedztwami swap, insert, invert. Wartości funkcji celu dla elementów sąsiedztwa była wyznaczana sposobem przyspieszonym polegającym na analizie zależności

nowego rozwiązania od poprzedniego tak, jak miało to miejsce w algorytmie symulowanego wyżarzania. Kadencja ograniczenia była wprost proporcjonalnie zależna od wielkości instancji oraz zadawanego stałego parametru. Rozwiązanie początkowe było wyznaczane jako permutacja naturalna, rozwiązanie losowe lub rozwiązanie po algorytmie zachłannym. Kryterium aspiracji odpowiadało na pytanie, czy przetwarzany element sąsiedztwa ma mniejszą wartość funkcji celu od elementu najlepszego dotychczas znalezionego. Napisano dwie wersje algorytmu, które różniły się strukturą tabu; była to lista wiązana oraz macierz. Obie przechowywał atrybut ruchu (indeksy używane do wyznaczenia elementu sąsiedniego) oraz długość kadencji. W przypadku listy był to 3-wartościowy obiekt, zaś w przypadku macierzy atrybut zawierał indeksy do pola macierzy, w którym przechowywano aktualną długość kadencji dla danego atrybutu.

Wprowadzanymi parametrami algorytmu były:

- liczba iteracji,
- długość listy tabu,
- parametr kadencji,
- liczba iteracji do metody zdarzeń krytycznych,
- liczba przechowywanych w pamięci ostatnio wyznaczonych rozwiązań,
- typ rozwiązania początkowego,
- typ sąsiedztwa.

3 Eksperymenty obliczeniowe

W ramach projektu przeprowadzono badania wpływu parametrów na jakość rozwiązań dostarczanych przez zaimplementowane algorytmy. Ponadto przeprowadzono eksperymenty obliczeniowe, które polegały na pomiarze czasu wykonania oraz jakości zwracanych rozwiązań dostrojonego algorytmu w zależności od wielkości instancji problemu. Czas był mierzony w milisekundach, zaś do samego pomiaru czasu wykorzystano pakiet z biblioteki standardowej C++ - *std::chrono*. Instancje były podawane z pliku - były to wybrane instancje testowe dostarczone razem z zadaniem projektowym. Komputer, na którym przeprowadzono testy posiadał procesor Intel Core i7-6700HQ oraz był wyposażony w 16 GB RAMu.

Podczas badań wpływu parametrów na jakość rozwiązań ustawiano domyślne parametry o wartościach z tabel 1 i 2 dla odpowiednich algorytmów. Wykorzystane w tym badaniu parametry są zawarte w 2 kolumnie tych tabel. Oczywiście badany parametr był zmieniany. Podczas badania czasu i jakości dostarczanych przez algorytmy rozwiązań wykorzystano wyniki badań parametrów i empirycznie wyznaczono optymalne parametry pracy zaimplementowanych algorytmów. Są one zawarte w 3 kolumnie tabel 1 i 2.

Parametr	Badanie parametrów	Badanie czasu i jakości rozwiązań
Temperatura początkowa	1000	2000
Liczba iteracji	1000	500
Liczba iteracji dla stałej temp.	500	1000
Parametr schematu chłodzenia	0.95	15
Schemat chłodzenia	geometryczny	logarytmiczny
Rodzaj sąsiedztwa	swap	insert
Rozwiązanie początkowe	permutacja naturalna	po algorytmie zachłannym

Tablica 1: Parametry użyte podczas eksperymentów obliczeniowych dla algorytmu symulowanego wyżarzania

Parametr	Badanie parametrów	Badanie czasu i jakości rozwiązań
Długość listy tabu	50	20
Liczba iteracji	100	200
Parametr kadencji	0.125	0.5
Liczba iteracji bez poprawy do restartu	50	200
Liczba przechowywanych tymczasowo rozwiązań	2	5
Rodzaj sąsiedztwa	swap	insert
Rozwiązanie początkowe	permutacja naturalna	po algorytmie zachłannym

Tablica 2: Parametry użyte podczas eksperymentów obliczeniowych dla algorytmu poszukiwania z zakazami

3.1 Analiza wpływu parametrów algorytmu na jakość rozwiązań

Przedstawione w tej sekcji wykresy reprezentują wyniki badań wpływu parametrów na jakość dostarczanych przez algorytmy rozwiązań. Każdy wykres przedstawia dane 4-wymiarowe, gdzie:

- oś X przedstawia wartość badanego parametru,
- oś Y przedstawia błąd średni względny, liczony jako

$$E_{mr} = \frac{S_m - S_{opt}}{S_{opt}} * 100\%,$$

gdzie S_m oznacza średnią wartość funkcji celu rozwiązania z powtórzonych obliczeń dla tej samej instancji i wartości parametru, zaś S_{opt} oznacza wartość funkcji celu rozwiązania optymalnego,

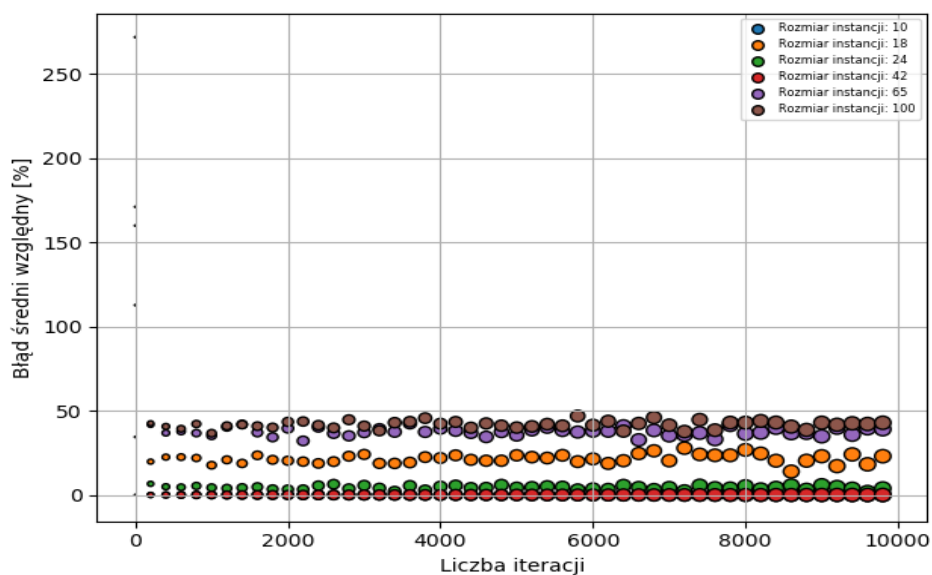
- kolor punktu przedstawia rozmiar badanej instancji,
- rozmiar punktu obrazuje średni czas trwania badania z powtórzonych obliczeń dla tej samej instancji i wartości parametru.

Przy wyznaczaniu parametrów optymalnej pracy algorytmów kierowano się kryteriami:

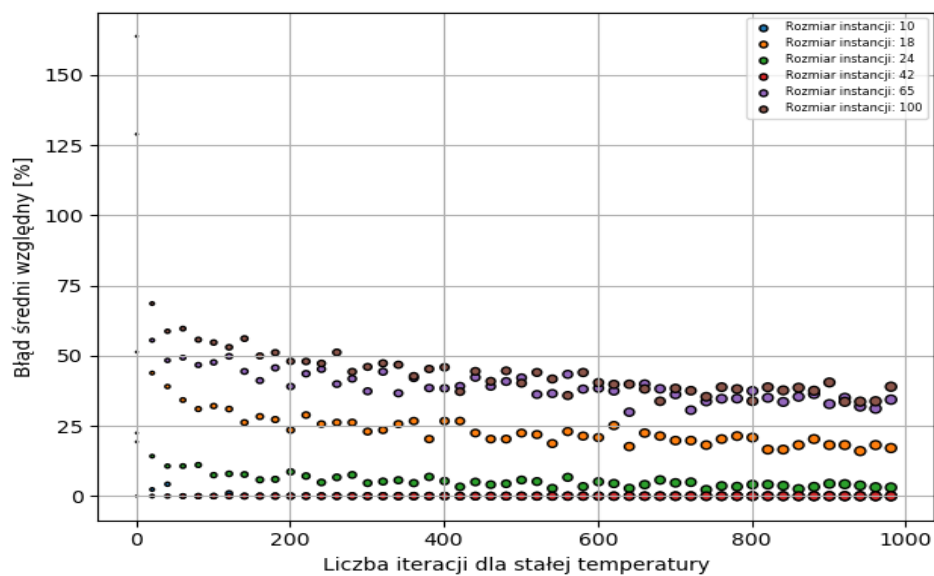
1. możliwie małego błędu średniego względnego,
2. możliwie dużego zagęszczenia punktów pomiarowych,
3. możliwie krótkiego czasu badania.

Kryterium o mniejszym numerze porządkowym było istotniejsze. W przypadkach niejednoznacznych istotność kryterium nie była brana pod uwagę. Wyznaczone empirycznie optymalne parametry pracy algorytmów są zawarte w 3. kolumnie tabel 1 oraz 2.

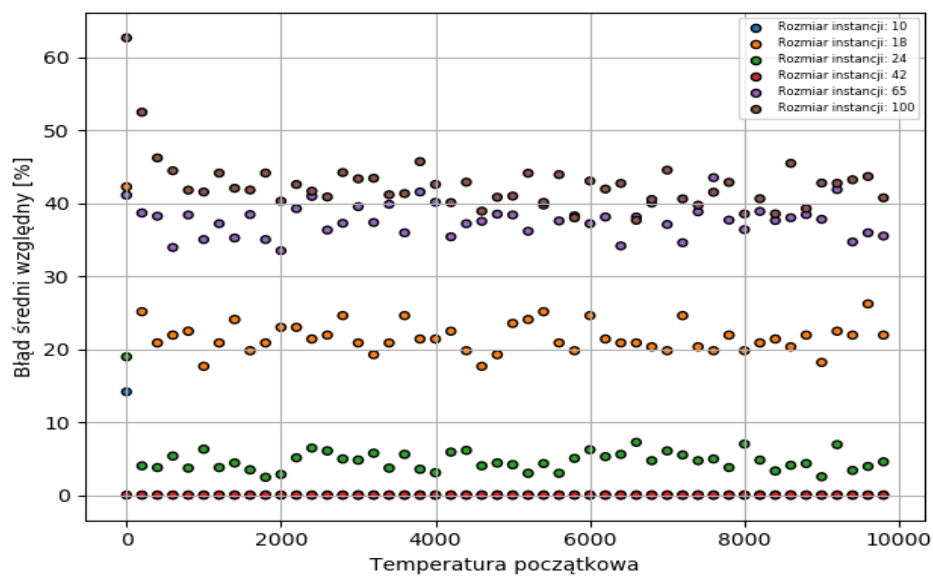
3.1.1 Symulowane wyżarzanie



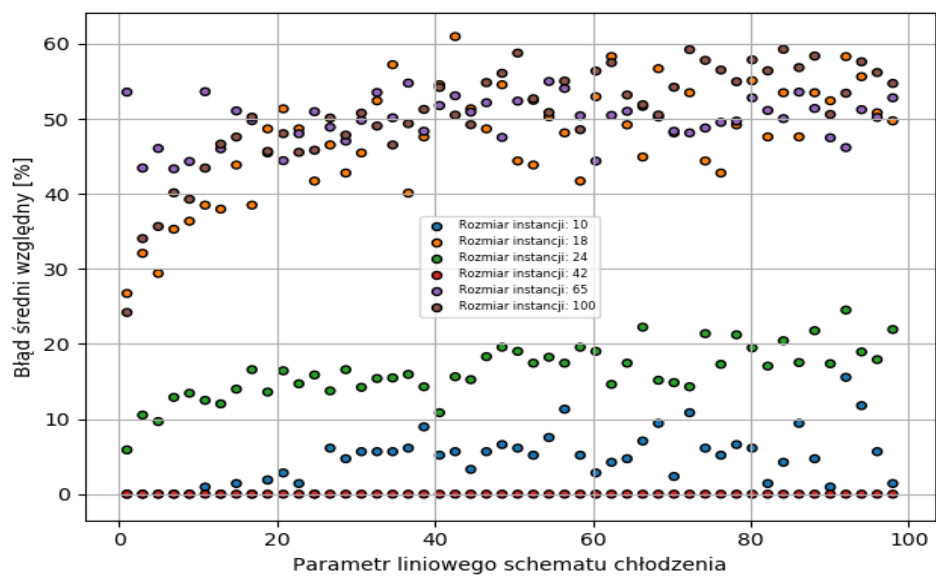
Rysunek 1: Symulowanie wyżarzanie - liczba iteracji



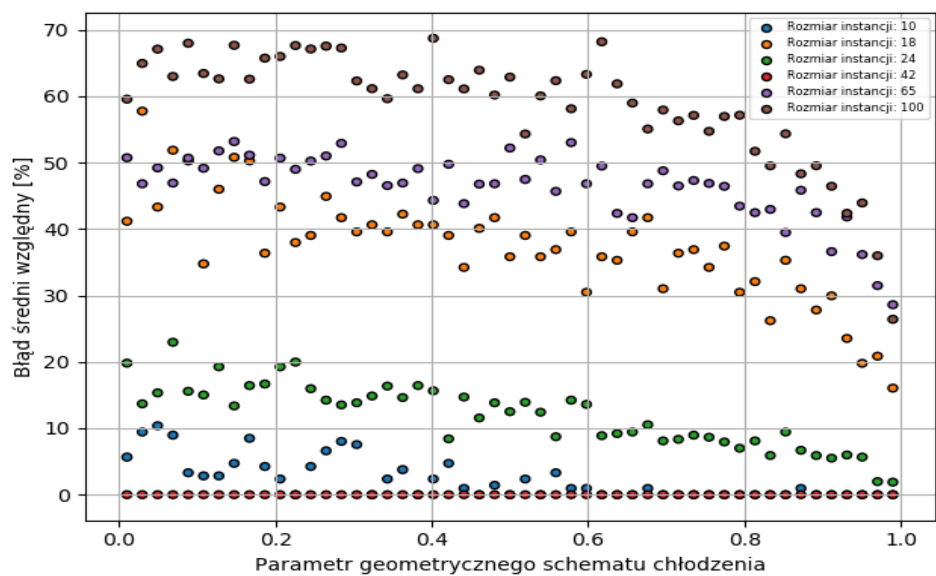
Rysunek 2: Symulowanie wyżarzanie - liczba iteracji dla stałej temperatury



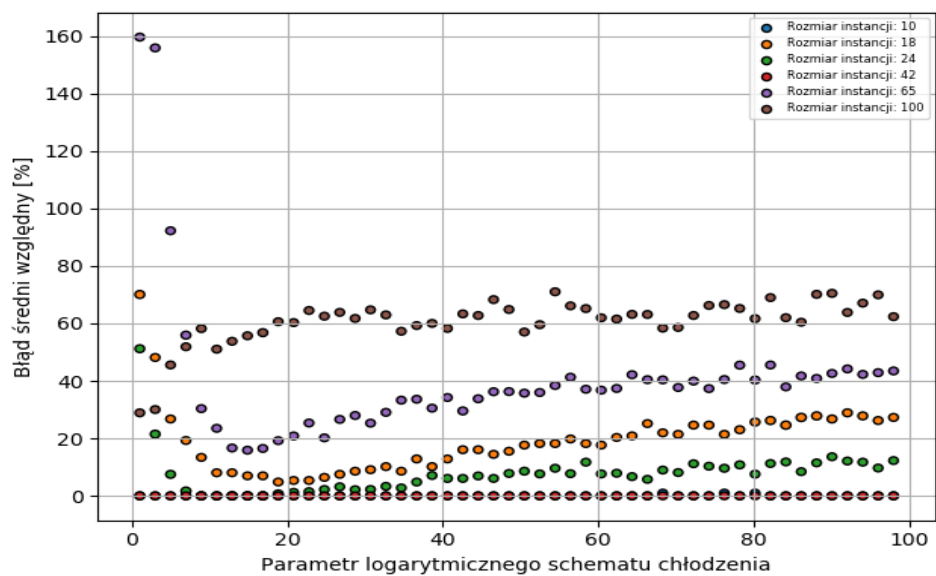
Rysunek 3: Symulowanie wyżarzanie - temperatura początkowa



Rysunek 4: Symulowanie wyżarzanie - parametr liniowego schematu chłodzenia

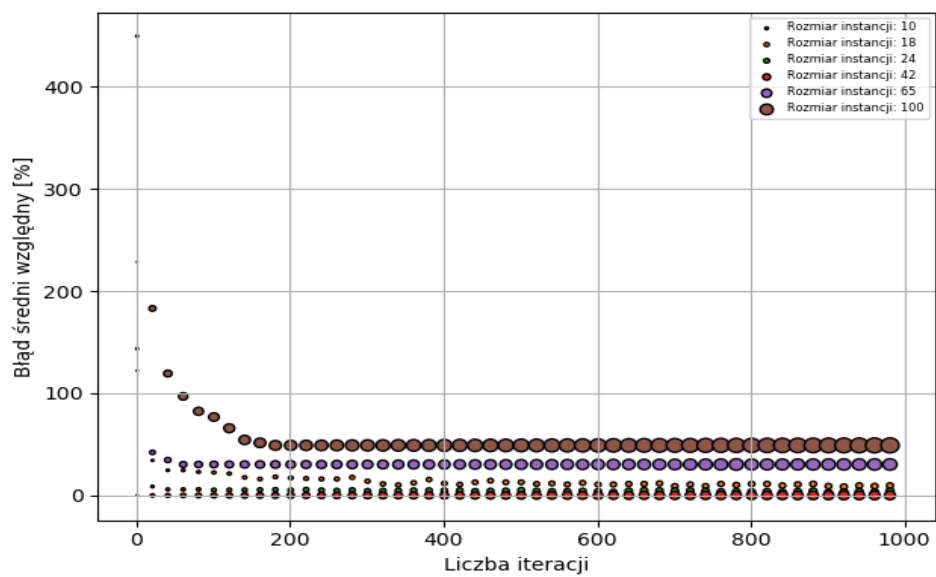


Rysunek 5: Symulowanie wyżarzanie - parametr geometrycznego schematu chłodzenia

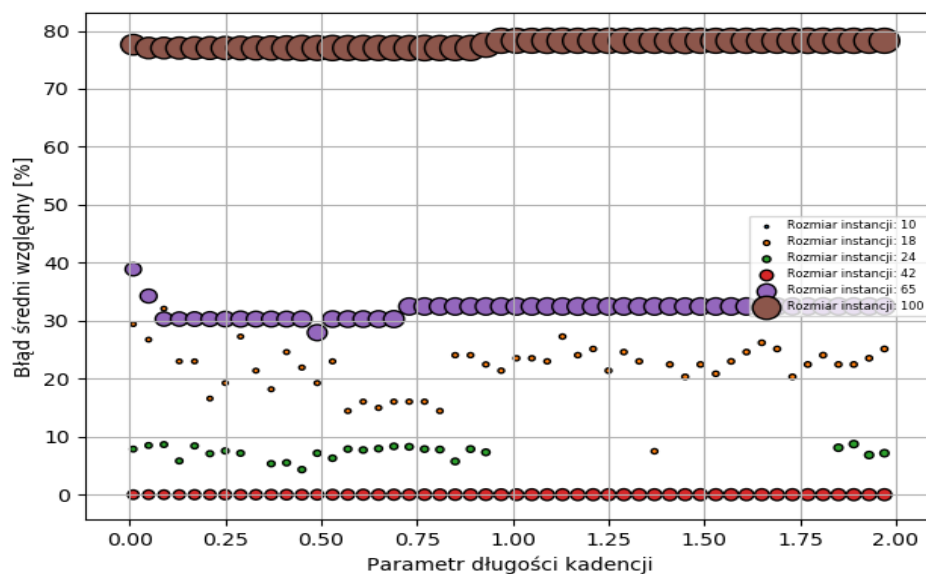


Rysunek 6: Symulowanie wyżarzanie - parametr logarytmicznego schematu chłodzenia

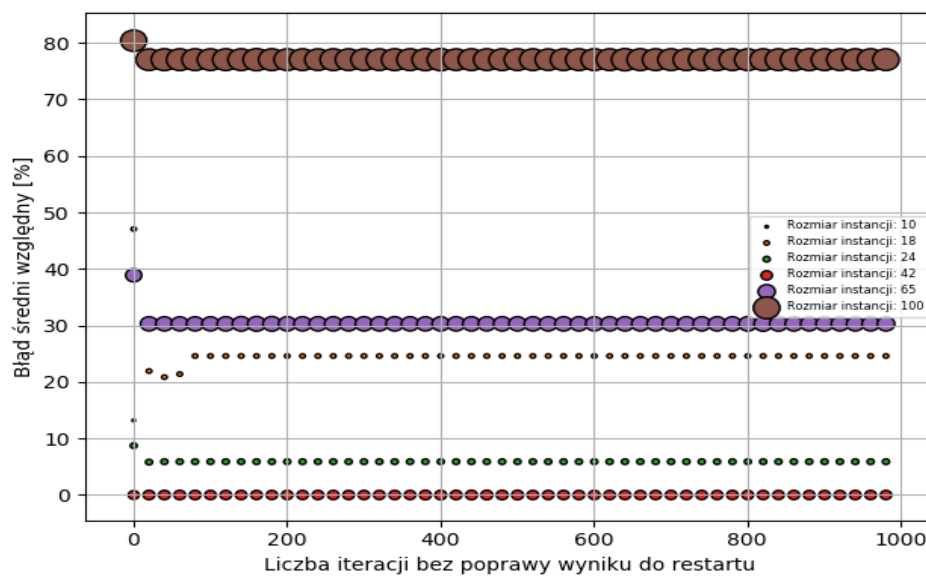
3.1.2 Poszukiwanie z zakazami



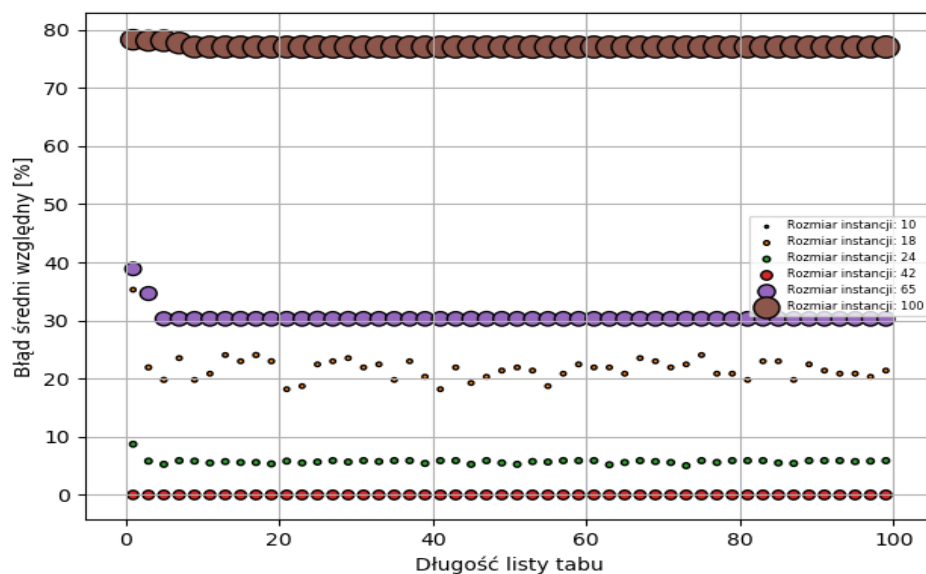
Rysunek 7: Poszukiwanie z zakazami - liczba iteracji



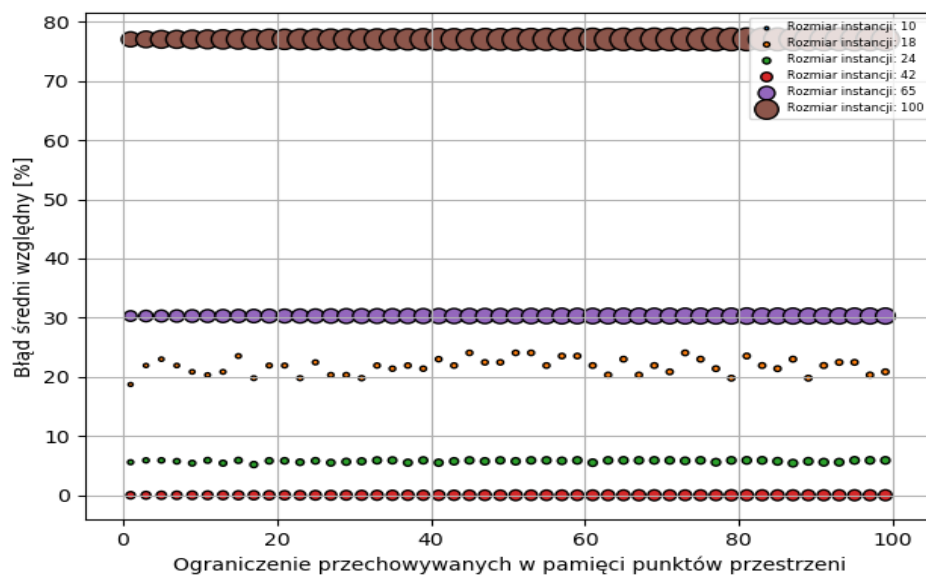
Rysunek 8: Poszukiwanie z zakazami - parametr długości kadencji



Rysunek 9: Poszukiwanie z zakazami - liczba iteracji bez poprawy wyniku do ponownego wylosowania punktu przestrzeni



Rysunek 10: Poszukiwanie z zakazami - długość listy tabu



Rysunek 11: Poszukiwanie z zakazami - ograniczenie na liczbę przechowywanych w pamięci odwiedzonych punktów przestrzeni

3.2 Testy czasowe i jakościowe

Na wstępie należy zaznaczyć, iż wszystkie pomiary w tej sekcji zostały wykonane dla stałych parametrów i powtórzone zostały 30 razy.

3.2.1 Tabela - zestawienie danych pomiarowych

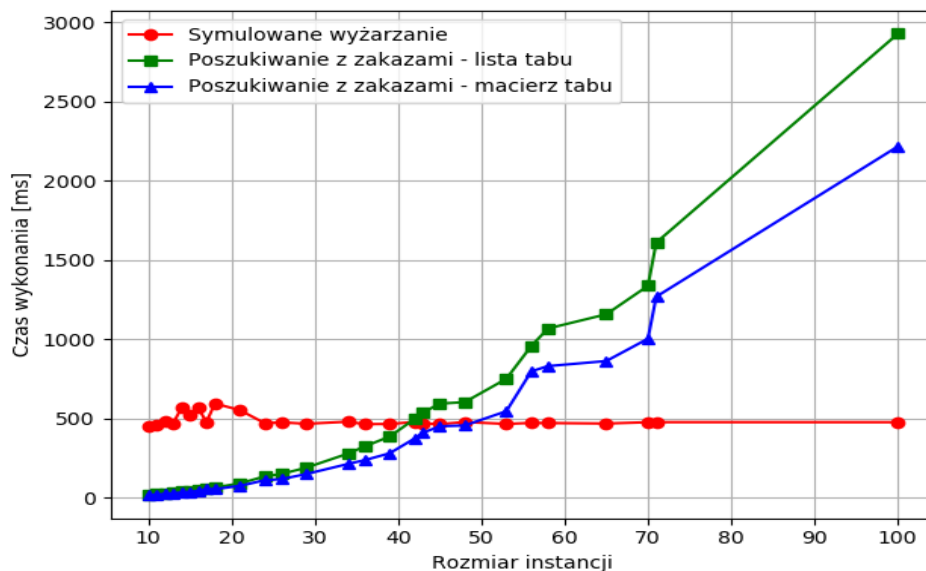
Rozmiar instancji	SA: czas [ms]	TSL: czas [ms]	TSM: czas [ms]	SA: błąd średni względny [%]	TSL: błąd średni względny [%]	TSM: błąd średni względny [%]
10	454.13	18.50	15.38	0.00	3.30	3.30
11	458.76	22.56	18.90	0.00	13.37	13.37
12	481.56	28.26	22.96	0.00	0.00	0.00
13	468.33	31.50	27.62	2.60	0.00	0.00
14	566.75	40.37	31.57	4.80	35.20	35.20
15	525.18	42.34	36.86	9.97	0.00	0.00
16	568.85	49.17	41.94	4.49	33.97	33.97
17	476.04	60.19	54.14	5.13	15.38	15.38
18	594.31	64.05	53.71	18.72	0.00	0.00
21	552.85	89.94	74.19	0.00	10.71	10.71
24	470.04	133.85	109.82	1.57	7.39	7.39
26	475.32	149.35	116.79	8.11	2.35	2.35
29	466.50	189.31	149.36	4.22	8.76	8.76
34	479.41	278.26	212.84	16.41	10.50	10.50
36	463.99	323.48	236.74	14.66	6.52	6.52
39	465.25	385.67	280.14	19.15	8.10	8.10
42	477.56	497.06	373.79	41.63	12.16	12.16
43	469.81	537.09	409.57	3.06	0.37	0.37
45	464.10	593.79	450.52	18.47	14.07	14.07
48	477.20	602.66	455.79	2.93	17.04	17.04
53	465.60	750.19	545.50	4.65	22.40	22.40
56	471.73	957.68	797.50	22.57	12.81	12.81
58	470.98	1067.83	830.45	5.29	6.62	6.62
65	467.99	1157.58	862.42	26.32	6.63	6.63
70	476.03	1337.06	1003.37	3.68	7.41	7.41
71	476.58	1610.25	1268.66	28.21	9.95	9.95
100	476.30	2928.58	2217.11	18.27	10.79	10.79

Tablica 3: Zestawienie wyników testów czasowych i jakościowych

Legenda:

- SA - symulowane wyżarzanie,
- TSL - poszukiwanie z zakazami (lista tabu),
- TSM - poszukiwanie z zakazami (macierz tabu).

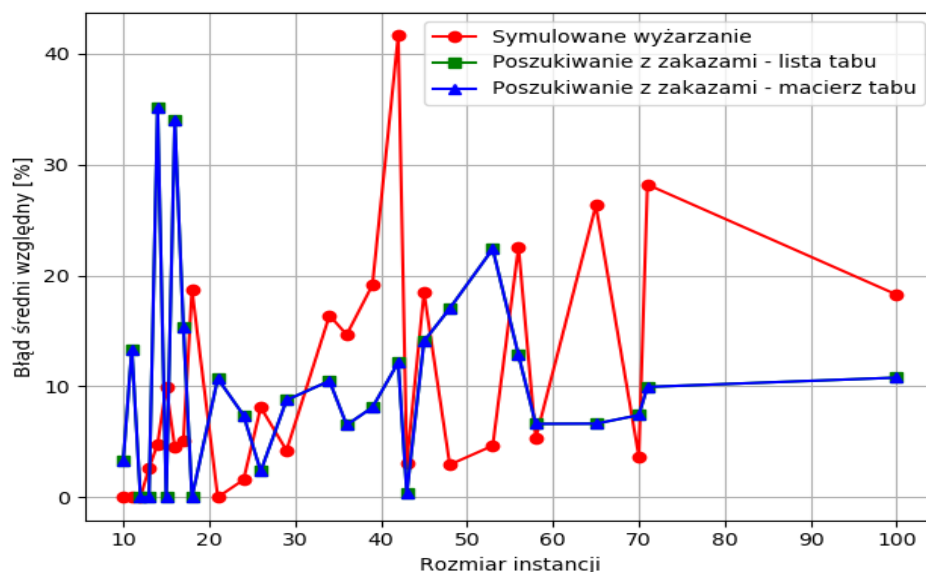
3.2.2 Analiza wyników przeprowadzonych testów



Rysunek 12: Wydajność zaimplementowanych algorytmów

Symulowane wyżarzanie eksponuje złożoność obliczeniową $O(1)$. Jest to zasługa akceleracji obliczeń wartości funkcji celu dla elementu sąsiedniego. Bez akceleracji, złożoność ta byłaby zależna od rozmiaru instancji i wynosiłaby $O(n)$. Nie jest zaskakujące też, iż poszukiwanie z zakazami wykazuje złożoność $O(n^2)$, gdyż w każdej iteracji przeglądane jest całe sąsiedztwo obecnego rozwiązania o wielkości $O(n^2)$.

Implementacja macierzowa poszukiwania z zakazami uwidacznia się jako szybsza, ze względu na redukcję złożoności operacji sprawdzania struktury tabu wykonywanej w pętli sąsiedztwa. Dostęp do informacji o posiadaniu danego elementu jako zakazanego jest w przypadku macierzy natychmiastowy, zaś dla listy operacja ta ma złożoność $O(n)$. Jest to uzyskiwane kosztem dłuższej aktualizacji macierzy tabu ($O(n^2)$), natomiast dla listy operacja ta ma złożoność $O(n)$.



Rysunek 13: Jakość rozwiązań dostarczanych przez zaimplementowane algorytmy

Z wykresu 13 można wywnioskować, że znaczna większość instancji problemu została rozwiązana z błędem mniejszym, niż 30% przez każdy z algorytmów, co biorąc pod uwagę ich niedługi czas pracy wydaje się być wynikiem bardzo dobrym. Ciekawym wydaje się być fakt, iż w wykonanych badaniach można zauważyć wzajemnie „dopełnianie się” algorytmów; podczas kiedy jeden z nich zwraca rozwiązanie z ustalonym błędem, drugi może poprawić rozwiązanie pierwszego. Jak wynika z tabeli 3 wybranie lepszego z obu rozwiązań algorytmów daje w rezultacie maksymalny błąd równy 14.07% (dla instancji o rozmiarze 45). Niepokoi natomiast fakt, że względnie wysokie błędy pojawiają się przy rozwiązywaniu małych instancji. Z racji, iż algorytmy te w zbadanej konfiguracji nie działał jednakowo długo dla każdej instancji, nie można jednoznacznie stwierdzić, który z algorytmów okazał się być lepszy pod kątem jakości zwracanych rozwiązań w wykonanych badaniach.

4 Wnioski

Algorytmy poszukiwania lokalnego, chociaż nie gwarantują znalezienia rozwiązania optymalnego w ogóle, to jednak jak wynika z przeprowadzonych badań, okazują się być obiecującą metodą do rozwiązywania problemu komiwojażera. Warunkiem ich akceptowalnego działania jest uwzględnienie oraz odpowiednie wyważenie zawartej w nich eksploracji oraz eksploatacji przestrzeni rozwiązań. Zaletą tych algorytmów jest również szybka zbieżność do rozwiązania optymalnego (choć nie zawsze globalnie optymalnego).

Problematyczne okazuje się być dokładne wyznaczenie optymalnych parametrów pracy tych algorytmów z uwagi na często niewyraźne reakcje uwidaczniające się w zmianie czasu pracy lub/i jakości dostarczanych rozwiązań. Aby takie badania były rzetelne, po każdorazowym zauważeniu poprawy działania algorytmu dla wybranego przedziału wartości parametru, należałoby zwiększyć rozdzielczość badań i przeprowadzić pomiary ponownie.

Algorytmy te mogłyby być zastosowane w sytuacjach, w których dokładność rozwiązania nie jest tak ważna, jak czas jego znalezienia. Ich użycie byłoby możliwe w systemach czasu rzeczywistego, w których rozwiązanie pojawiałoby się praktycznie natychmiastowo, a jego dokładność rosłaby w czasie i nie przeszkadzałoby to działaniu systemu. System ten wykorzystywałby uzyskane informacje o rozwiązaniu do natychmiastowej poprawy swojego działania; można wyobrazić sobie sytuację, w której przyniosłoby to wymierne korzyści.