

Programowanie Efektywnych Algorytmów

Sprawozdanie z zadania projektowego numer 2 - *Implementacja i analiza efektywności algorytmu Przeszukiwania z Ograniczeniami (ang. Tabu Search) oraz Symulowanego Wyżarzania (ang. Simulated Annealing) dla asymetrycznego problemu komiwojażera (ATSP).*

Autor: Krzysztof Krajewski 272877

Wprowadzenie

Problem komiwojażera w wersji asymetrycznej (ATSP, ang. Asymmetric Traveling Salesman Problem) jest wariantem klasycznego problemu komiwojażera (TSP, ang. Traveling Salesman Problem), w którym koszty przejścia między dwoma miastami nie są symetryczne, tzn. koszt przejścia z miasta A do miasta B może różnić się od kosztu przejścia z miasta B do miasta A. Formalnie, problem można opisać za pomocą skierowanego grafu pełnego $G = (V, E)$, gdzie V to zbiór wierzchołków (miast), a E to zbiór skierowanych krawędzi o przypisanych kosztach. Celem jest znalezienie cyklu Hamiltona, który minimalizuje łączny koszt podróży i odwiedza każde miasto dokładnie raz.

Złożoność obliczeniowa problemu ATSP

ATSP jest problemem NP-trudnym, co oznacza, że nie jest znany algorytm, który mógłby rozwiązać go w czasie wielomianowym dla dowolnej liczby miast. Rozmiar przestrzeni rozwiązań rośnie wykładniczo wraz z liczbą miast N , ponieważ liczba możliwych permutacji wynosi $(n-1)!$. Z tego względu optymalne rozwiązania dla ATSP są trudne do uzyskania dla dużych instancji problemu. W tym sprawozdaniu przeprowadzimy analizę następujących podejść do rozwiązywania dużych instancji tego problemu:

- Przeszukiwanie z Ograniczeniami (ang. Tabu Search)
- Symulowanego Wyżarzania (ang. Simulated Annealing)

Omówienie algorytmów

Wszystkie przedstawione algorytmy działają z użyciem klasy CityMatrix, która zawiera dwuwymiarową tablicę przechowującą macierz odległości pomiędzy miastami wraz z metodami obsługującymi jej inicjalizację, generację i wczytywanie.

Przeszukiwanie z Ograniczeniami (ang. Tabu Search)

Algorytm Tabu Search zaimplementowany w metodzie TSP::tabuSearch rozwiązuje problem ATSP, zaczynając od rozwiązania początkowego wygenerowanego za pomocą algorytmu zachłannego. Następnie algorytm iteracyjnie poprawia bieżące rozwiązanie, eksplorując sąsiedztwo (wygenerowane jako kombinacja wszystkich możliwych zamian dwóch wierzchołków w danym rozwiązaniu) oraz poprzez losowe modyfikacje trasy (sprawiając, że algorytm przestaje być deterministyczny).

Algorytm wykorzystuje listę tabu do ograniczania powrotu do wcześniej odwiedzanych rozwiązań, co zapobiega lokalnym cyklom. Lista jest aktualizowana po wykonaniu ruchu, przypisując

wartość TABU_TENURE do odpowiedniego pola w macierzy jednocześnie zmniejszając wartość TABU_TENURE dla wszystkich pozostałych, niezerowych wartości.

Dodatkowo, w ramach dywersyfikacji przeszukiwania została wprowadzona wartość kary. Jest ona przechowywana w dolnej połowie macierzy tabu. Co jeden wybrany ruch, w miejsce odpowiadające mu w macierzy kar zostaje wpisana wartość powiększona o pewną stałą wartość w porównaniu do poprzedniej. W ten sposób algorytm jest zniechęcany do tworzenia cykli i wykonywania ostatnio wykonanych ruchów.

Podczas każdej iteracji ruchy są oceniane pod kątem wpływu na koszt całkowity trasy (uwzględniając kary za częstość wyboru ruchu), a najlepszy z nich jest wybierany jako kandydat na nowe rozwiązanie. Jeśli ruch jest tabu (oznaczony na liście tabu), może zostać wybrany jedynie wtedy, gdy prowadzi do poprawy najlepszego dotychczasowego rozwiązania. Jeśli nie znaleziono dopuszczalnego ruchu (wszystkie są tabu), algorytm wybiera najmniej kosztowny ruch z dostępnych, ignorując ograniczenie tabu.

Kod 1. Pseudokod algorytmu Tabu Search.

Rozpocznij algorytm TabuSearch:

Znajdź początkową trasę za pomocą algorytmu zachłannego i ustaw ją jako aktualną trasę oraz najlepszą dotychczasową trasę.

Utwórz listę tabu, w której wszystkie ruchy są początkowo niezabronione.

Dopóki aktualny czas nie przekroczy czasu zakończenia:

Wygeneruj sąsiedztwo bieżącej trasy.

Wygeneruj losowe trasy.

Dla każdej wygenerowanej trasy:

Oblicz koszt trasy biorąc pod uwagę listę kar.

Dodaj ruch do listy potencjalnych ruchów.

Wybierz najlepszy ruch z listy potencjalnych ruchów:

Jeśli ruch jest dozwolony:

Ustaw ruch jako bieżący.

W przeciwnym wypadku, jeśli ruch poprawia bieżącą trasę.

Ustaw ruch jako bieżący

Jeśli wszystkie ruchy są zabronione, wybierz ruch o najmniejszym koszcie, ignorując tabu.

Wykonaj wybrany ruch, aktualizując bieżącą trasę.

Jeśli nowa trasa ma mniejszy koszt niż najlepsza dotychczasowa:

Ustaw nową trasę jako najlepszą dotychczasową trasę.

Zaktualizuj listę tabu, oznaczając wykonany ruch jako zabroniony na określony czas.

Zaktualizuj listę kar.

Po zakończeniu zwróć najlepszą znaną trasę i jej koszt.

Symulowane Wyżarzanie (ang. Simulated Annealing)

Algorytm Simulated Annealing (SA) zaimplementowany w metodzie TSP::simulatedAnnealing rozwiązuje problem ATSP, wykorzystując proces chłodzenia, który jest inspirowany fizycznym procesem wykorzystywanym w metalurgii. Algorytm zaczyna od obliczenia początkowej temperatury na podstawie różnicy kosztów tras z zebranych próbek. Wzór na wartość temperatury początkowej zaczerpnięty został z rozkładu Boltzmana i ma postać: $-\text{avg_cost_dif} / \log(0.8) * \text{STARTING_TEMPERATURE_MULTIPLIER}$. Po przekształceniu: Początkowa trasa generowana jest za pomocą algorytmu zachłannego i traktowana jako rozwiązanie wyjściowe.

W każdej iteracji algorytmu generowany jest losowy ruch, który zmienia trasę. Nowa trasa jest oceniana pod kątem kosztu: jeśli jest lepsza, jest natychmiast akceptowana. Jeśli jest gorsza, akceptowana jest z pewnym prawdopodobieństwem, które zależy od temperatury – im wyższa temperatura, tym większa szansa na zaakceptowanie gorszego rozwiązania (jest to etap, w którym algorytm eksploruje przestrzeń rozwiązań). To umożliwia algorytmowi unikanie utknięcia w lokalnych minimach. Po wykonaniu określonej liczby prób w danej iteracji, temperatura jest obniżana zgodnie z parametrem chłodzenia, co sprawia, że w kolejnych etapach algorytm staje się bardziej ukierunkowany na poprawę rozwiązania (to jest etap, w którym algorytm eksploatuje obszar, w którym się znalazł).

Jeśli po danej iteracji uzyskana trasa ma lepszy koszt niż dotychczasowe najlepsze rozwiązanie, zostaje ono zaktualizowane. Algorytm kończy działanie po upływie określonego czasu, zwracając najlepszą znalezioną trasę oraz jej koszt.

Kod 2. Pseudokod algorytmu Simulated Annealing.

Rozpocznij algorytm SimulatedAnnealing:

Znajdź początkową trasę za pomocą algorytmu zachłannego i ustaw ją jako aktualną trasę oraz najlepszą dotychczasową trasę.

Oblicz początkową temperaturę T na podstawie różnicy kosztów z próbek.

Dopóki aktualny czas nie przekroczy czasu zakończenia:

Dla $i < K_T$:

Wygeneruj losowy ruch w sąsiedztwie bieżącej trasy.

Jeśli koszt nowej trasy jest mniejszy niż koszt bieżącej trasy:

Ustaw nową trasę jako bieżącą.

W przeciwnym wypadku:

Oblicz prawdopodobieństwo nowej trasy $\left(P = \frac{1}{e^{\frac{\text{new_tour_cost} - \text{best_tour_cost}}{T}}} \right)$.

Zaakceptuj nową trasę z obliczonym prawdopodobieństwem.

$i++$.

Jeśli nowa trasa ma mniejszy koszt niż najlepsza dotychczasowa:

Zaktualizuj najlepszą dotychczasową trasę.

Zaktualizuj temperaturę T na podstawie schematu schładzania.

Po zakończeniu zwróć najlepszą znalezioną trasę i jej koszt.

Opis najważniejszych klas w projekcie

W projekcie znajdują się trzy główne klasy, które pełnią kluczowe funkcje w rozwiązaniu problemu komiwojażera.

Klasa `cityMatrix` odpowiada za przechowywanie i zarządzanie macierzą miast oraz odległości między nimi. Zawiera metody umożliwiające ładowanie macierzy z plików, tworzenie losowych macierzy, a także wyświetlanie i czyszczenie macierzy. W tej klasie znajduje się również struktura `Tour`, która przechowuje trasę (ścieżkę miast) oraz całkowity koszt tej trasy. Klasa ta zawiera także zmienne związane z parametrami algorytmów, takie jak `starting_temperature` i `cooling_rate`, wykorzystywane w algorytmie symulowanego wyżarzania, oraz czas, po którym algorytm ma zakończyć swoje działanie (`stop_criterion_time`).

Klasa `TSP` implementuje algorytmy rozwiązujące problem komiwojażera, w tym algorytm `Tabu Search`, symulowanego wyżarzania. Odpowiada za obliczanie kosztu trasy przy pomocy metody `calculatePathCost`, jak również za implementację poszczególnych algorytmów optymalizacyjnych, takich jak `tabuSearch` oraz `simulatedAnnealing`.

Klasa `Menu` służy do interakcji z użytkownikiem, umożliwiając wybór plików wejściowych, ustawienie kryteriów zakończenia działania algorytmu, wybór parametrów algorytmów oraz zapis wyników do pliku. Zawiera metody do ładowania danych z plików, tworzenia macierzy losowych oraz uruchamiania odpowiednich algorytmów na zadanych danych. Klasa ta jest odpowiedzialna za zarządzanie menu aplikacji, umożliwiając użytkownikowi łatwą konfigurację i uruchamianie różnych algorytmów rozwiązujących problem TSP.

Badanie algorytmów

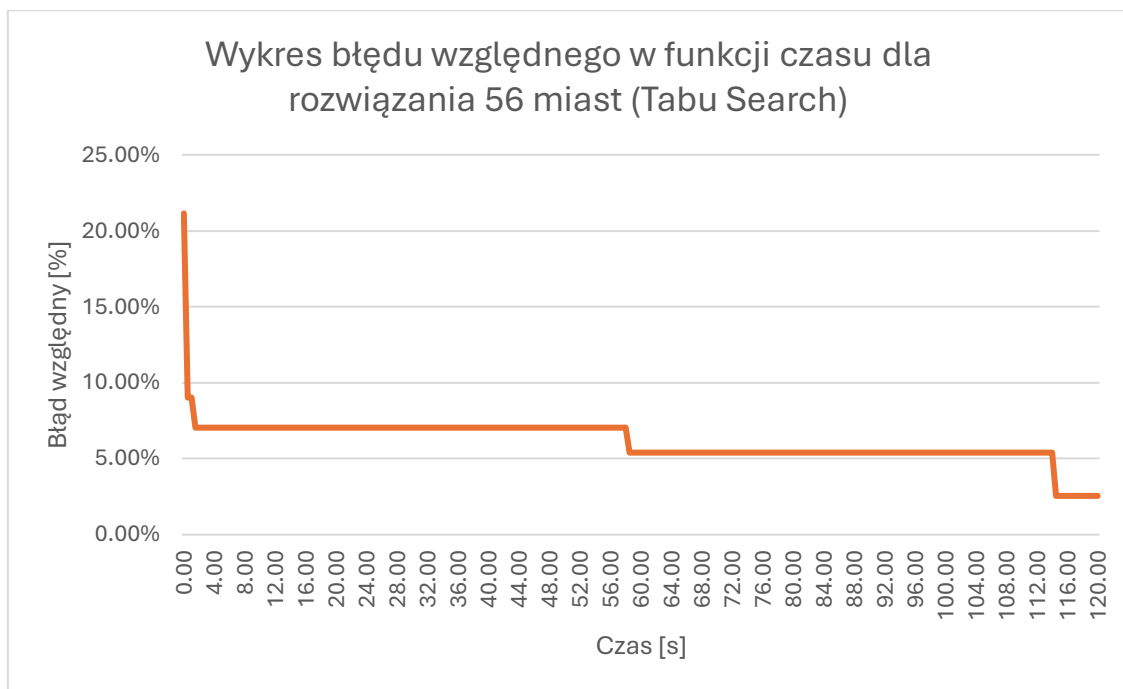
Tabele 1-3 wraz z wykresami 1-3 zawierają wyniki dla algorytmu `Tabu Search`. Tabele 4-9 wraz z wykresami 4 oraz 5 zawierają natomiast wyniki dla algorytmu `Simulated Annealing`.

Wyniki badań problemu 56 miast (Tabu Search)

Tabela 1. Fragment tabeli zawierającej wyniki badania problemu dla 56 miast (Tabu Search).

Best tour cost	Time [s]	Relative error
1948	0.00	21.14%
1753	0.50	9.02%
1753	1.00	9.02%
1721	1.50	7.03%
1721	56.00	7.03%
1721	56.50	7.03%
1721	57.00	7.03%
1721	57.50	7.03%
1721	58.00	7.03%
1695	58.50	5.41%
1695	59.00	5.41%
1695	59.50	5.41%
1695	114.00	5.41%
1649	114.50	2.55%

1649	115.00	2.55%
1649	115.50	2.55%
1649	116.00	2.55%
1649	116.50	2.55%
1649	117.00	2.55%
1649	117.50	2.55%
1649	118.00	2.55%
1649	118.50	2.55%
1649	119.00	2.55%
1649	119.50	2.55%
1649	120.00	2.55%



Wykres 1. Wykres błędu względnego w funkcji czasu dla rozwiązania 56 miast (Tabu Search).

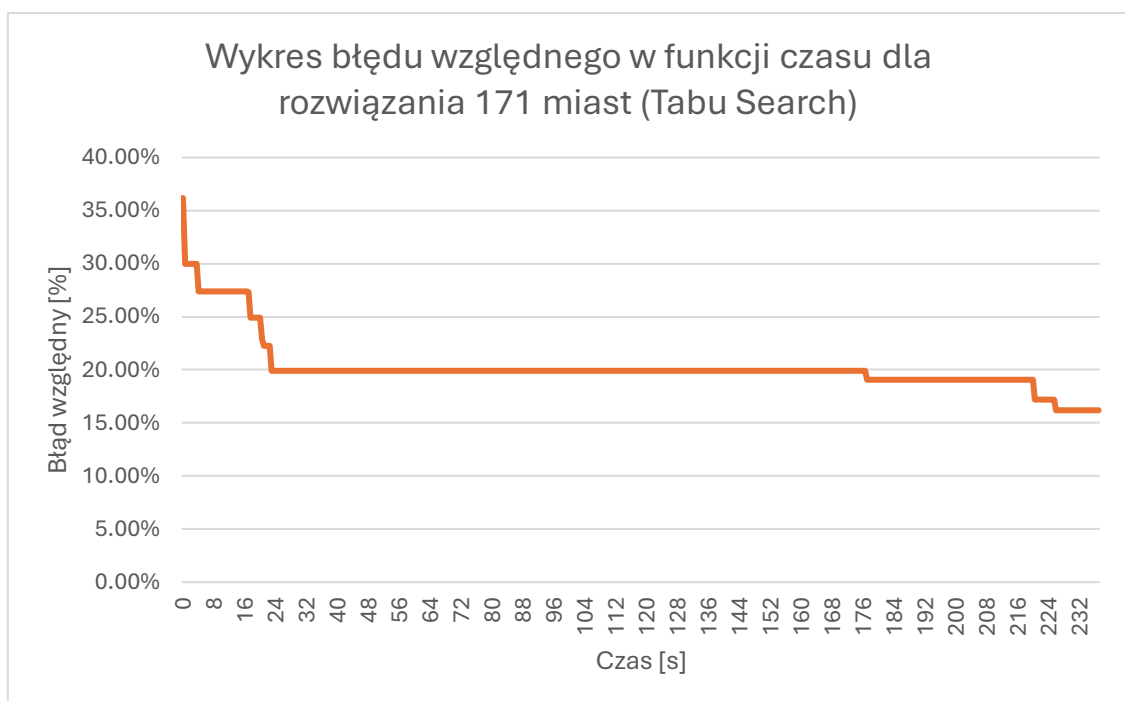
Wyniki badań problemu 171 miast (Tabu Search)

Tabela 2. Fragment tabeli zawierającej wyniki badania problemu dla 171 miast (Tabu Search).

Best tour cost	Time [s]	Relative error
3752	0	36.19%
3581	0.5	29.98%
3581	1	29.98%
3581	1.5	29.98%
3581	2	29.98%
3581	2.5	29.98%
3581	3	29.98%

3581	3.5	29.98%
3509	4	27.37%
3509	4.5	27.37%
3509	5	27.37%
3509	11	27.37%
3509	11.5	27.37%
3509	12	27.37%
3509	12.5	27.37%
3509	13	27.37%
3509	13.5	27.37%
3509	14	27.37%
3509	14.5	27.37%
3509	15	27.37%
3509	15.5	27.37%
3509	16	27.37%
3509	16.5	27.37%
3508	17	27.33%
3441	17.5	24.90%
3441	18	24.90%
3441	18.5	24.90%
3441	19	24.90%
3441	19.5	24.90%
3441	20	24.90%
3386	20.5	22.90%
3369	21	22.29%
3369	21.5	22.29%
3369	22	22.29%
3369	22.5	22.29%
3303	23	19.89%
3303	23.5	19.89%
3303	24	19.89%
3303	24.5	19.89%
3303	25	19.89%
3303	25.5	19.89%
3281	218	19.09%
3281	218.5	19.09%
3281	219	19.09%
3281	219.5	19.09%
3281	220	19.09%
3228	220.5	17.17%
3228	221	17.17%
3228	221.5	17.17%
3201	226	16.19%
3201	226.5	16.19%
3201	227	16.19%
3201	227.5	16.19%

3201	228	16.19%
3201	228.5	16.19%
3201	229	16.19%
3201	229.5	16.19%
3201	230	16.19%
3201	230.5	16.19%
3201	231	16.19%
3201	231.5	16.19%
3201	232	16.19%
3201	235.5	16.19%
3201	236	16.19%
3201	236.5	16.19%
3201	237	16.19%



Wykres 2. Wykres błędu względnego w funkcji czasu dla rozwiązania 171 miast (Tabu Search).

Wyniki badań problemu 359 miast (Tabu Search)

Tabela 3. Fragment tabeli zawierającej wyniki badania problemu dla 359 miast (Tabu Search).

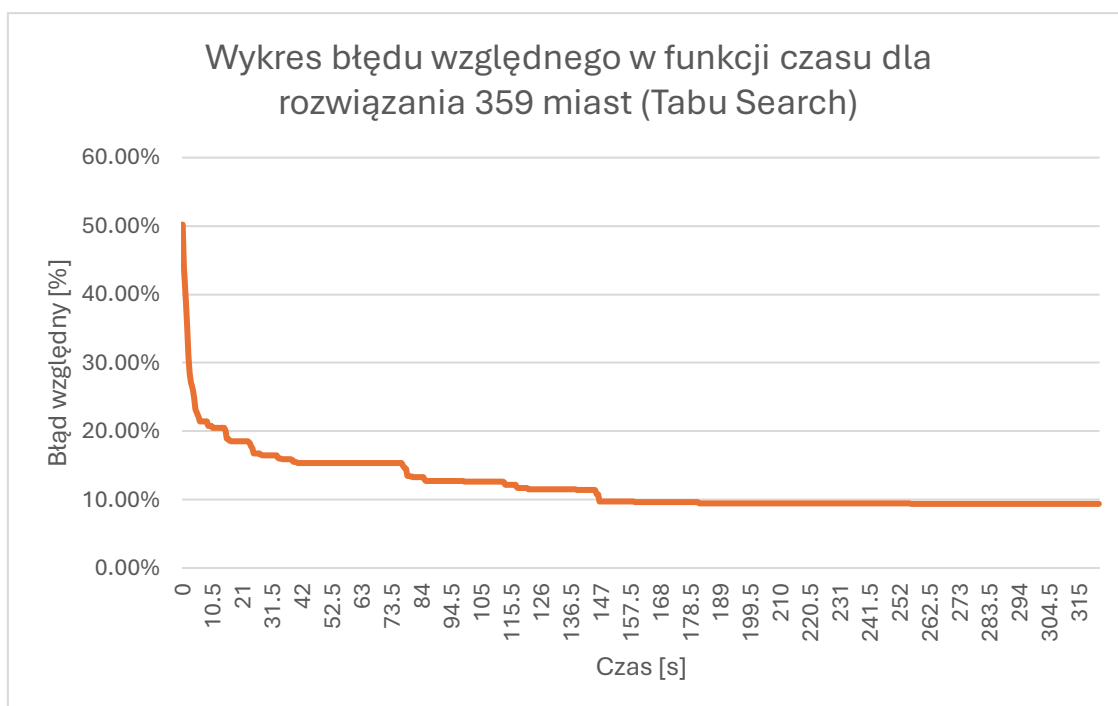
Best tour cost	Time [s]	Relative error
1747	0	50.21%
1676	0.5	44.11%
1619	1	39.21%
1577	1.5	35.60%
1525	2	31.13%
1494	2.5	28.46%

1479	3	27.17%
1467	3.5	26.14%
1452	4	24.85%
1433	4.5	23.22%
1428	5	22.79%
1420	5.5	22.10%
1412	6	21.41%
1412	6.5	21.41%
1412	7	21.41%
1412	7.5	21.41%
1412	8	21.41%
1412	8.5	21.41%
1405	9	20.81%
1405	9.5	20.81%
1405	10	20.81%
1401	10.5	20.46%
1401	11	20.46%
1401	11.5	20.46%
1401	12	20.46%
1401	12.5	20.46%
1401	13	20.46%
1401	13.5	20.46%
1401	14	20.46%
1401	14.5	20.46%
1397	15	20.12%
1383	15.5	18.92%
1383	16	18.92%
1380	16.5	18.66%
1379	17	18.57%
1379	17.5	18.57%
1379	18	18.57%
1379	18.5	18.57%
1379	19	18.57%
1379	19.5	18.57%
1379	20	18.57%
1379	20.5	18.57%
1379	21	18.57%
1379	21.5	18.57%
1379	22	18.57%
1379	22.5	18.57%
1378	23	18.49%
1375	23.5	18.23%
1370	24	17.80%
1366	24.5	17.45%
1358	25	16.77%
1358	25.5	16.77%

1358	26	16.77%
1358	26.5	16.77%
1358	27	16.77%
1356	27.5	16.60%
1355	28	16.51%
1355	28.5	16.51%
1355	29	16.51%
1355	29.5	16.51%
1355	30	16.51%
1355	30.5	16.51%
1355	31	16.51%
1355	31.5	16.51%
1355	32	16.51%
1355	32.5	16.51%
1355	33	16.51%
1350	33.5	16.08%
1349	34	15.99%
1349	34.5	15.99%
1348	35	15.91%
1348	35.5	15.91%
1348	36	15.91%
1348	36.5	15.91%
1348	37	15.91%
1348	37.5	15.91%
1348	38	15.91%
1347	38.5	15.82%
1344	39	15.56%
1344	39.5	15.56%
1342	40	15.39%
1341	40.5	15.31%
1341	41	15.31%
1341	41.5	15.31%
1341	42	15.31%
1341	77	15.31%
1337	77.5	14.96%
1333	78	14.62%
1332	78.5	14.53%
1320	79	13.50%
1320	79.5	13.50%
1319	80	13.41%
1319	80.5	13.41%
1313	85	12.90%
1311	85.5	12.73%
1311	86	12.73%
1311	86.5	12.73%
1311	95.5	12.73%

1311	96	12.73%
1311	96.5	12.73%
1311	97	12.73%
1311	97.5	12.73%
1311	98	12.73%
1311	98.5	12.73%
1310	99	12.64%
1310	111.5	12.64%
1310	112	12.64%
1310	112.5	12.64%
1309	113	12.55%
1304	113.5	12.12%
1304	114	12.12%
1304	114.5	12.12%
1304	115	12.12%
1304	115.5	12.12%
1304	116	12.12%
1304	116.5	12.12%
1304	117	12.12%
1300	117.5	11.78%
1299	118	11.69%
1299	118.5	11.69%
1299	119	11.69%
1297	122	11.52%
1297	122.5	11.52%
1297	123	11.52%
1297	123.5	11.52%
1297	135	11.52%
1297	135.5	11.52%
1297	136	11.52%
1297	136.5	11.52%
1297	137	11.52%
1297	137.5	11.52%
1297	138	11.52%
1296	138.5	11.44%
1296	139	11.44%
1296	144.5	11.44%
1296	145	11.44%
1288	145.5	10.75%
1288	146	10.75%
1276	146.5	9.72%
1276	147	9.72%
1276	147.5	9.72%
1276	148	9.72%
1276	158	9.72%
1276	158.5	9.72%

1275	159	9.63%
1275	159.5	9.63%
1275	160	9.63%
1275	160.5	9.63%
1275	161	9.63%
1275	161.5	9.63%
1275	181	9.63%
1273	181.5	9.46%
1273	182	9.46%
1273	182.5	9.46%
1273	183	9.46%
1273	183.5	9.46%
1273	255.5	9.46%
1272	256	9.37%
1272	256.5	9.37%
1272	257	9.37%
1272	257.5	9.37%
1272	258	9.37%
1272	258.5	9.37%
1272	259	9.37%
1272	259.5	9.37%
1272	260	9.37%
1272	321.5	9.37%
1272	322	9.37%



Wykres 3. Wykres błędu względnego w funkcji czasu dla rozwiązania 359 miast (Tabu Search).

Wyniki badań problemu 56 miast (Simulated Annealing)

Tabela 4. Fragment tabeli zawierającej wyniki badania problemu dla 56 miast i $\text{cooling_factor} = 0.9999$ (Simulated Annealing).

Best tour cost	Time [s]	Relative error
1948	0	21.14%
1948	0.5	21.14%
1948	1	21.14%
1948	1.5	21.14%
1948	2	21.14%
1948	2.5	21.14%
1948	3	21.14%
1948	3.5	21.14%
1948	4	21.14%
1948	4.5	21.14%
1948	5	21.14%
1948	5.5	21.14%
1948	6	21.14%
1948	6.5	21.14%
1948	7	21.14%
1948	7.5	21.14%
1948	8	21.14%
1948	8.5	21.14%
1948	9	21.14%
1948	9.5	21.14%
1773	10	10.26%
1731	10.5	7.65%
1731	114	7.65%
1728	114.5	7.46%
1723	115	7.15%
1723	115.5	7.15%
1723	116	7.15%
1723	116.5	7.15%
1723	117	7.15%
1723	117.5	7.15%
1723	118	7.15%
1723	118.5	7.15%
1723	119	7.15%
1723	119.5	7.15%

Tabela 5. Fragment tabeli zawierającej wyniki badania problemu dla 56 miast i $\text{cooling_factor} = 0.9995$ (Simulated Annealing).

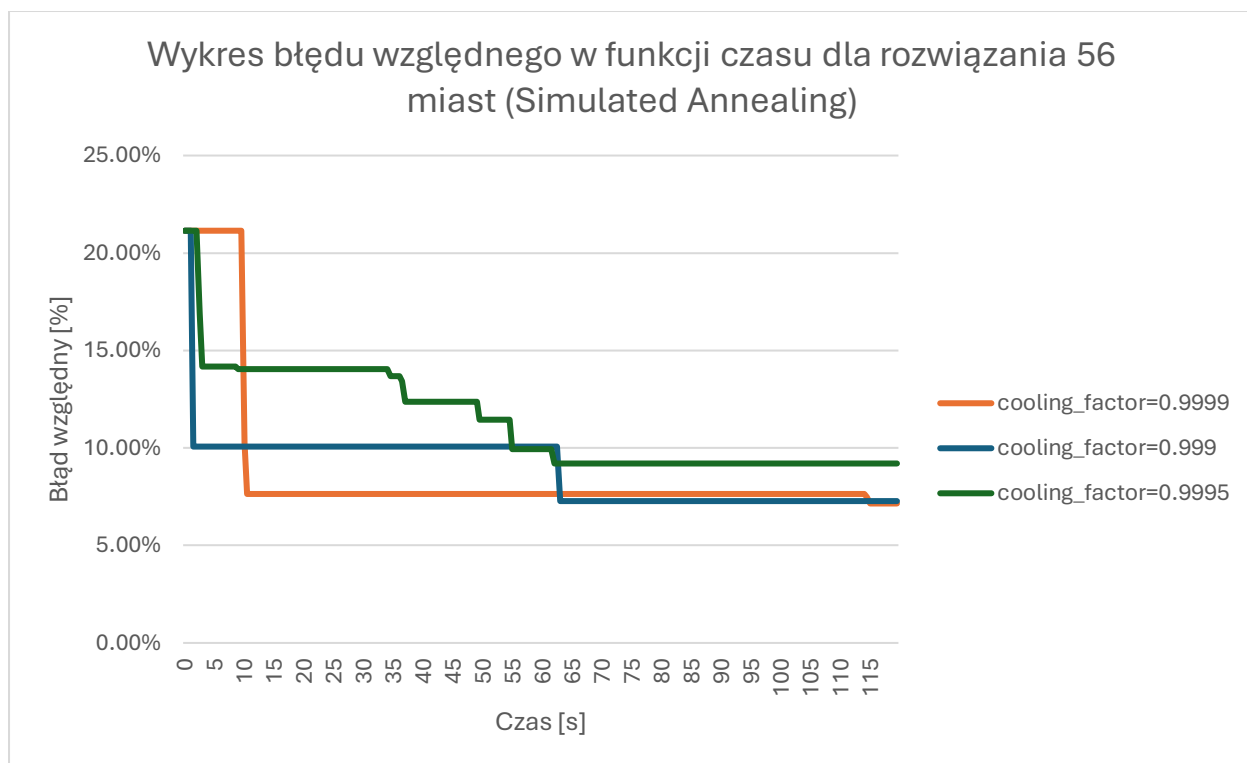
Best tour cost	Time [s]	Relative error
1948	0	21.14%

1948	0.5	21.14%
1948	1	21.14%
1948	1.5	21.14%
1948	2	21.14%
1880	2.5	16.92%
1836	3	14.18%
1836	3.5	14.18%
1836	4	14.18%
1836	8	14.18%
1836	8.5	14.18%
1834	9	14.05%
1834	9.5	14.05%
1834	34	14.05%
1828	34.5	13.68%
1828	35	13.68%
1828	35.5	13.68%
1828	36	13.68%
1824	36.5	13.43%
1807	37	12.38%
1807	48	12.38%
1807	48.5	12.38%
1807	49	12.38%
1792	49.5	11.44%
1792	50	11.44%
1792	50.5	11.44%
1792	51	11.44%
1792	51.5	11.44%
1792	52	11.44%
1792	52.5	11.44%
1792	53	11.44%
1792	53.5	11.44%
1792	54	11.44%
1792	54.5	11.44%
1768	55	9.95%
1768	55.5	9.95%
1768	56	9.95%
1768	61.5	9.95%
1756	62	9.20%
1756	62.5	9.20%
1756	119.5	9.20%

Tabela 6. Fragment tabeli zawierającej wyniki badania problemu dla 56 miast i $\text{cooling_factor} = 0.999$ (Simulated Annealing).

Best tour cost	Time [s]	Relative error
1948	0	21.14%

1948	0.5	21.14%
1948	1	21.14%
1770	1.5	10.07%
1770	62.5	10.07%
1725	63	7.28%
1725	119.5	7.28%



Wykres 4. Wykres błędu względnego w funkcji czasu dla rozwiązania 56 miast (Simulated Annealing).

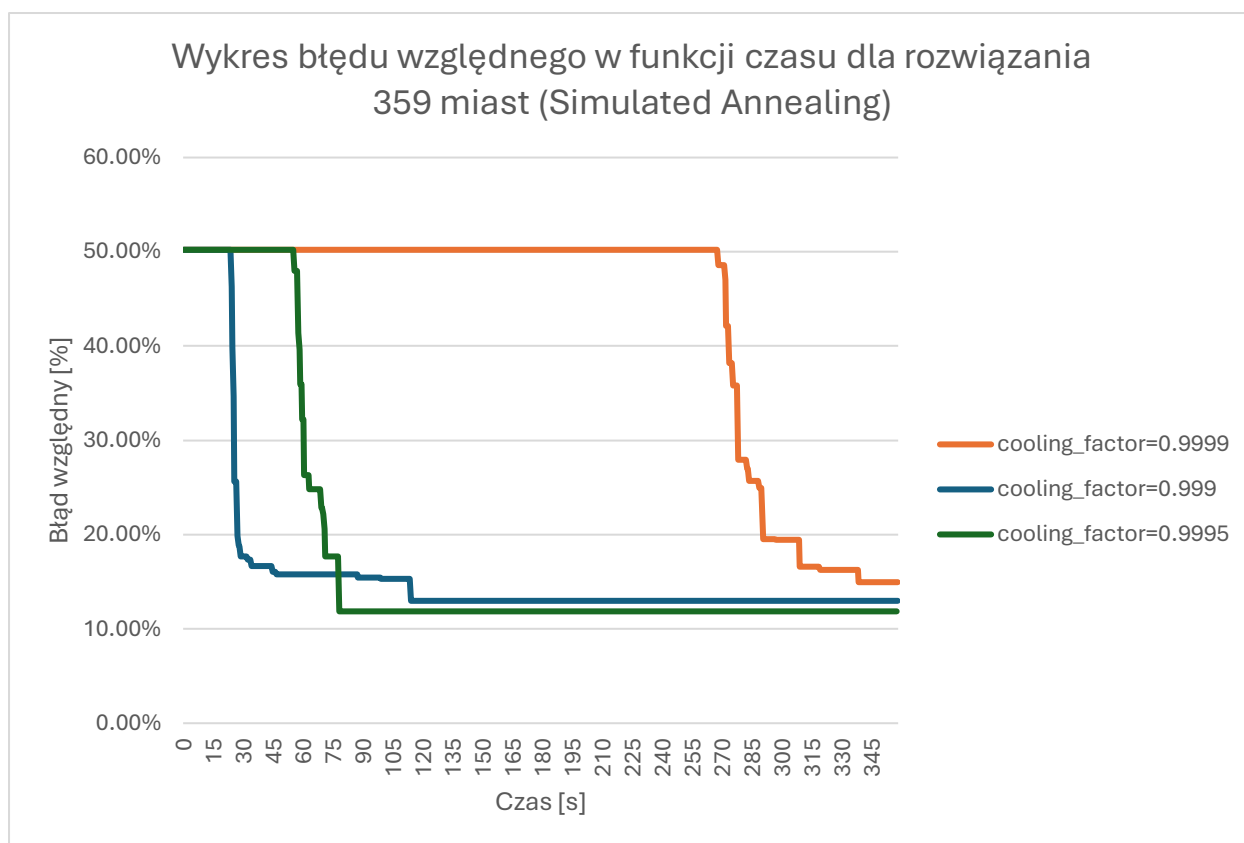
Wyniki badań problemu 359 miast (Simulated Annealing)

Tabela 7. Fragment tabeli zawierającej wyniki badania problemu dla 359 miast i $\text{cooling_factor} = 0.9999$ (Simulated Annealing).

Best tour cost	Time [s]	Relative error
1747	0	50.21%
1747	267	50.21%
1747	267.5	50.21%
1728	268	48.58%
1728	268.5	48.58%
1728	269	48.58%
1728	269.5	48.58%
1728	270	48.58%
1728	270.5	48.58%

1728	271	48.58%
1710	271.5	47.03%
1653	272	42.13%
1653	272.5	42.13%
1653	273	42.13%
1607	273.5	38.18%
1607	274	38.18%
1607	274.5	38.18%
1607	275	38.18%
1580	275.5	35.86%
1580	276	35.86%
1580	276.5	35.86%
1580	277	35.86%
1580	277.5	35.86%
1488	278	27.94%
1488	278.5	27.94%
1488	279	27.94%
1488	279.5	27.94%
1488	280	27.94%
1488	280.5	27.94%
1488	281	27.94%
1488	281.5	27.94%
1488	282	27.94%
1477	282.5	27.00%
1477	283	27.00%
1462	283.5	25.71%
1462	284	25.71%
1462	284.5	25.71%
1462	285	25.71%
1462	285.5	25.71%
1462	286	25.71%
1462	286.5	25.71%
1462	287	25.71%
1462	287.5	25.71%
1462	288	25.71%
1453	288.5	24.94%
1453	289	24.94%
1453	289.5	24.94%
1433	290	23.22%
1390	290.5	19.52%
1390	291	19.52%
1390	291.5	19.52%
1390	292	19.52%
1390	292.5	19.52%
1390	293	19.52%
1390	293.5	19.52%

1390	294	19.52%
1390	294.5	19.52%
1390	295	19.52%
1390	295.5	19.52%
1390	296	19.52%
1390	296.5	19.52%
1389	297	19.43%
1389	308	19.43%
1389	308.5	19.43%
1356	309	16.60%
1356	309.5	16.60%
1356	310	16.60%
1356	310.5	16.60%
1356	318.5	16.60%
1352	319	16.25%
1352	319.5	16.25%
1352	320	16.25%
1352	338	16.25%
1337	338.5	14.96%
1337	339	14.96%
1337	357.5	14.96%
1337	358	14.96%



Wykres 5. Wykres błędu względnego w funkcji czasu dla rozwiązania 359 miast (Simulated Annealing).

Tabela 8. Fragment tabeli zawierającej wyniki badania problemu dla 359 miast i $\text{cooling_factor} = 0.9995$ (Simulated Annealing).

Best tour cost	Time [s]	Relative error
1747	0	50.21%
1747	55	50.21%
1721	55.5	47.98%
1721	56	47.98%
1721	56.5	47.98%
1721	57	47.98%
1644	57.5	41.36%
1624	58	39.64%
1581	58.5	35.94%
1581	59	35.94%
1538	59.5	32.24%
1538	60	32.24%
1469	60.5	26.31%
1469	61	26.31%
1469	61.5	26.31%
1469	62	26.31%
1469	62.5	26.31%
1452	63	24.85%
1452	63.5	24.85%
1452	64	24.85%
1452	64.5	24.85%
1452	65	24.85%
1452	65.5	24.85%
1452	66	24.85%
1452	66.5	24.85%
1452	67	24.85%
1452	67.5	24.85%
1452	68	24.85%
1452	68.5	24.85%
1429	69	22.87%
1429	69.5	22.87%
1421	70	22.18%
1403	70.5	20.64%
1369	71	17.71%
1301	357	11.87%
1301	357.5	11.87%

Tabela 9. Fragment tabeli zawierającej wyniki badania problemu dla 359 miast i $\text{cooling_factor} = 0.999$ (Simulated Annealing).

Best tour cost	Time [s]	Relative error
1747	0	50.21%
1747	23.5	50.21%
1701	24	46.26%
1627	24.5	39.90%
1566	25	34.65%
1461	25.5	25.62%
1461	26	25.62%
1461	26.5	25.62%
1394	27	19.86%
1382	27.5	18.83%
1381	28	18.74%
1369	28.5	17.71%
1369	29	17.71%
1369	29.5	17.71%
1369	30	17.71%
1369	30.5	17.71%
1369	31	17.71%
1369	31.5	17.71%
1365	32	17.37%
1365	32.5	17.37%
1365	33	17.37%
1365	33.5	17.37%
1357	34	16.68%
1357	34.5	16.68%
1350	46	16.08%
1347	46.5	15.82%
1347	47	15.82%
1347	87	15.82%
1343	87.5	15.48%
1343	88	15.48%
1343	98.5	15.48%
1341	99	15.31%
1341	113.5	15.31%
1314	114	12.98%
1314	358	12.98%

Podsumowanie

Uzyskane trasy dla Tabu Search

ftv55 – 1649

15 17 52 26 25 24 42 22 41 21 50 23 54 27 49 43 53 44 28 45 30 46 34 1 3 4 6 5 47 48 31 29 0 33 2 13
35 7 32 8 36 9 37 11 19 20 40 18 39 38 10 51 14 12 16 15

ftv170 - 3201

142 149 161 152 14 15 159 16 18 19 20 29 22 23 24 25 26 27 28 30 31 34 35 39 38 37 75 11 10 76 74
12 13 17 21 32 158 36 157 33 41 42 155 156 40 45 44 46 47 48 51 52 53 43 55 54 58 59 60 50 49 168
72 73 78 82 79 80 81 0 1 77 71 61 68 67 167 70 87 85 86 83 84 69 66 63 64 56 57 62 65 88 153 154 89
90 91 94 96 165 163 98 95 92 93 108 166 107 106 105 97 99 101 100 103 117 118 119 120 121 122
123 162 102 104 110 109 114 164 127 126 125 129 128 130 131 132 133 134 7 8 9 2 3 4 5 169 111 112
113 115 116 124 137 138 135 139 140 141 6 151 160 150 148 144 143 147 136 146 145 142

rbg358 - 1272

266 43 12 4 44 226 263 27 24 17 13 3 164 218 321 260 55 34 9 135 39 227 315 64 36 29 337 91 37 28
334 187 205 272 167 25 38 225 11 81 145 6 246 15 284 327 8 23 212 22 114 97 208 63 5 345 193 233
19 104 98 107 94 239 21 90 69 305 180 47 150 203 74 214 52 328 314 59 125 170 66 14 80 0 88 58
257 286 297 197 93 26 112 103 175 172 141 219 196 118 95 178 320 310 77 42 248 139 106 181 171
16 86 165 101 144 123 89 276 204 317 35 18 92 275 46 238 192 206 182 306 336 32 31 232 249 87
288 119 199 209 283 347 83 122 303 200 234 244 211 210 265 207 127 67 253 128 293 99 129 185 54
57 213 130 111 299 222 316 224 134 82 223 136 85 84 33 179 173 108 116 138 113 268 190 271 245
140 110 304 142 109 61 68 353 287 153 65 251 155 151 143 236 298 60 319 243 156 30 357 53 117
158 149 241 189 342 282 274 159 121 285 161 152 290 162 133 335 105 174 2 100 322 349 195 163
56 247 169 348 343 313 240 191 311 146 72 307 40 177 296 216 255 184 270 356 312 331 186 346 79
339 230 147 132 20 102 262 332 258 194 73 62 51 131 340 281 324 279 308 75 50 352 259 329 183
176 228 71 96 217 201 49 78 115 278 273 220 250 215 338 277 302 301 344 291 221 267 326 325 350
137 231 148 126 202 45 7 330 261 242 323 351 41 294 120 254 160 256 318 300 264 341 280 229 309
154 124 333 166 237 355 269 235 188 70 295 289 157 198 10 48 354 168 252 292 76 1 266

Uzyskane trasy dla Simulated Annealing

ftv55 – 1723

15 17 52 0 33 1 2 26 24 42 22 41 21 50 23 54 27 49 45 28 53 43 44 30 46 34 31 29 25 20 40 19 18 39
10 51 13 35 6 5 47 48 3 4 7 32 8 36 9 37 11 38 14 12 16 15

ftv170 - 3752

-

rbg358 - 1301

266 43 215 242 207 255 320 310 225 11 6 246 50 344 288 34 9 220 250 12 163 145 353 313 119 188
239 21 87 83 156 30 333 48 278 277 302 200 134 82 253 229 309 178 184 79 77 42 248 140 110 323
336 269 90 305 272 256 318 181 104 97 135 298 103 132 171 213 37 28 16 231 114 20 328 61 72 172
319 243 10 282 274 235 202 285 222 316 224 68 124 99 283 347 291 169 348 292 233 78 63 346 259

260 223 158 149 226 76 1 86 350 210 108 116 167 56 284 247 194 67 95 128 293 139 106 241 189 295
 326 53 254 160 84 32 31 122 303 301 296 216 80 0 49 5 118 214 186 107 94 227 315 270 329 64 236
 69 39 345 289 159 121 120 175 115 281 324 262 332 306 351 197 130 111 331 138 113 268 177 340
 238 192 33 131 182 180 165 73 62 142 352 183 265 258 190 297 304 174 8 15 81 338 168 70 38 228
 71 314 263 27 273 237 355 147 234 244 357 325 211 240 204 327 26 112 312 101 127 98 307 40 219
 196 162 133 335 317 51 19 47 109 251 144 123 102 137 129 100 208 193 126 294 286 22 199 54 57
 217 201 252 356 195 88 58 261 354 24 157 330 257 136 349 299 93 198 35 18 245 339 230 187 205
 300 179 173 206 191 46 321 337 279 308 209 45 25 60 164 218 221 267 23 75 7 2 185 212 154 322 85
 105 153 65 334 117 249 66 14 17 13 3 311 290 161 152 176 271 146 89 276 343 287 92 275 342 59 44
 91 264 341 280 203 74 55 52 148 96 41 151 143 155 232 4 125 170 166 36 29 150 141 266

Heurystyczne algorytmy, takie jak przeszukiwanie tabu i symulowane wyżarzanie, osiągnęły w większości przypadków wyniki zbliżone do optymalnych w znacznie krótszym czasie w porównaniu do metod dokładnych. Obie metody wykazały się dużą skutecznością w znajdowaniu dobrych rozwiązań w ustalonym ograniczeniu czasowym, jednak jakość wyników była zależna od parametrów takich jak temperatura początkowa, współczynnik chłodzenia czy liczba iteracji. Badania wykazały, że wydłużenie czasu wykonywania algorytmów heurystycznych zwiększało ich szansę na znalezienie lepszych rozwiązań, co było widoczne w przypadku przeszukiwania tabu i symulowanego wyżarzania. Zależność ta była bardziej widoczna dla większych instancji problemu. Mimo tego, algorytmy potrafiły znaleźć rozwiązanie bliskie optymalnemu już na samym początku wywołania. Ostatecznie algorytmy heurystyczne, w odróżnieniu od metod dokładnych, wymagają głębszego przemyślenia problemu z jakim mamy do czynienia i odpowiedniego dobrania parametrów wywołania pod daną instancję.

TS szybciej znajdowało dobre rozwiązania w początkowej fazie działania, co było szczególnie widoczne, natomiast SA miał większą zdolność do eksploracji globalnej przestrzeni rozwiązań z uwagi na całkowicie losowe dobieranie sąsiadów. Dodatkowo TS charakteryzowało się większą stabilnością wyników. Algorytm, dzięki swojej deterministycznej strukturze (z pewnym stopniem losowości w generowaniu sąsiadów), częściej zwracał rozwiązania o zbliżonej jakości w kolejnych uruchomieniach. SA natomiast wykazywało większą zmienność wyników w zależności od parametrów, takich jak temperatura początkowa i współczynnik chłodzenia. W niektórych przypadkach mogło utknąć w lokalnych minimach, jeśli temperatura opadała zbyt szybko.

Bibliografia

- <https://www2.imm.dtu.dk/courses/02719/tabu/4tabu2.pdf>
- https://www.youtube.com/watch?v=mQ9tCn3vxHw&ab_channel=LuisR.Izquierdo
- http://www.pi.zarz.agh.edu.pl/intObl/notes/IntObl_w2.pdf
- Z.Michalkiewicz, D.B.Vogel, „Jak to rozwiązać czyli nowoczesna heurystyka”, WNT 2006