



Politechnika Wrocławska

Projektowanie efektywnych algorytmów
Rozwiązywanie problemu komiwojażera (TSP) za
pomocą różnych metod

Data: 23.01.2025r

Wykonawca: Jędrzej Radłowski 272927

Spis treści

Opis problemu.....	3
Opis zadania	3
Wybrane metody optymalizacyjne	4
Hipotezy badawcze.....	5
Specyfikacja sprzętu i procedura badawcza	6
Wyniki	7
Tabu Search.....	7
Plik ftv47.atsp	7
Plik ftv170.atsp	7
Plik rbg403.atsp.....	8
Wyniki ogólne	8
Simulated Annealing.....	9
Plik ftv47.atsp	9
Plik ftv170.atsp	9
Plik rbg403.atsp.....	10
Wyniki ogólne	11
Ant Colony Optimization	13
Plik ftv47.atsp	13
Plik ftv170.atsp	13
Plik rbg403.atsp.....	14
Wyniki ogólne	14
Wnioski końcowe	16

Opis problemu

Problem komiwojażera jest zagadnieniem optymalizacyjnym, należy do klasy problemów NP-trudnych, co oznacza, że znalezienie rozwiązania musi mieć złożoność przynajmniej wykładniczą. W praktyce oznacza to, że czasy sprawdzania, a tym samym znalezienie optymalnego rozwiązania zauważalnie zwiększają się, wraz ze wzrostem rozpatrywanego problemu.

Dotyczy on zagadnienia związanego z grafami ważonymi, a polega on na znalezieniu cyklu Hamiltona o jak najmniejszej sumie wag. Sam cykl Hamiltona natomiast charakteryzuje się tym, że każdy z wierzchołków zawiera się w tym cyklu tylko raz. W odniesieniu do samej nazwy, czyli komiwojażera (osoby która odwiedza różne miasta w celu sprzedaży dóbr), polega na wyznaczeniu najkrótszej trasy dla zbioru miejscowości, tak aby każda z miejscowości została odwiedzona tylko jeden raz.

Opis zadania

W ramach sprawozdania przeanalizowano efektywność wybranych metod rozwiązania problemu TSP, w szczególności analizę czasową danej metody, a co za tym idzie sytuację w której dana z metod warta jest wykonania.

Badane instancje danych pochodzą ze strony TSPLIB (<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>), zostały one zmodyfikowane tak, aby były kompatybilne z moim programem. W pierwszym wierszu podana jest liczba wierzchołków (miast), a w kolejnych same dane jako macierz odległości. Wielkość testowanych instancji to {48,171,403}. Zostały one dobrane tak, aby reprezentowały różnorodne przypadki problemu, przy jednoczesnym uwzględnieniu wyników poprzednio testowanych algorytmów. W tamtym badaniu maksymalna wielkość instancji wyniosła 25, zatem teraz została wybrana instancja większa, w celu podkreślenia różnicy wydajności tych algorytmów. W zależności od czasu wykonywania się programu, liczba testów się zmieniała, co dokładniej będzie opisane w dalszej części sprawozdania. Testom zostały poddane grafy asymetryczne. Jest to spowodowane, błędnym zapisaniem wielu instancji na stronie, z której to były one pobierane, a dodatkowo znacznym nakładem pracy, co spowodowałoby znaczne wydłużenie się czasu potrzebnego na badanie algorytmów.

Wybrane metody optymalizacyjne

W tym sprawozdaniu analizowane będą trzy różne metody służące znalezieniu rozwiązania dla problemu komiwojażera.

Metoda przeszukiwania z zakazami (Tabu Search)

Metoda przeszukiwania z zakazami to metaheurystyka, której celem jest znajdowanie rozwiązań optymalnych lub bliskich optymalnym w przestrzeniach kombinatorycznych. Algorytm ten opiera się na iteracyjnym poprawianiu rozwiązania początkowego poprzez eksplorację sąsiedztwa, przy jednoczesnym unikaniu powrotu do już odwiedzonych rozwiązań. Kluczowym elementem tej metody jest lista tabu, która przechowuje informacje o ruchach lub rozwiązaniach, które przez określoną liczbę iteracji są traktowane jako niedozwolone. Czas przebywania elementu na liście tabu jest kontrolowany za pomocą tzw. długości listy tabu, która stanowi istotny parametr algorytmu.

Metoda przeszukiwania z zakazami ma zdolność unikania lokalnych minimów dzięki możliwości przyjęcia rozwiązań chwilowo gorszych niż aktualnie najlepsze. W przeciwieństwie do algorytmów stricte heurystycznych, które mogą zatrzymać się na pierwszym napotkanym minimum lokalnym, Tabu Search systematycznie eksploruje przestrzeń rozwiązań. Zastosowanie tej metody w problemie komiwojażera (TSP) obejmuje eksplorację różnych tras poprzez modyfikacje takie jak odwracanie segmentów (np. techniki 2-opt i 3-opt) czy inne lokalne zmiany porządku miast. Tabu Search jest elastycznym narzędziem, które można dostosować do specyficznych problemów poprzez odpowiedni dobór parametrów.

Metoda symulowanego wyżarzania (Simulated Annealing)

Metoda symulowanego wyżarzania to algorytm inspirowany procesami fizycznymi, szczególnie termodynamicznym wyżarzaniem metali, podczas którego materiał schładzany jest w kontrolowany sposób, aby osiągnąć stan minimalnej energii. W zastosowaniach optymalizacyjnych algorytm ten wykorzystuje analogię między temperaturą, a prawdopodobieństwem akceptacji gorszych rozwiązań. Dzięki tej analogii Simulated Annealing umożliwia eksplorację globalnej przestrzeni rozwiązań, unikając utknięcia w lokalnym minimum.

Algorytm rozpoczyna się od losowego rozwiązania początkowego, lub rozwiązaniu pochodzącego z algorytmu najbliższego sąsiada oraz wartości temperatury, która stopniowo maleje zgodnie z wybranym schematem chłodzenia, np. geometrycznym lub logarytmicznym. Nowe rozwiązania są generowane poprzez lokalne modyfikacje obecnego rozwiązania, np. zamianę kolejności dwóch miast w przypadku problemu komiwojażera. Każde nowe rozwiązanie jest oceniane na podstawie różnicy kosztów między obecnym a nowym rozwiązaniem, a decyzja o jego akceptacji zależy od funkcji Boltzmann, która uwzględnia temperaturę. Na początku działania algorytmu, gdy temperatura jest wysoka, istnieje duże prawdopodobieństwo akceptacji rozwiązań o wyższym koszcie, co pozwala na szeroką

eksplorację przestrzeni. W miarę obniżania temperatury algorytm staje się bardziej rygorystyczny i skupia się na eksploracji lokalnego sąsiedztwa.

Teoretycznie, metoda symulowanego wyżarzania jest w stanie znaleźć rozwiązanie globalnie optymalne, jeśli temperatura spada wystarczająco wolno. W praktyce oznacza to odpowiedni dobór parametrów, takich jak początkowa temperatura, czy właśnie sposób i tempo jej schładzania.

Algorytm kolonii mrówek (Ant Colony Optimization)

Algorytm kolonii mrówek (ang. Ant Colony Optimization, ACO) to metaheurystyka inspirowana zachowaniem kolonii mrówek podczas poszukiwania najkrótszej drogi do źródła pokarmu. Kluczową cechą tej metody jest wykorzystanie mechanizmu śladów feromonowych, które są modelowane jako informacje wspólne dla wszystkich jednostek (mrówek) w systemie. Algorytm ten jest szczególnie efektywny w problemach dyskretnych, takich jak problem komiwojażera, gdzie jego zadaniem jest wyznaczenie najkrótszej trasy odwiedzającej wszystkie miasta.

Działanie ACO opiera się na iteracyjnym generowaniu tras przez mrówki, które podejmują decyzje o kolejnych miastach na podstawie kombinacji poziomu feromonów oraz wartości heurystycznych (np. odwrotności odległości). Wraz z przebiegiem algorytmu trasy, które prowadzą do lepszych wyników, są wzmacniane większą ilością feromonów, co zachęca kolejne mrówki do ich eksploracji. Jednocześnie, w celu zapobiegania zbyt szybkiemu skupieniu się na jednym rozwiązaniu, algorytm stosuje mechanizm parowania feromonów, który powoduje ich stopniowe zanikanie. Dzięki temu możliwa jest eksploracja nowych, obiecujących obszarów przestrzeni rozwiązań.

Ważnym elementem ACO jest równowaga między eksploracją a eksploatacją. W początkowych fazach algorytmu większy nacisk kładzie się na eksplorację, umożliwiając mrówkom przeszukiwanie szerokiego zakresu możliwych tras. W miarę postępu działania algorytmu eksploatacja dominujących tras staje się bardziej intensywna, co prowadzi do zbieżności w kierunku rozwiązania bliskiego optymalnemu. W teorii algorytm kolonii mrówek może być skutecznie skalowany na różne instancje problemu poprzez dostosowanie takich parametrów, jak liczba mrówek, współczynnik parowania feromonów czy waga heurystyki.

Hipotezy badawcze

1. Przekazywanie wyniku metody najbliższego sąsiada jako rozwiązanie początkowe dla algorytmów TS i SA zdecydowanie poprawi ich wynik końcowy.
2. Ustawienie odpowiednich parametrów będzie kluczowe do poprawnego działania programów.
3. Najcięższe będzie „dostroić” algorytm mrówkowy.

Specyfikacja sprzętu i procedura badawcza

Maszyna testowa wyposażona jest w procesor M3 Apple (8 rdzeni) o taktowaniu 4 x 3.2 GHz & 4 x 2 GHz. Posiada ona 16 GB zunifikowanej pamięci RAM. Badania zostały przeprowadzone w momencie gdy żaden inny program nie działał w tle, a sam komputer był podłączony do zasilania sieciowego. Same wyniki prezentowane są na sprzęcie na którym prowadzono badania.

Tak jak było wcześniej opisane, badania zostały przeprowadzone tylko dla instancji asymetrycznych. Wielkości badanych instancji to {48; 171; 403}. Dla algorytmów który wykonywały się mniej niż minutę, testowano je stukrotnie, w momencie gdy czas ten zaczął przekraczać minutę, liczbę testów zmniejszono do dwudziestu. Dla wszystkich algorytmów przyjęto maksymalny czas pojedynczej iteracji na wartość pięciu minut. W momencie gdy czas ten został przekroczony to iteracja została przerywana. Czasy, optymalna ścieżka, koszt przejścia a także błędy zostały zapisywane do plików .csv.

Wyniki

Tabu Search

Plik ftv47.atsp

Tabela 1. Uśrednione wyniki dla Tabu Search dla pliku ftv47.atsp

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia liczba iteracji	Średnia różnica błędu	Średni błąd [%]	Metoda
1776	300	2173	63720852,33	16852	22,3536	3-opt i NN
	300	4847	58806851	2902,5	162,429	3-opt i random
	300	2173	73060088,5	4389	30,0181	2-opt i NN
	300	4716	73124993,5	2940	164,541	2-opt i random

Tabela 1 przedstawia średnie czasy wykonania, średnią liczbę iteracji oraz średnie błędy dla czterech różnych metod. Widoczne jest gołym okiem, że algorytm o wiele lepiej działa gdy zostaje mu przekazany wynik z algorytmu najbliższego sąsiada jako rozwiązanie startowe. Po analizie jednak okazuje się, że jest to dokładnie rozwiązanie wyniki algorytmu najbliższego sąsiada, a przez pięć minut algorytm nie znajduje żadnego lepszego rozwiązania. Biorąc pod uwagę rozwiązania, gdzie pierwszym rozwiązaniem było rozwiązanie losowe, zauważyć można, że 3-opt i 2-opt uzyskują podobne wyniki, o około 2% rozwiązanie lepsze uzyskał 3-opt, co nie zmienia faktu iż rozwiązania te nie należą według mnie do najlepszych.

Plik ftv170.atsp

Tabela 2. Uśrednione wyniki dla Tabu Search dla pliku ftv170.atsp

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia liczba iteracji	Średnia różnica błędu	Średni błąd [%]	Metoda
2755	300	3582	23780115	827	30,0181	3-opt i NN
	300	21803	23741968	19048	691,3975	3-opt i random
	300	3582	27841513	827	30,0181	2-opt i NN
	300	21989	27945146	19234	698,149	2-opt i random

Tutaj podobnie jak poprzednio, zauważyć można, że algorytmu nie znajdują lepszego rozwiązania przez 300 sekund niż to, które jest wynikiem najbliższego sąsiada. Dodatkowo tutaj, przy wzroście liczby wierzchołków do 170, średni błąd procentowy dla pierwszego rozwiązania losowego wyniósł prawie 700%. Tutaj także 3-opt posiada minimalnie lepsze wyniki.

Plik rbg403.atsp

Tabela 3. Uśrednione wyniki dla Tabu Search dla pliku rbg403.atsp

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia liczba iteracji	Średnia różnica błędu	Średni błąd [%]	Metoda
2465	300	6854	11451868	4389	178,053	3-opt i NN
	300	19317	11409760,5	16852	683,651	3-opt i random
	300	6854	13195480,5	4389	178,053	2-opt i NN
	300	18055	13204329,5	15590	632,454	2-opt i random

Podobnie jak w poprzednich przypadkach, Tabu Search nie był w stanie znaleźć lepszego rozwiązania niż rozwiązanie algorytmu najbliższego sąsiada. Co jednak interesujące, przy wzroście liczby wierzchołków do 403, średni błąd procentowy dla pierwszego rozwiązania losowego wyniósł o nawet 60% mniej niż poprzednio. Tym razem także to 2-opt zwrócił lepsze wyniki.

Wyniki ogólne

Tabela 4. Wielkość błędu dla różnych instancji i metod

Rozmiar instancji	Średni błąd [%]	Metoda
48	22,3536	3-opt i NN
171	30,0181	3-opt i NN
403	178,053	3-opt i NN
48	162,429	3-opt i random
171	691,3975	3-opt i random
403	683,651	3-opt i random
48	30,0181	2-opt i NN
171	30,0181	2-opt i NN
403	178,053	2-opt i NN
48	164,541	2-opt i random
171	698,149	2-opt i random
403	632,454	2-opt i random

Jak widać w Tabeli 4 przekazanie rozwiązania NN znacznie poprawia wynik końcowy algorytmu Tabu Search. Dodatkowo zauważalna jest tutaj także niewielka przewaga 3-opt nad 2-opt.

Simulated Annealing

Plik ftv47.atsp

Wariant z chłodzeniem geometrycznym

Tabela 5. Uśrednione wyniki dla Simulated Annealing dla pliku ftv47.atsp, chłodzenie geometryczne

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia temperatura początkowa	Średnia temperatura końcowa	Średnia różnica błędów	Średni błąd [%]	Metoda	Współczynnik schładzania
1776	0,0252111	2131	44981,33333	46858,4	355	19,9887	3-opt dla NN, geometryczny	99
	0,209530567	2129	46537,66667	32696,69135	353	19,8761	3-opt dla NN, geometryczny	90
	1,4188041	2128	36940	33969,2445	352	19,8198	3-opt dla random, geometryczny	99
	0,187408633	2076	36659	34149,70694	300	16,8919	3-opt dla random, geometryczny	90
	9,55104	2173	60884,33333	9,94E-10	397	22,3536	2-opt dla NN, geometryczny	99
	0,928358667	2173	60877,33333	9,24E-10	406	22,3536	2-opt dla NN, geometryczny	90
	9,41901	2200	52618,33333	9,93E-10	424	23,8739	2-opt dla random, geometryczny	99
	0,915394	2671,333333	46811	9,60E-10	895	50,41293333	2-opt dla random, geometryczny	90

Tabela 5 przedstawia średnie czasy wykonania, średnie temperatury, a także błąd średni dla różnych metod przeszukiwania sąsiedztwa oraz schładzania temperatury. W tym przypadku można zauważyć, że 3-opt za każdym razem zwracał rozwiązanie, które mieściło się w 20% od rozwiązania optymalnego. Widać także, że współczynnik schładzania dla tego pliku nie miał zbyt dużego znaczenia. Jedyne rozwiązanie, które jakkolwiek drastycznie odbiegało od reszty to rozwiązanie algorytmu z użyciem 2-opt i pierwszym rozwiązaniem losowym. Dla reszty różnice są minimalne.

Wariant z chłodzeniem logarytmicznym

Tabela6. Uśrednione wyniki dla Simulated Annealing dla pliku ftv47.atsp, chłodzenie logarytmiczne

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia temperatura początkowa	Średnia temperatura końcowa	Średnia różnica błędów	Średni błąd [%]	Metoda	Współczynnik schładzania
1776	0,209625533	2129	48073,66667	34377,52568	353	19,8761	3-opt dla NN, logarytmiczny	99
	0,2072998	2115	36677,66667	33697,0905	339	19,61333333	3-opt dla random, logarytmiczny	99
	0,930256	2173	60537	9,19E-10	397	19,35056667	2-opt dla NN, logarytmiczny	99
	0,922037	2590	44290	9,50E-10	814	19,0878	2-opt dla random, logarytmiczny	90

Tabela 6 od tabeli 5 różni się tym, że tutaj brane są pod uwagę wyniki gdzie schładzanie odbywało się logarytmicznie. Zauważyć można, że zarówno 2-opt i 3-opt jak i NN i rozwiązanie losowe charakteryzują się podobnym błędem. Znaczną różnicą jest tutaj czas, 0.2s w porównaniu do 0.9s.

Plik ftv170.atsp

Wariant z chłodzeniem geometrycznym

Tabela 7. Uśrednione wyniki dla Simulated Annealing dla pliku ftv170.atsp, chłodzenie geometryczne

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia temperatura początkowa	Średnia temperatura końcowa	Średnia różnica błędów	Średni błąd [%]	Metoda	Współczynnik schładzania
2755	0,05484025	3306	140857,6667	141939,5	551	20	3-opt dla NN, geometryczny	99
	4,39138	3582	137175,6667	9,62425E-10	827	30,0181	3-opt dla NN, geometryczny	90
	23,3359514	3306,666667	110080	54188,66679	551,6666667	20,02416667	3-opt dla random, geometryczny	99
	4,234225	4342,666667	92454,33333	9,47129E-10	1587,666667	57,62856667	3-opt dla random, geometryczny	90
	22,453	3582	200606	9,94E-10	827	30,0181	2-opt dla NN, geometryczny	99
	2,173903333	3582	201206,3333	9,58E-10	2392,666667	30,0181	2-opt dla NN, geometryczny	90
	21,8531	8279	118767,6667	9,96E-10	5524	200,508	2-opt dla random, geometryczny	99
	2,098193333	9612	85324,33333	9,47E-10	6857	248,893	2-opt dla random, geometryczny	90

Tabela 7 przedstawia ponownie najważniejsze dane wynikowe dla algorytmu symulowanego wyżarzania, tym razem dla pliku ftv170.atsp. Algorytm zachował stabilność wyników. 3-opt ponownie wykazał przewagę nad 2-opt, szczególnie dla rozwiązań losowych, gdzie to różnica pomiędzy dwoma była znacząca.

Wariant z chłodzeniem logarytmicznym

Tabela 8. Uśrednione wyniki dla Simulated Annealing dla pliku ftv170.atsp, chłodzenie logarytmiczne

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia temperatura początkowa	Średnia temperatura końcowa	Średnia różnica błęd	Średni błąd [%]	Metoda
2755	4,38818	3582	137808,6667	9,66708E-10	827	30,0181	3-opt dla NN, logarytmiczny
	4,26108	4315	104691	9,56417E-10	1560	42,17783333	3-opt dla random, logarytmiczny
	2,191283333	3582	201220,6667	9,58E-10	827	30,0181	2-opt dla NN, logarytmiczny
	2,10453	8938	111380	9,33E-10	6183	56,6243	2-opt dla random, logarytmiczny

Tabela 8 przedstawia wyniki przy chłodzeniu logarytmicznym dla tej samej instancji. Wyniki, szczególnie dla 2-opt, znacząco uległy poprawie, w porównaniu do tych gdzie schładzano geometrycznie. Nadaj jednak 3-opt, szczególnie dla pierwszego rozwiązania losowego, jest znacznie lepszy od 2-opt.

Plik rbg403.atsp

Wariant z chłodzeniem geometrycznym

Tabela 9. Uśrednione wyniki dla Simulated Annealing dla pliku rbg403.atsp, chłodzenie geometryczne

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia temperatura początkowa	Średnia temperatura końcowa	Średnia różnica błęd	Średni błąd [%]	Metoda	Współczynnik schładzania
2465	89,78425	5824,666667	288643,6667	9,92856E-10	3360	136,295	3-opt dla NN, geometryczny	99
	8,64421	5973	425055	9,65783E-10	3508	142,312	3-opt dla NN, geometryczny	90
	89,35466667	5835	317352,3333	9,96041E-10	3370	136,714	3-opt dla random, geometryczny	99
	8,63654	5983	353136,3333	9,56417E-10	3518	142,718	3-opt dla random, geometryczny	90
	44,2514	6854	184720,3333	9,95E-10	4389	178,053	2-opt dla NN, geometryczny	99
	4,275763333	6854	186137,6667	9,85E-10	4558	178,053	2-opt dla NN, geometryczny	90
	45,0602	7330	592090,3333	9,98E-10	4863	197,3633333	2-opt dla random, geometryczny	99
	4,396796667	7593,666667	652367	9,39E-10	5129	208,0593333	2-opt dla random, geometryczny	90

W tabeli 9 widać wyniki dla największej instancji z chłodzeniem geometrycznym. Algorytm wykazał znaczny spadek jakości w porównaniu do mniejszych instancji, co sugeruje problemy ze skalowalnością.

Wariant z oraz chłodzeniem logarytmicznym

Tabela 10. Uśrednione wyniki dla Simulated Annealing dla pliku rbg403.atsp, chłodzenie logarytmiczne

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia temperatura początkowa	Średnia temperatura końcowa	Średnia różnica błęd	Średni błąd [%]	Metoda
2465	8,62794	5989	363720,3333	9,54306E-10	3524	142,9613333	3-opt dla NN, logarytmiczny
	8,69626	5986,666667	331197,6667	9,16387E-10	3522	143,1236667	3-opt dla random, logarytmiczny
	4,277813333	6854	187236,3333	9,91E-10	4389	143,1913333	2-opt dla NN, logarytmiczny
	4,39034	7571	654966	9,19E-10	5106	142,867	2-opt dla random, logarytmiczny

W Tabeli 10 przedstawione są wyniki dla chłodzenia logarytmicznego przy tej samej instancji. Chłodzenie logarytmiczne pozwoliło na osiągnięcie porównywalnych wyników z chłodzeniem geometrycznym, ale w krótszym czasie.

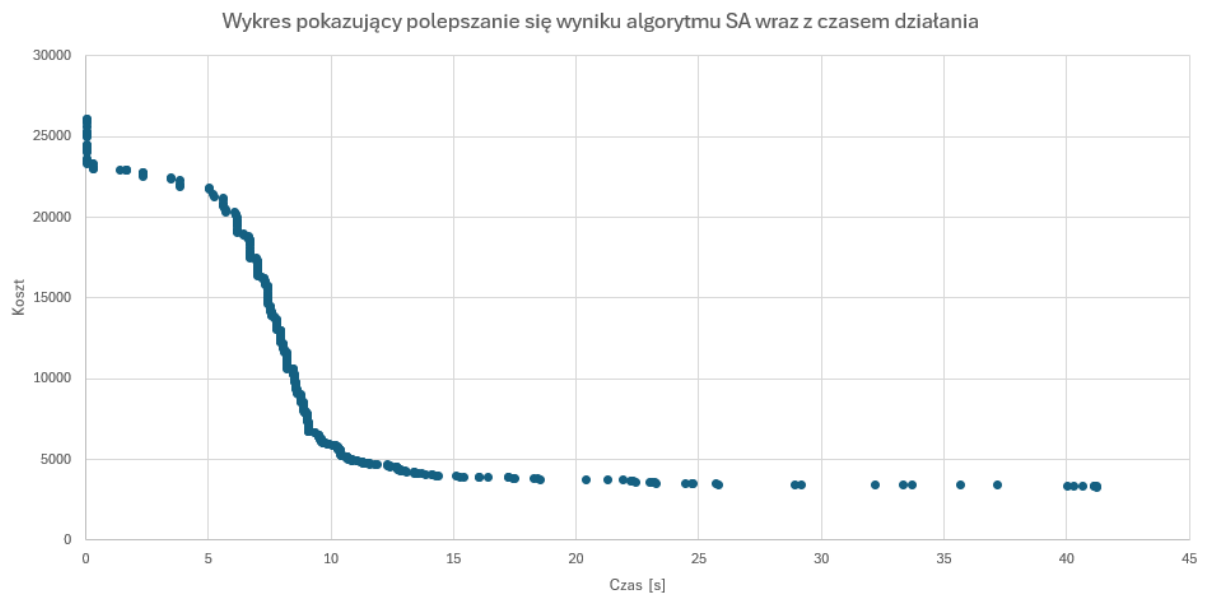
Wyniki ogólne

Tabela 11. Wielkość błędu dla różnych instancji i metod

Rozmiar instancji	Średni błąd [%]	Metoda
48	19,9324	3-opt dla NN, geometryczny
171	25,00905	3-opt dla NN, geometryczny
403	139,3035	3-opt dla NN, geometryczny
48	18,35585	3-opt dla random, geometryczny
171	38,82636667	3-opt dla random, geometryczny
403	139,716	3-opt dla random, geometryczny
48	22,3536	2-opt dla NN, geometryczny
171	30,0181	2-opt dla NN, geometryczny
403	178,053	2-opt dla NN, geometryczny
48	37,14341667	2-opt dla random, geometryczny
171	224,7005	2-opt dla random, geometryczny
403	202,7113333	2-opt dla random, geometryczny
48	19,8761	3-opt dla NN, logarytmiczny
171	30,0181	3-opt dla NN, logarytmiczny
403	142,9613333	3-opt dla NN, logarytmiczny
48	19,61333333	3-opt dla random, logarytmiczny
171	42,17783333	3-opt dla random, logarytmiczny
403	143,1236667	3-opt dla random, logarytmiczny
48	19,35056667	2-opt dla NN, logarytmiczny
171	30,0181	2-opt dla NN, logarytmiczny
403	143,1913333	2-opt dla NN, logarytmiczny
48	19,0878	2-opt dla random, logarytmiczny
171	56,6243	2-opt dla random, logarytmiczny
403	142,867	2-opt dla random, logarytmiczny

Tabela 11 przedstawia średni błąd w procentach a także jaka metoda została użyta, oraz wielkość badanej instancji. Ponownie zauważalna jest przewaga 3-opt nad 2-opt. Schemat schładzania logarytmiczny jest nieznacznie gorszy w przypadku 3-opt, a jednocześnie znacznie lepszy w przypadku 2-opt.

Rysunek 1. Wykres pokazujący polepszanie się wyniku algorytmu SA wraz z czasem działania



Na Rysunku 1 znajduje się wykres przedstawiający polepszanie się wyniku algorytmu SA wraz z czasem działania. Jest to rozwiązanie charakteryzujące się najmniejszym błędem dla instancji o wielkości 171. Jak można zauważyć, przez pierwsze 15 sekund algorytm ciągle znajduje rozwiązania o lepszym koszcie. Po 15 sekundzie trwania algorytmu, częstotliwość znajdowania lepszego rozwiązania maleje drastycznie, a samo rozwiązanie nie jest poprawiane o duże wartości.

Ant Colony Optimization

Plik ftv47.atsp

Tabela 12. Uśrednione wyniki dla Ant Colony Optimization dla pliku ftv47.atsp

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia różnica błęd	Średni błąd [%]	Liczba mrówek	Ulatnianie	Alfa	Beta	Min feromonów	Max feromonów	Iteracje
1776	0,568436	1968	191,6	10,788296	5	0,5	1	10	0,1	7	1500
	0,7374816	1959	183,4	10,326602	10	0,5	1	10	0,1	7	1500
	2,962944	1935	159,4	8,975226	50	0,5	1	10	0,1	7	1500
	0,73746	1945	168,8	9,504502	10	0,2	1	10	0,1	7	1500
	0,7482634	1961	185,4	10,439194	10	0,5	1	10	0,1	7	1500
	0,7425862	1966	190	10,69818	10	0,7	1	10	0,1	7	1500
	0,74629	1963	186,8	10,51802	10	0,7	0,5	10	0,1	7	1500
	0,7452118	1957	181,4	10,213968	10	0,7	1	10	0,1	7	1500
	0,7487004	1960	184,2	10,371616	10	0,7	10	10	0,1	7	1500
	0,738342	4016	2240,4	126,1484	10	0,7	1	1	0,1	7	1500
	0,7412906	2042	265,6	14,95494	10	0,7	1	5	0,1	7	1500
	0,7424982	1938	162	9,12162	10	0,7	1	10	0,1	7	1500
	0,7524476	1958	182	10,247746	10	0,7	1	10	0,1	10	1500
	0,748428	1973	197,4	11,114856	10	0,7	1	10	4	10	1500
	0,7452484	1956	180,4	10,157654	10	0,7	1	10	4	6	1500
	0,7472686	1956	180	10,135132	10	0,7	1	10	1	10	1500
	0,748266	1956	180,2	10,146388	10	0,7	1	10	1	5	1500
	0,74322	1964	188,4	10,60809	10	0,7	1	10	0,1	5	1500
	0,04870602	2014	237,6	13,37838	10	0,7	1	10	0,1	5	100
	0,4990316	1955	179	10,078834	10	0,7	1	10	0,1	5	1000
	0,09784812	1994	218	12,2748	10	0,7	1	10	0,1	5	200

Tabela 12 prezentuje czasy, liczbę iteracji oraz błędy dla algorytmu kolonii mrówek (ACO) przy najmniejszej instancji. Jak widać, dla tej instancji algorytm ten świetnie sobie poradził ze znalezieniem dobrej trasy – najmniejszy błąd to zaledwie 9%.

Plik ftv170.atsp

Tabela 13. Uśrednione wyniki dla Ant Colony Optimization dla pliku ftv170.atsp

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia różnica błęd	Średni błąd [%]	Liczba mrówek	Ulatnianie	Alfa	Beta	Min feromonów	Max feromonów	Iteracje
2755	4,8229875	3406	651,2	23,637	5	0,5	1	10	0,1	7	1500
	6,73979	3378	622,8	22,60618	10	0,5	1	10	0,1	7	1500
	30,19795	3296	541,2	19,6443	50	0,5	1	10	0,1	7	1500
	7,619102	3363	608,4	22,08348	10	0,2	1	10	0,1	7	1500
	7,290074	3354	599,4	21,7568	10	0,5	1	10	0,1	7	1500
	7,749766667	3381	625,8	22,71508	10	0,7	1	10	0,1	7	1500
	7,784412	3367	611,6	22,19962	10	0,7	0,5	10	0,1	7	1500
	7,72724	3381	625,8	22,71508	10	0,7	1	10	0,1	7	1500
	7,778218	3370	615,4	22,33756	10	0,7	10	10	0,1	7	1500
	8,100354	16097	13341,6	484,2684	10	0,7	1	1	0,1	7	1500
	7,9800975	3774	1019,4	37,00184	10	0,7	1	5	0,1	7	1500
	7,940774	3333	577,8	20,97278	10	0,7	1	10	0,1	7	1500
	7,88345	3367	611,6	22,19964	10	0,7	1	10	0,1	10	1500
	7,666638	3370	614,8	22,3158	10	0,7	1	10	4	10	1500
	7,797555	3346	591,4	21,46642	10	0,7	1	10	4	6	1500
	7,7210975	3382	627,2	22,76586	10	0,7	1	10	1	10	1500
	8,100754	3364	609,2	22,11254	10	0,7	1	10	1	5	1500
	8,575924	3396	641,2	23,27404	10	0,7	1	10	0,1	5	1500
	0,5426184	3477	721,6	26,19236	10	0,7	1	10	0,1	5	100
	5,47005	3372	617,2	22,4029	10	0,7	1	10	0,1	5	1000
	1,081334	3441,4	686,4	24,9147	10	0,7	1	10	0,1	5	200

Tabela 13 prezentuje wyniki dla średniej co do wielkości instancji. Tutaj już średni błąd jest większy niż poprzednio, lecz nadal są to zadowalające wyniki, szczególnie dla takich wyników czasowym. Zauważyć można, że ta sama konfiguracja parametrów znowu charakteryzuje się najlepszym wynikiem. Trwa ona jednak znacząco dłużej niż pozostałe, więc jeżeli nie zależy nam na najdokładniejszym rozwiązaniu, to warto spojrzeć na rozwiązanie o wartości parametrów : {10, 0.7, 1, 10, 0.1, 7}, które jest gorsze o około 33 (koszt drogi), a trwa o cztery razy krócej.

Plik rbg403.atsp

Tabela 14. Uśrednione wyniki dla Ant Colony Optimization dla pliku rbg403.atsp

Optymalna długość drogi	Średni czas [s]	Średnia długość drogi	Średnia różnica błęd	Średni błąd [%]	Liczba mrówek	Ulatnianie	Alfa	Beta	Min	Max	Iteracje
2465	27,87632	6514	4049	164,2594	5	0,5	1	10	0,1	7	1500
	43,86524	6500,6	4035,6	163,7158	10	0,5	1	10	0,1	7	1500
	191,2928	6469,4	4004,4	162,4502	50	0,5	1	10	0,1	7	1500
	44,52836	6499	4034	163,651	10	0,2	1	10	0,1	7	1500
	44,55776	6499	4034	163,6512	10	0,5	1	10	0,1	7	1500
	44,72494	6489,4	4024,4	163,2616	10	0,7	1	10	0,1	7	1500
	44,81986	6495,4	4030,4	163,5052	10	0,7	0,5	10	0,1	7	1500
	44,95874	6485,8	4020,8	163,1156	10	0,7	1	10	0,1	7	1500
	44,77284	6490,6	4025,6	163,3104	10	0,7	10	10	0,1	7	1500
	44,91374	8290,2	5825,2	236,3164	10	0,7	1	1	0,1	7	1500
	44,85852	6568,4	4103,4	166,4664	10	0,7	1	5	0,1	7	1500
	44,70872	6493,6	4028,6	163,432	10	0,7	1	10	0,1	7	1500
	44,7921	6497,2	4032,2	163,5782	10	0,7	1	10	0,1	10	1500
	43,3702	6501,4	4036,4	163,7486	10	0,7	1	10	4	10	1500
	43,40868	6487,4	4022,4	163,1806	10	0,7	1	10	4	6	1500
	43,0337	6499,6	4034,6	163,6756	10	0,7	1	10	1	10	1500
	43,05212	6498	4033	163,6104	10	0,7	1	10	1	5	1500
	44,77208	6501,2	4036,2	163,7402	10	0,7	1	10	0,1	5	1500
	2,994814	6535,2	4070,2	165,1196	10	0,7	1	10	0,1	5	100
	31,39635	6515,5	4050,5	164,3205	10	0,7	1	10	0,1	5	1000

W tabeli 14 prezentują się wyniki dla największej instancji. Tutaj już, algorytm miał problem ze znalezieniem dobrego rozwiązania. Tutaj już wybór parametrów nie miał tak dużego znaczenia, większość wyników jest bowiem bardzo podobna.

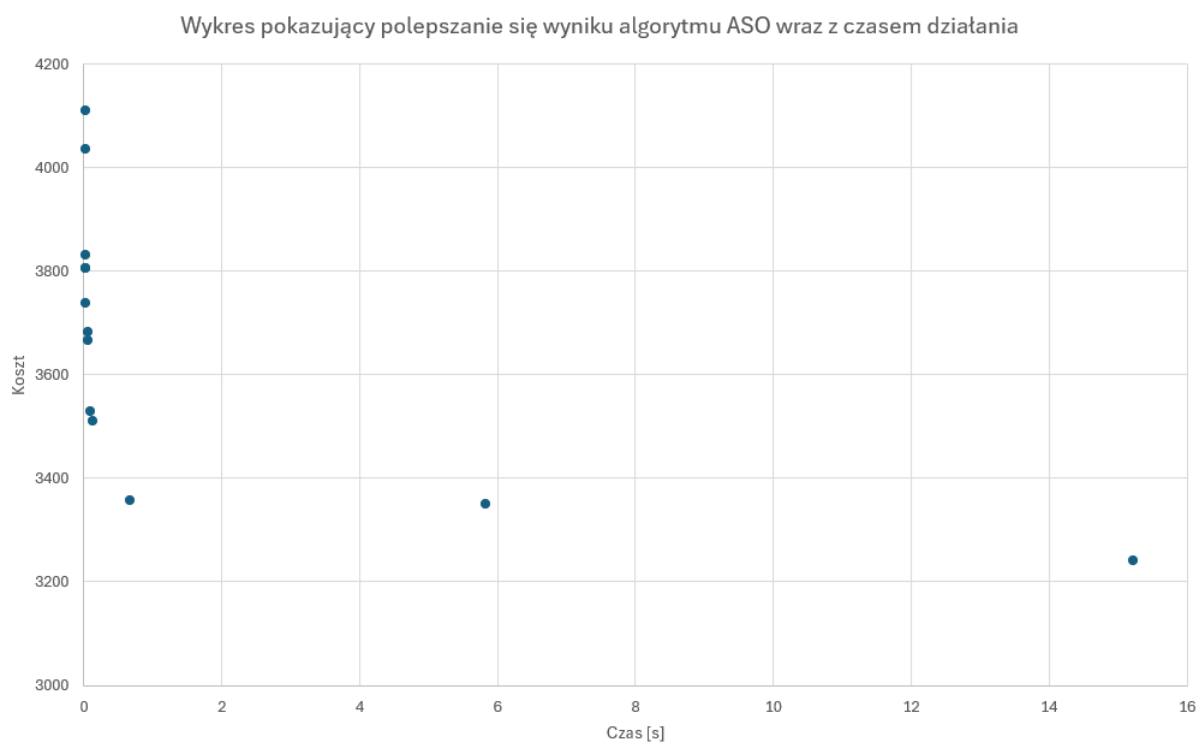
Wyniki ogólne

Tabela 15. Wielkość błędu dla różnych instancji

Rozmiar instancji	Średni błąd [%]
48	16,20011638
171	45,21820857
403	167,405445

Jak widać w Tabeli 15 średni błąd nie wzrasta znacząco między pierwszym i drugim rozmiarem instancji. Zauważalny wzrost widoczny jest między drugim a trzecim rozmiarem. Wyjaśnieniem takiego zjawiska może być fakt, że instancja o rozmiarze 403, charakteryzuje się tym, że graf tam znajdujący się nie jest pełen. Należałoby natomiast przeprowadzić więcej badań aby potwierdzić to przypuszczenie.

Rysunek 2. Wykres pokazujący polepszanie się wyniku algorytmu ACO wraz z czasem działania



Na Rysunku 2 znajduje się wykres przedstawiający polepszanie się wyniku algorytmu ACO wraz z czasem działania. Jest to rozwiązanie charakteryzujące się najmniejszym błędem dla instancji o wielkości 171. Jak można zauważyć, przez pierwsze 1 sekund algorytm ciągle znajduje rozwiązania o lepszym koszcie. Po 2 sekundzie trwania algorytmu, częstotliwość znajdowania lepszego rozwiązania maleje drastycznie, bo rozwiązanie jest poprawiane jedynie 2 razy, a samo rozwiązanie nie jest poprawiane o duże wartości.

Wnioski końcowe

Sprawozdanie w formie w jakiej jest obecnie nie pozwala na dokładne wyciągnięcie wniosków dotyczących efektywności testowanych algorytmów. Problem TSP, będący problemem NP-trudnym, stawia wysokie wymagania przed algorytmami stosowanymi do jego rozwiązania. Badania należy uzupełnić o brakujące aktualnie pomiary, wtedy też będzie można poprawnie wysnuć wnioski dotyczące każdej z metod. Przy obecnych badaniach wnioski jak i pytania do każdego z algorytmów prezentują się następująco.

Algorytm Tabu Search, w badaniach wyszedł najgorzej. Należałoby jednak przeprowadzić dokładniejszą analizę czasu w którym algorytm znajdował lepsze rozwiązanie. Dla aktualnych danych algorytm za każdym razem działał przez 5 minut, więc bez dogłębnej analizy pojedynczej iteracji algorytmu ciężko powiedzieć kiedy faktycznie znalezione było rozwiązanie. Dodatkowo należałoby przeprowadzić dodatkowe testy sprawdzające wpływ rozmiaru listy tabu (teraz ustawione $0.6 * \text{wielkość instancji}$) oraz wielkości kadencji.

Algorytm Simulated Annealing wypadł lepiej od poprzedniego. Tutaj także sprawdzono dokładnie każdą z iteracji, wiadome częstotliwość poprawiania rozwiązania przez algorytm jest duża, a kiedy spada ona, przy jednoczesnym niskim poprawianiu się wyników. Zbadano także wpływ różnych schematów schładzania temperatury, dzięki czemu zauważono nieznaczną przewagę schematu geometrycznego w przypadku 3-opt i znaczną poprawę działania algorytmu przy schemacie logarytmicznym dla 2-opt. Warto tutaj wspomnieć, że temperatura jest wyznaczana metodą Hao Chena. Szacuje ona średnią różnicę energii między losowymi rozwiązaniami. Dzięki temu algorytm dostosowuje temperaturę, co pozwala efektywnie eksplorować przestrzeń rozwiązań

Algorytm Ant Colony Optimization zgodnie z przewidywaniem wypadł najlepiej. Dogłębna analiza każdego z wyników, sprawia że wiadome jest kiedy częstotliwość poprawiania rozwiązania jest duża, a także kiedy ona spada. Co ciekawe, zauważono że wpływ parametrów na działanie algorytmu nie wpływał znacząco na działanie algorytmu. Warto podkreślić, że liczba mrówek nie jest równa liczbie miast, jest znacząco mniejsza. Jest to spowodowane początkowymi testami, które ukazały, że różnica w liczbie mrówek między 10 a 141, jedyne co zmienia to znacząco wydłuża czas trwania algorytmu, rozwiązanie nie jest znacząco, a czasami w ogóle, lepsze. Tutaj także zauważalna jest znacząca różnica między dwoma rozmiarami instancji, co prawdopodobnie spowodowane jest badaniem grafu niepełnego. Należałoby przetestować instancję podobnie dużą, lecz tym razem charakteryzującą się posiadaniem grafu pełnego.

(KOMENTARZ – sprawozdanie nie jest w formie pełnej, co spowodowane jest brakiem czasu na odpowiednią ilość testów. Przesyłam sprawozdanie w takiej formie, aby udokumentować aktualny stan moich badań. Badania jednak będę dalej prowadził tak aby doprowadzić do o wiele bardziej akceptowalnej formy sprawozdania, a samo końcowe sprawozdanie prześlę w momencie gdy skończę przeprowadzać resztę badań)