

Projektowanie Efektywnych Algorytmów

Sprawozdanie z zadania projektowego numer 3 - *Implementacja i analiza efektywności algorytmu genetycznego (ewolucyjnego) dla wybranego optymalizacyjnego 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 efektywności **Algorytmu Genetycznego (ang. GA – Genetic Algorithm)**.

Omówienie algorytmu

Algorytm genetyczny jest heurystyczną metodą poszukiwania rozwiązań, która realizuje przetwarzanie genetyczne. Opiera się on na koncepcji ewolucji, gdzie populacja potencjalnych rozwiązań (chromosomów) podlega procesom selekcji, krzyżowania i mutacji, w celu znalezienia optymalnego rozwiązania problemu. Rozwiązania problemu są kodowane jako chromosomy, które mogą przyjmować formę ciągów binarnych, liczb całkowitych, sekwencji genów o allelach ze zbioru $\{-1, 0, 1\}$, lub liczb rzeczywistych. Wybór reprezentacji zależy od specyfiki problemu, a w przypadku TSP chromosomem jest ścieżka będąca potencjalnym rozwiązaniem.

Algorytm zaczyna od wygenerowania początkowej populacji chromosomów, która może być losowa lub oparta na innych heurystykach. Populacja jest zbiorem potencjalnych rozwiązań. Każdy chromosom w populacji jest oceniany za pomocą funkcji oceny (inaczej funkcji przystosowania), która określa jakość danego rozwiązania. Funkcja oceny powinna odzwierciedlać cel problemu, a jej optymalizacja powinna prowadzić do znalezienia poszukiwanego rozwiązania. Na podstawie oceny, wybierane są chromosomy, które będą brały udział w tworzeniu kolejnego pokolenia. Chromosomy o wyższej wartości funkcji oceny mają większe szanse na wybór. Selekcja może być deterministyczna lub probabilistyczna. Najczęściej stosowane metody selekcji to selekcja turniejowa i selekcja proporcjonalna. Z wybranych rodzicielskich chromosomów tworzone są nowe chromosomy potomne poprzez wymianę fragmentów ich kodu genetycznego. Krzyżowanie ma na celu łączenie dobrych cech z różnych chromosomów. W chromosomach potomnych losowo dokonywane są drobne zmiany, co

pozwała na wprowadzenie do populacji nowej różnorodności i uniknięcie przedwczesnej zbieżności. Mutacja jest operatorem jednoargumentowym, w odróżnieniu do krzyżowania, które jest operatorem dwuargumentowym. Nowo utworzone chromosomy potomne zastępują część lub całość starych chromosomów w populacji. W zależności od strategii selekcji, rodzice mogą pozostać w populacji lub zostać całkowicie zastąpieni przez potomstwo. Proces oceny, selekcji i operacji genetycznych jest powtarzany do osiągnięcia warunku zatrzymania, np. osiągnięcia zadowalającego rozwiązania lub braku poprawy w ostatnich iteracjach.

Generacja populacji początkowej

Prezentowany algorytm rozpoczyna od generowania populacji początkowej. Realizuje to zadanie poprzez wypełnienie jej wszystkimi możliwymi rozwiązaniami znalezionymi **metodą zachłanną** (po jednej na każdy wierzchołek) i dopełnieniem populacji rozwiązaniami losowymi.

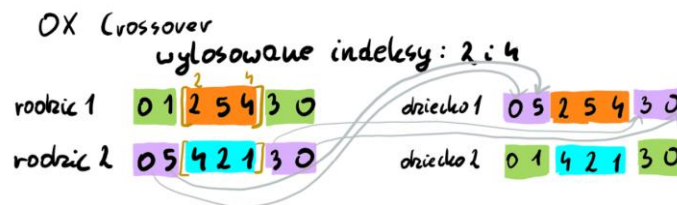
Selekcja

Kolejnym krokiem jest selekcja rodziców. Odbывается się ona przy użyciu **metody ruletkowej** (ang. **roulette wheel selection** lub **fitness proportionate selection**). Dzięki zastosowaniu tej metody rodzice o najlepszym przystosowaniu (według funkcji celu) będą wybierani z największym prawdopodobieństwem.

Operatory krzyżowania i mutacji

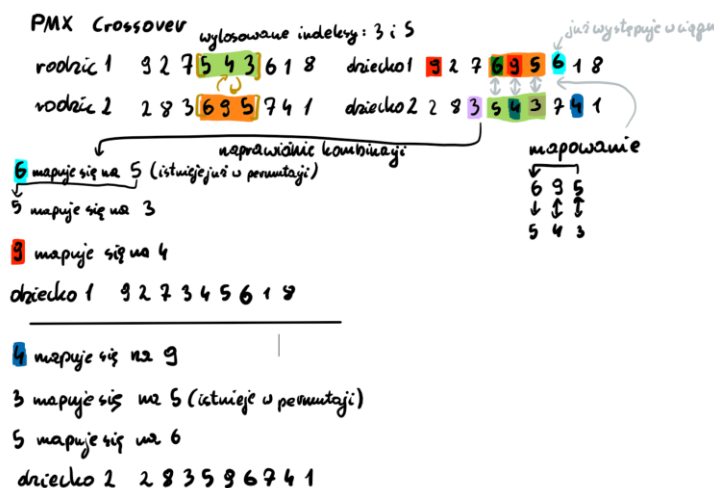
Następnym krokiem algorytmu jest przeprowadzenie operacji krzyżowania i mutacji na wybranych rodzicach, aby stworzyć nowe trasy potomne. Wprowadzone zostały dwa operatory krzyżowania:

- OX (ang. Order Crossover)



Rysunek 1. Przykład krzyżowania OX.

- PMX (ang. Partially Mapped Crossover)



Rysunek 2. Przykład krzyżowania PMX.

oraz dwa operatory mutacji:

- Swap

Swap Mutation
wylosowane indeksy: 1 i 4
0 1 2 3 4 0 → 0 4 2 3 1 0

Rysunek 3. Przykład mutacji Swap.

- Inverse

Inverse Mutation
wylosowane indeksy: 1 i 4
0 1 2 3 4 0 → 0 4 2 3 1 0

Rysunek 4. Przykład mutacji Inverse.

Wybór nowej populacji (sukcesja)

Końcowym etapem iteracji algorytmu jest sukcesja. Realizowana jest ona poprzez zmieszanie aktualnej populacji z wytworzonym na wcześniejszym etapie potomstwem. Taka mieszanka rozwiązań jest sortowana i nową populację stanowi najlepsze N rozwiązań, gdzie N jest wielkością populacji.

Algorytm w każdym z badań uruchamiany był na 2 minuty. Problemem, który rozwiązywał jest asymetryczny TSP zaproponowany przez pracowników Uniwersytetu Heidelberg ([ftv170](#)).

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 algorytmu, takie jak `mutation_factor` i `crossover_factor` oraz czas, po którym algorytm ma zakończyć swoje działanie (`stop_criterion_time`).

Klasa `TSP` implementuje algorytm genetyczny rozwiązujący problem asymetryczny komiwojażera. Znajdują się w niej również wszystkie potrzebne metody wspomagające działanie samego algorytmu genetycznego jak i działania całego programu pod względem algorytmiki.

Klasa `Menu` służy do interakcji z użytkownikiem, umożliwiając wybór plików wejściowych, ustawienie kryteriów zakończenia działania algorytmu oraz wybór parametrów algorytmu. 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

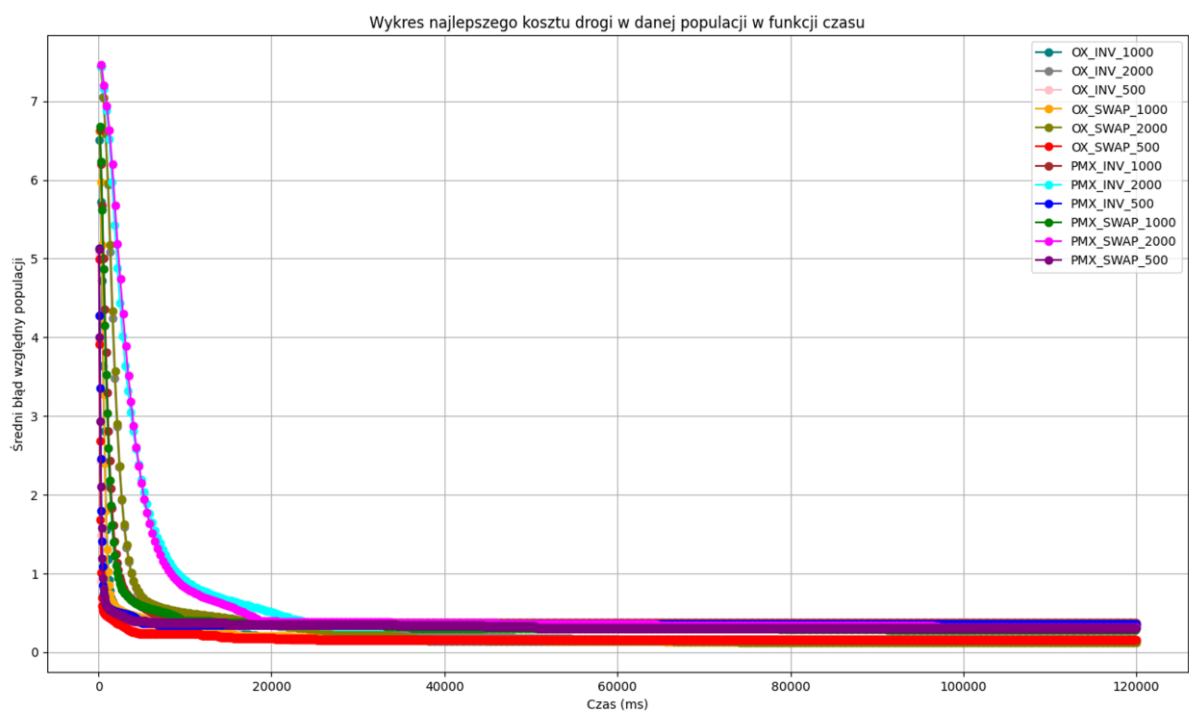
Wyniki badań wpływu rozmiaru populacji

Tabela 1 oraz wykresy 1 i 2 zawierają rezultaty badania wpływu wielkości populacji na wyniki algorytmu dla trzech różnych wartości (500, 1000 oraz 2000 osobników). Współczynnik krzyżowania wynosi 0.8, a współczynnik mutacji 0.01.

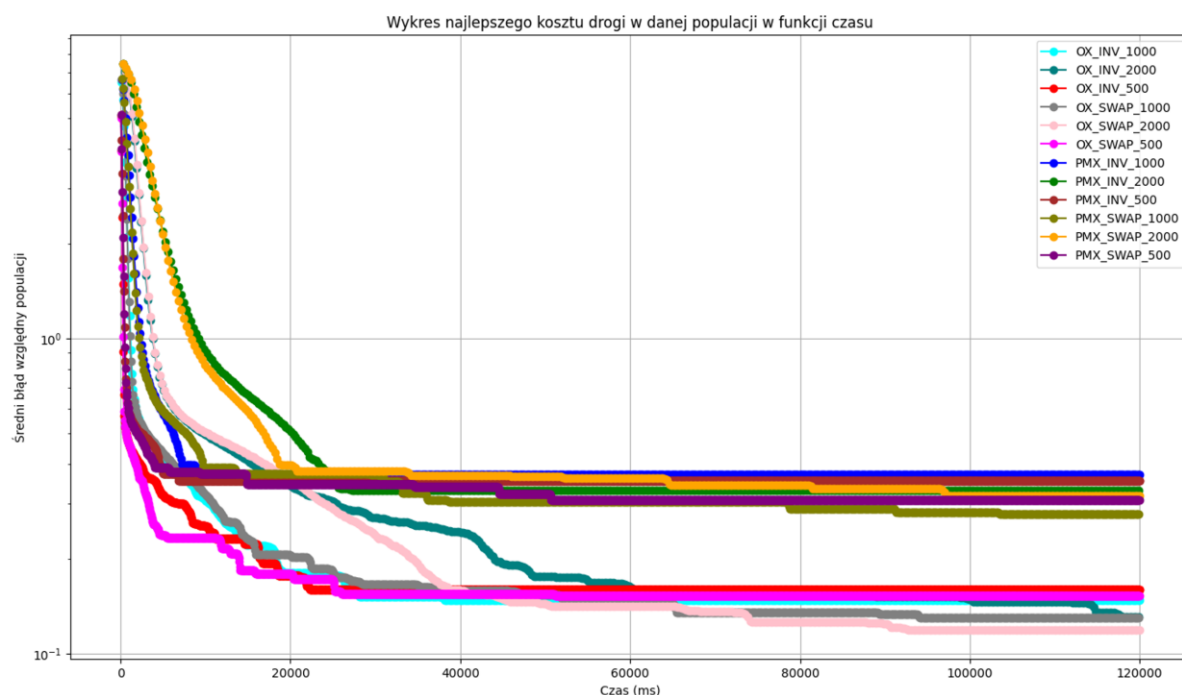
Tabela 1. Tabela zawierająca wyniki badania wpływu wielkości populacji na wyniki.

Nr	Typ	populacja 500		populacja 1000		populacja 2000	
		Najlepszy	Średni błąd	Najlepszy	Średni błąd	Najlepszy	Średni błąd
1	OX SWAP	3443	0.249728	3202	0.16225	3123	0.133575
2		3191	0.158258	3127	0.135027	3178	0.163697
3		3244	0.177495	3260	0.183303	3173	0.151724
4		3341	0.212704	3123	0.133575	3219	0.168421
5		3467	0.258439	3114	0.130309	3089	0.121234
6		3176	0.152813	3192	0.158621	3082	0.118693
7		3203	0.162613	3237	0.174955	3137	0.138657
8		3318	0.204356	3142	0.140472	3153	0.144465
9		3281	0.190926	3146	0.141924	3122	0.133212
10		3294	0.195644	3206	0.163702	3124	0.133938
1	OX INV	3407	0.236661	3208	0.164428	3125	0.134301
2		3358	0.218875	3200	0.161525	3206	0.163702
3		3194	0.159347	3163	0.148094	3172	0.151361
4		3388	0.229764	3257	0.182214	3179	0.153902
5		3575	0.297641	3188	0.157169	3183	0.155354
6		3594	0.304537	3195	0.15971	3115	0.130672
7		3337	0.211252	3291	0.194555	3255	0.181488
8		3326	0.20726	3187	0.156806	3150	0.143376
9		3293	0.195281	3222	0.16951	3211	0.165517
10		3452	0.252995	3380	0.22686	3189	0.157532
1	PMX SWAP	3622	0.314701	3675	0.333938	3629	0.317241
2		3738	0.356806	3674	0.333575	3735	0.355717
3		3663	0.329583	3520	0.277677	3680	0.335753
4		3752	0.372051	3561	0.292559	3635	0.319419
5		3700	0.343013	3746	0.35971	3665	0.330309
6		3666	0.330672	3665	0.330309	3714	0.348094
7		3602	0.307441	3735	0.355717	3668	0.331397
8		3665	0.330309	3676	0.334301	3637	0.320145
9		3626	0.316152	3666	0.330672	3665	0.330309
10		3675	0.333938	3701	0.343376	3752	0.372366
1	PMX INV	3752	0.366606	69121	3764	3752	0.39158

2		3695	0.341198	3719	0.349909	3730	0.353902
3		3675	0.333938	3662	0.370599	3752	0.368784
4		3672	0.352813	3752	0.375681	3737	0.356443
5		3720	0.363702	3728	0.353176	3752	0.37314
6		3740	0.357532	3719	0.349909	3743	0.358621
7		3752	0.375681	3783	0.387296	3691	0.339746
8		3680	0.335753	3730	0.353902	3712	0.347368
9		3752	0.375681	3752	0.364791	3728	0.353176
10		3746	0.380399	3752	0.371325	3663	0.329583



Wykres 1. Wykres średniego błędu względnego populacji w funkcji czasu dla badania wpływu wielkości populacji.



Wykres 2. Wykres średniego błędu względnej populacji w funkcji czasu dla badania wpływu wielkości populacji (skala logarytmiczna).

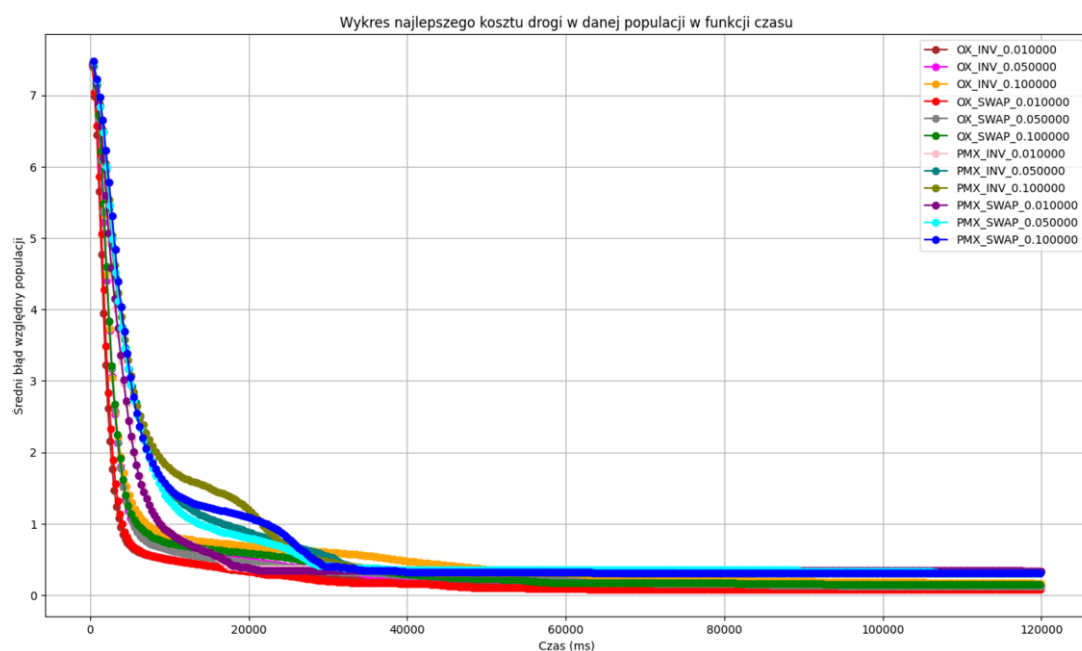
Wyniki badań wpływu współczynnika mutacji

Tabela 2 oraz wykresy 3 i 4 zawierają rezultaty badania wpływu współczynnika mutacji na wyniki algorytmu dla trzech różnych wartości (0.01, 0.05 oraz 0.10). Współczynnik krzyżowania wynosi 0.8, a wielkość populacji - 2000.

Tabela 2. Tabela zawierająca wyniki badania wpływu współczynnika mutacji na wyniki dla populacji 2000.

Nr	Typ	wsp. mutacji 0.01		wsp. mutacji 0.05		wsp. mutacji 0.10	
		Najlepszy	Średni błąd	Najlepszy	Średni błąd	Najlepszy	Średni błąd
1	OX SWAP	3285	0.192377	3151	0.143739	3198	0.160799
2		3139	0.139383	3141	0.141186	3214	0.166606
3		3180	0.154265	3182	0.154991	3204	0.162976
4		3246	0.178221	3176	0.152813	3232	0.173851
5		3236	0.174592	3084	0.123775	3198	0.160799
6		3173	0.151724	3311	0.201815	3168	0.149909
7		2971	0.078403	3092	0.125922	3153	0.148646
8		3111	0.12922	3162	0.147731	3285	0.192377
9		3190	0.157895	3113	0.134299	3233	0.173503
10		3132	0.136842	3121	0.132849	3209	0.164791
1	OX INV	3140	0.139746	3235	0.177484	3232	0.177608
2		3211	0.165517	3157	0.145917	3223	0.169873
3		3228	0.171688	3165	0.14882	3294	0.197584
4		3182	0.154991	3220	0.174939	3209	0.168934
5		3211	0.165517	3177	0.157824	3219	0.168421

6		3172	0.154389	3183	0.155354	3229	0.172051
7		3243	0.177132	3236	0.174592	3404	0.241825
8		3321	0.205445	3180	0.158621	3195	0.15971
9		3257	0.182214	3243	0.177132	3240	0.176044
10		3452	0.252995	3380	0.22686	3189	0.157532
1	PMX SWAP	3716	0.34882	3658	0.327768	3658	0.327768
2		3671	0.332486	3714	0.348094	3665	0.330309
3		3752	0.368058	3618	0.313249	3619	0.313612
4		3752	0.384392	3665	0.330309	3658	0.327768
5		3725	0.352087	3714	0.348094	3658	0.327768
6		3679	0.33539	3752	0.365517	3702	0.343739
7		3684	0.337205	3697	0.341924	3721	0.350635
8		3733	0.354991	3662	0.32922	3749	0.360799
9		3713	0.347731	3665	0.330309	3649	0.324501
10		3684	0.337205	3674	0.333575	3658	0.327768
1	PMX INV	3752	0.377132	3719	0.349909	3620	0.313975
2		3730	0.353902	3752	0.384755	3752	0.413067
3		3714	0.348094	3718	0.349546	3752	0.379673
4		3752	0.391289	3719	0.349909	3752	0.41343
5		3752	0.387296	3752	0.365517	3752	0.36951
6		3752	0.364065	3743	0.358621	3665	0.330309
7		3726	0.35245	3654	0.326316	3688	0.338657
8		3746	0.35971	3752	0.387296	3752	0.384029
9		3752	0.387296	3719	0.349909	3639	0.320871
10		3710	0.346642	3625	0.315789	3752	0.384392



Wykres 3. Wykres średniego błędu względny populacji w funkcji czasu dla badania wpływu współczynnika mutacji.



Wykres 4. Wykres średniego błędu względnej populacji w funkcji czasu dla badania wpływu współczynnika mutacji (skala logarytmiczna).

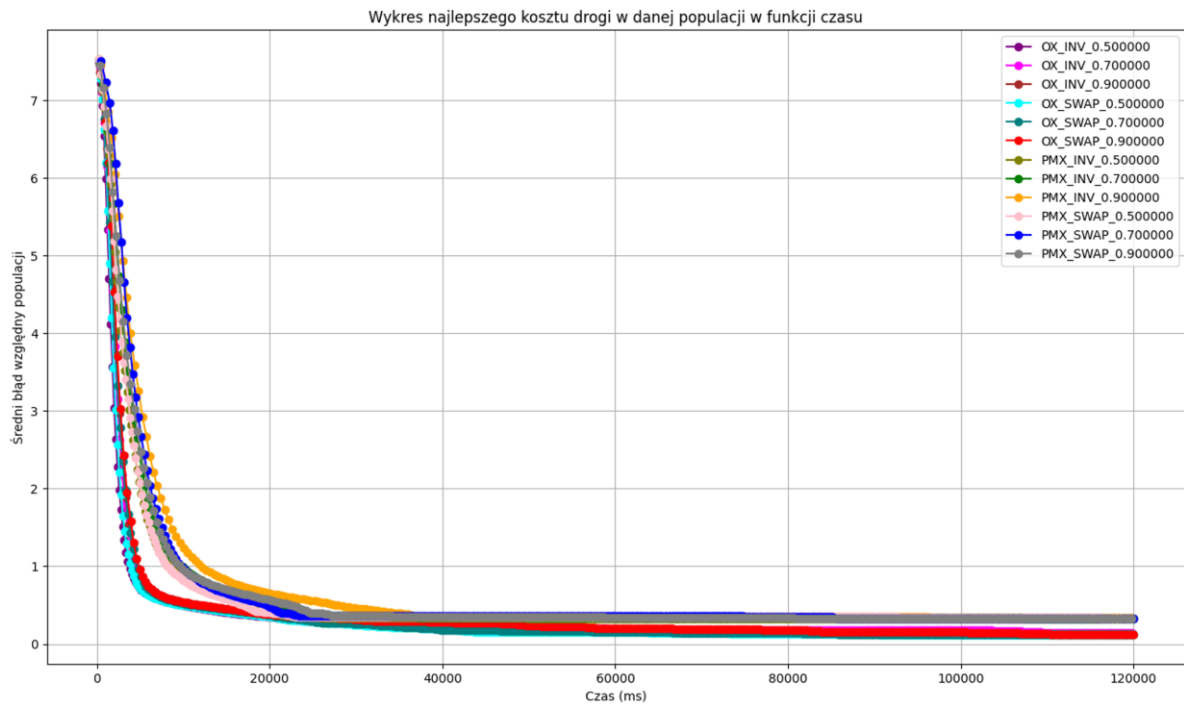
Wyniki badań wpływu współczynnika krzyżowania

Tabela 3 oraz wykresy 5 i 6 zawierają rezultaty badania wpływu współczynnika krzyżowania na wyniki algorytmu dla trzech różnych wartości (0.5, 0.7 oraz 0.9). Współczynnik mutacji wynosi 0.01, a wielkość populacji - 2000.

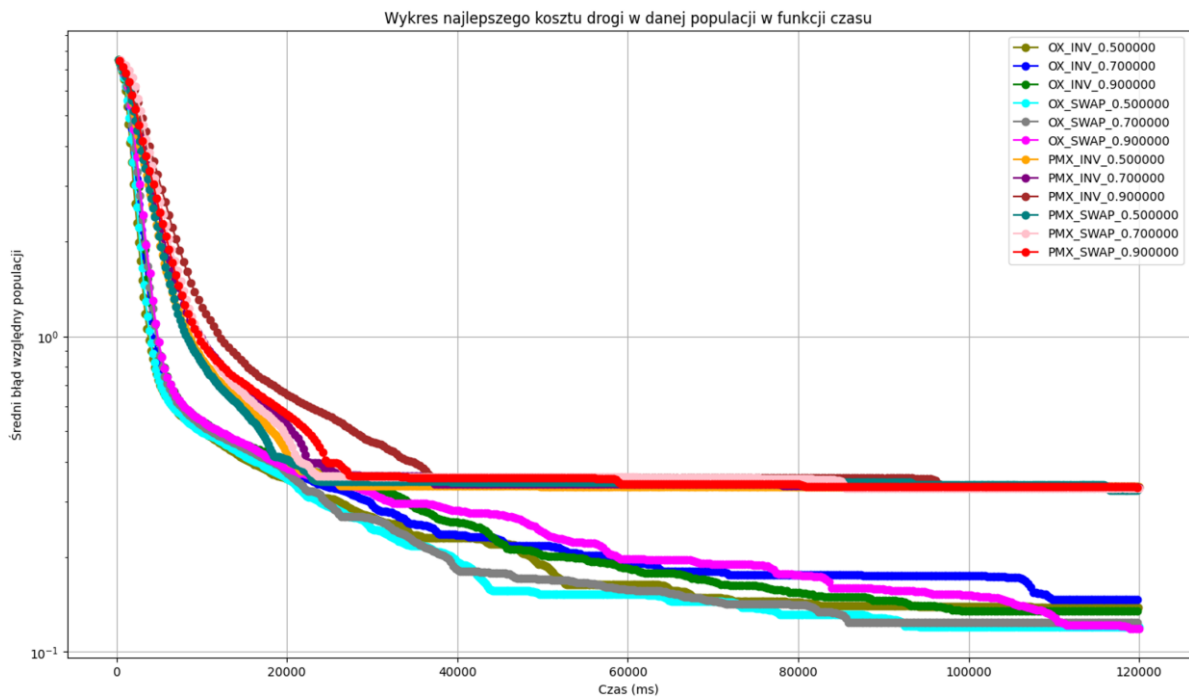
Tabela 3. Tabela zawierająca wyniki badania wpływu współczynnika krzyżowania na wyniki dla populacji 2000.

		wsp. krzyżowania 0.5		wsp. krzyżowania 0.7		wsp. krzyżowania 0.9	
Nr	Typ	Najlepszy	Średni błąd	Najlepszy	Średni błąd	Najlepszy	Średni błąd
1	OX SWAP	3127	0.135027	3108	0.128131	3180	0.154265
2		3085	0.119782	3183	0.155354	3144	0.145236
3		3127	0.135027	3200	0.161525	3221	0.174575
4		3115	0.130672	3167	0.149546	3223	0.169873
5		3138	0.13902	3128	0.13539	3187	0.156806
6		3191	0.158258	3173	0.151724	3204	0.164054
7		3209	0.169508	3160	0.147005	3078	0.11833
8		3120	0.132486	3170	0.150635	3279	0.1902
9		3144	0.141198	3266	0.199949	3142	0.140472
10		3128	0.13539	3096	0.123775	3098	0.124501
1	OX INV	3221	0.169147	3160	0.147005	3282	0.191289
2		3135	0.137931	3190	0.157895	3174	0.152087
3		3234	0.177131	3183	0.155354	3118	0.13415
4		3147	0.142287	3254	0.182197	3163	0.148094

5		3169	0.150272	3188	0.157169	3119	0.135911
6		3202	0.16225	3190	0.157895	3234	0.173866
7		3182	0.154991	3200	0.161525	3191	0.171594
8		3256	0.181851	3221	0.169147	3206	0.163702
9		3203	0.162613	3269	0.18657	3173	0.152309
10		3156	0.145554	3177	0.153176	3273	0.188022
1	PMX SWAP	3688	0.338657	3667	0.331034	3715	0.349198
2		3671	0.34282	3665	0.330309	3678	0.341561
3		3649	0.324501	3723	0.351361	3685	0.337568
4		3701	0.343376	3674	0.333575	3752	0.370962
5		3704	0.344465	3717	0.349183	3740	0.357532
6		3752	0.363339	3727	0.366956	3752	0.380396
7		3693	0.340472	3752	0.373866	3739	0.357169
8		3735	0.355717	3752	0.374955	3702	0.344827
9		3696	0.341561	3725	0.352087	3665	0.330309
10		3688	0.338657	3671	0.332486	3693	0.340472
1	PMX INV	3752	0.387296	3752	0.362976	3682	0.336479
2		3752	0.364065	3752	0.387296	3752	0.377125
3		3688	0.338657	3752	0.36225	3752	0.379298
4		3752	0.365154	3752	0.377129	3677	0.334664
5		3752	0.380399	3709	0.346279	3684	0.337205
6		3719	0.349909	3670	0.332123	3730	0.353902
7		3752	0.370599	3719	0.349909	3752	0.370599
8		3752	0.37314	3752	0.387296	3752	0.36588
9		3752	0.376407	3690	0.339383	3752	0.37314
10		3673	0.333212	3752	0.369147	3736	0.35608



Wykres 5. Wykres średniego błędu względnego populacji w funkcji czasu dla badania wpływu współczynnika krzyżowania.



Wykres 6. Wykres średniego błędu względnego populacji w funkcji czasu dla badania wpływu współczynnika krzyżowania (skala logarytmiczna).

Podsumowanie

Badanie wpływu wielkości populacji

Analizując wyniki działania algorytmu genetycznego, można zauważyć, że różne metody krzyżowania i mutacji dają różne rezultaty w zależności od wielkości populacji. W przypadku metody OX SWAP wyniki są wyraźnie lepsze przy większych populacjach. Dla populacji liczącej 500 osobników średni wynik oscyluje w granicach 3,200–3,400, a średni błąd wynosi od 0,15 do 0,25. Dla populacji 1000

osobników wyniki są bardziej stabilne i lepsze, z zakresami 3,114–3,260 dla najlepszych rozwiązań i 0,13–0,17 dla błędów. Najlepsze efekty w tej metodzie osiągnięto przy populacji 2000, gdzie koszt rozwiązania wynosił 3,082–3,173, a średni błąd obniżył się do 0,12–0,16. OX SWAP okazał się szczególnie skuteczny w dużych populacjach, co wskazuje na jego zdolność do eksplorowania przestrzeni rozwiązań.

Dla metody OX INV wyniki były porównywalne do OX SWAP, szczególnie w przypadku średnich i dużych populacji. Dla populacji 500 najlepsze wyniki oscylowały w granicach 3,194–3,594, a średni błąd wynosił od 0,15 do 0,30. W populacji 1000 osobników zauważono poprawę wyników, gdzie najlepsze rozwiązania znajdowały się w przedziale 3,163–3,388, a średni błąd obniżył się do zakresu 0,14–0,19. Przy populacji 2000 wyniki były zbliżone do OX SWAP, z najlepszym rozwiązaniem wynoszącym 3,115 i średnim błędem na poziomie 0,13. OX INV, podobnie jak OX SWAP, działał dobrze przy większych populacjach, osiągając stabilne i niskie błędy.

Z kolei metoda PMX SWAP charakteryzowała się wyższymi kosztami rozwiązań i większymi błędami niż OX. W populacji liczącej 500 osobników najlepsze wyniki znajdowały się w przedziale 3,622–3,752, a średni błąd wynosił od 0,31 do 0,37. Przy populacji 1000 osobników wyniki nieco się poprawiły, osiągając zakres 3,520–3,746 dla najlepszych rozwiązań, przy średnim błędzie wynoszącym około 0,27–0,35. Populacja 2000 osobników przyniosła jeszcze lepsze wyniki, ale średni błąd pozostał wysoki, wahając się od 0,31 do 0,37. Wskazuje to na mniejszą skuteczność tej metody w porównaniu do OX SWAP i OX INV.

Metoda PMX INV okazała się najmniej skuteczna spośród analizowanych technik. Przy populacji 500 osobników wyniki wahały się w przedziale 3,672–3,752, a błędy wynosiły od 0,33 do 0,38. W populacji 1000 osobników pojawił się wyraźnie odstający wynik (69,121), co sugeruje możliwość błędu w obliczeniach. Pozostałe wyniki były bardziej stabilne i oscylowały wokół 3,700–3,750. W populacji 2000 osobników wyniki były nieco lepsze, z zakresem 3,663–3,752, ale błędy nadal były stosunkowo wysokie, wynosząc od 0,33 do 0,39.

Podsumowując, metody OX SWAP i OX INV wykazały się najwyższą skutecznością, szczególnie w przypadku większych populacji, gdzie zarówno koszty rozwiązań, jak i średnie błędy były najniższe. Z kolei PMX SWAP i PMX INV osiągały wyższe koszty oraz większe błędy, co wskazuje na ich mniejszą efektywność. Wyniki wyraźnie pokazują, że większe populacje poprawiają stabilność i jakość wyników.

Badanie wpływu współczynnika mutacji

Dla metody OX SWAP zauważono, że zwiększenie współczynnika mutacji (od 0,01 do 0,10) przynosi pewną poprawę wyników, choć średnie błędy pozostają stosunkowo stabilne w zakresie od około 0,123 do 0,192. Najlepszy wynik dla współczynnika mutacji 0,01 wynosi 2971, przy czym błąd średni to 0,0784, co wskazuje na dobrą jakość rozwiązania. Dla mutacji 0,05 najlepszy wynik poprawił się do 3084 z błędem średnim wynoszącym 0,1238, natomiast przy współczynniku mutacji 0,10 najlepszy wynik wynosił 3153, z błędem średnim 0,1486. Wskazuje to na większą stabilność tej metody przy wyższych współczynnikach mutacji.

Metoda OX INV charakteryzuje się podobnymi wynikami, choć dla współczynnika mutacji 0,01 najlepszy wynik to 3140, a średni błąd wynosi 0,1397. Przy współczynniku 0,05 wynik poprawił się do 3235, z błędem 0,1775, a dla współczynnika 0,10 wynosił 3232, z błędem 0,1776. OX INV

pokazuje tendencję do większych błędów w porównaniu do OX SWAP, szczególnie przy większych współczynnikach mutacji.

W przypadku metody PMX SWAP, średnie błędy były wyraźnie wyższe w całym zakresie współczynników mutacji. Dla mutacji 0,01 najlepszy wynik wynosił 3671 przy średnim błędzie 0,3325. Przy współczynniku 0,05 wynik poprawił się do 3618, z błędem wynoszącym 0,3132. Dla współczynnika 0,10 najlepszy wynik wynosił 3649, a średni błąd obniżył się do 0,3245. Mimo tych zmian PMX SWAP pozostaje mniej efektywny niż OX.

Metoda PMX INV okazała się najmniej efektywna, z najwyższymi średnimi błędami w każdym przypadku. Dla współczynnika mutacji 0,01 najlepszy wynik wynosił 3710, a średni błąd to 0,3466. Dla współczynnika 0,05 najlepszy wynik poprawił się do 3625, przy średnim błędzie wynoszącym 0,3158. Przy współczynniku 0,10 najlepszy wynik wynosił 3620, z błędem średnim 0,3139. Wskazuje to, że PMX INV nie radzi sobie dobrze z różnymi wartościami współczynnika mutacji.

Podsumowując, metoda OX SWAP wykazuje najwyższą skuteczność przy wszystkich wartościach współczynnika mutacji, z niskimi błędami i najlepszymi wynikami. OX INV jest blisko, choć ma tendencję do wyższych błędów. PMX SWAP i PMX INV są mniej skuteczne, z wyraźnie wyższymi średnimi błędami i gorszymi wynikami. Współczynnik mutacji na poziomie 0,05 wydaje się najlepszym kompromisem pomiędzy jakością rozwiązań a stabilnością.

Badanie wpływu współczynnika krzyżowania

Analiza danych pokazuje, że różne metody algorytmu genetycznego różnie reagują na zmiany współczynnika krzyżowania. W przypadku metody OX SWAP najlepsze rezultaty osiągnięto przy współczynniku 0.7, gdzie najniższe średnie błędy (np. 0.123 w jednej z prób) wskazują na efektywne dopasowanie i stabilność. Dla współczynnika 0.5 oraz 0.9 wyniki były mniej spójne, co sugeruje, że metoda ta najlepiej działa w umiarkowanym zakresie krzyżowania.

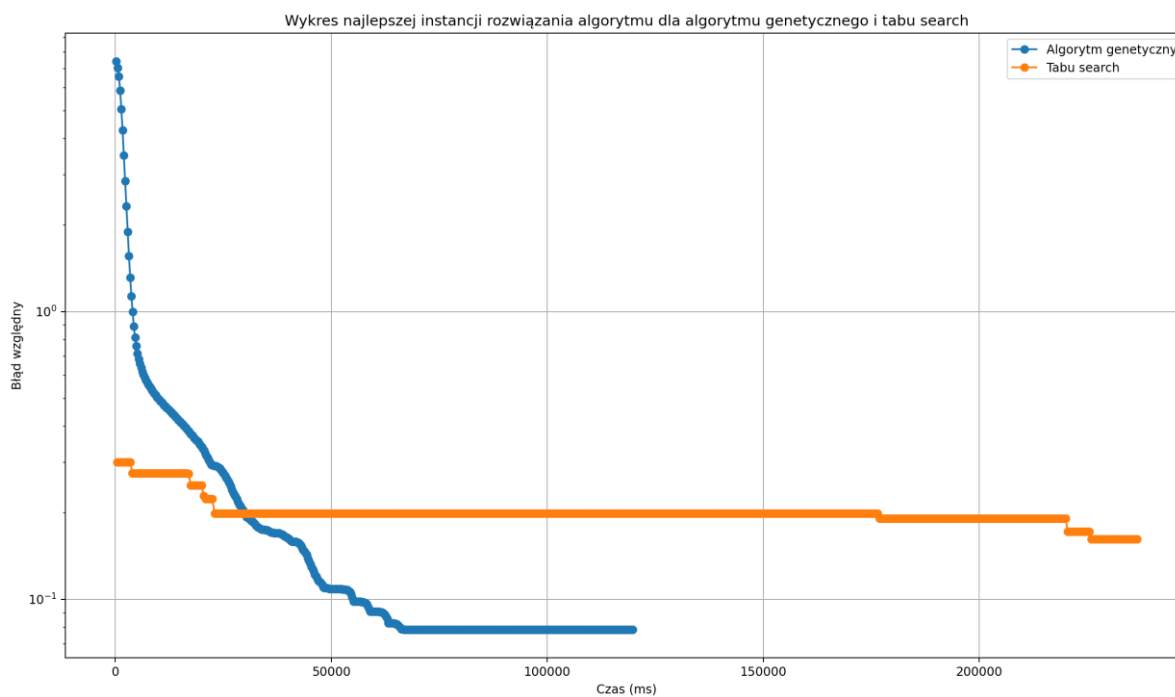
Metoda OX INV również osiąga najlepsze wyniki przy współczynniku 0.7. Błędy są tu wyraźnie niższe niż przy współczynnikach 0.5 i 0.9, co oznacza, że ta konfiguracja lepiej eksploruje przestrzeń rozwiązań. Jednak metoda OX SWAP wciąż jest bardziej stabilna i skuteczna w porównaniu do OX INV.

W przypadku metod PMX SWAP i PMX INV wyniki były mniej imponujące. Metoda PMX SWAP, niezależnie od współczynnika krzyżowania, osiągała wyższe średnie błędy (np. 0.330 przy współczynniku 0.7) w porównaniu do metod OX. Podobnie PMX INV wykazała małą wrażliwość na zmiany współczynnika krzyżowania, ale utrzymywała się na najniższym poziomie efektywności we wszystkich próbach.

Podsumowując, metoda OX SWAP przy współczynniku krzyżowania 0.7 okazała się najbardziej efektywna, oferując najlepszy balans między jakością rozwiązań a ich stabilnością. Współczynnik krzyżowania ma większy wpływ na metody OX niż na PMX, a PMX INV wypadła najgorzej w analizie, wskazując na problemy z optymalizacją w każdej konfiguracji.

Porównanie z Tabu Search

Wykres 7 pokazuje porównanie najlepszych wyników uzyskanych przez algorytm Tabu Search i Algorytm Genetyczny. Poniżej znajdują się również uzyskane ścieżki. Uwaga. Błąd względy w przypadku Algorytmu Genetycznego dotyczy średniego błędu całej populacji, natomiast w przypadku Tabu Search jest to błąd względny danej instancji.



Wykres 7. Wykres porównania błędu względnego w funkcji czasu najlepszych wyników Algorytmu Genetycznego i Tabu Search (skala logarymiczna).

Na wykresie można zauważyć, że algorytm genetyczny o wiele szybciej znajduje lepsze rozwiązanie i wyraźnie sprawniej przeszukuje przestrzeń rozwiązań problemu, mimo że Tabu Search wykonuje się dwa razy dłużej.

Tabu Search ftv170 – najlepsza droga: 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

Algorytm Genetyczny ftv170 – najlepsza droga: 2971

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

Bibliografia

- Z.Michalkiewicz, D.B.Vogel, „Jak to rozwiązać czyli nowoczesna heurystyka”, WNT 2006
- <https://medium.com/@becmjo/genetic-algorithms-and-the-travelling-salesman-problem-d10d1daf96a1>
- https://www.youtube.com/watch?v=M3KTWnTrU_c&ab_channel=TheCodingTrain
- <https://mat.uab.cat/~alseda/MasterOpt/GeneticOperations.pdf>
- https://www.youtube.com/watch?v=EZg-l2FF-JM&ab_channel=DEEBAKANNAN
- <https://www.aragorn.wi.pb.edu.pl/~wkwedlo/EA5.pdf>