

Zadanie projektowe nr 2.

Implementacja i analiza efektywności algorytmu Tabu Search oraz Symulowanego Wyżarzania dla problemu komiwojażera.

Sprawozdanie z zadania projektowego przedmiotu „Projektowanie efektywnych algorytmów”.

Rok akademicki 2019/2020, kierunek Informatyka.

Prowadzący:

Dr inż. Zbigniew Buchalski

Termin zajęć:

Wtorek, 9.15

1. Wstęp teoretyczny

Celem drugiego zadania projektowego była implementacja oraz analiza efektywności algorytmów opartych o przeszukiwanie lokalne. Zadanyymi algorytmami było przeszukiwanie z zakazami (**Tabu Search**) oraz symulowane wyżarzanie (**Simulated Annealing**). Są to algorytmy heurystyczne, a więc pozwalające znaleźć dobre rozwiązanie problemu przy akceptowalnych nakładach obliczeniowych, bez gwarancji optymalności rozwiązania oraz bez określenia jego jakości, czyli odległości od optimum globalnego. Cechą wspólną obu z implementowanych algorytmów jest to iż stanowią one rozwinięcie przeszukiwania lokalnego. Podczas wykonywania, każdy z algorytmów przeszukuje pewne obszary całej przestrzeni rozwiązań, zwane dalej sąsiedztwami danego rozwiązania. Na początku wybierane jest losowe rozwiązanie, którego sąsiedztwo będzie przeglądane w dalszych krokach algorytmu. Sąsiednim rozwiązaniem nazywane jest rozwiązanie różniące się w określony sposób od rozwiązania początkowego. Zbiór wszystkich możliwych do uzyskania rozwiązań sąsiednich dla danego rozwiązania początkowego jest nazywany sąsiedztwem tego rozwiązania. Badane algorytmy różni sposób przeglądania sąsiedztwa aż do osiągnięcia kryterium stopu.

Główną ideą przeszukiwania z zakazami jest dla danego rozwiązania startowego wybór takiego sąsiada dla którego wartość funkcji oceny ruchu (określenie opłacalności przyjęcia rozwiązania sąsiedniego jako bieżącego) jest najwyższa. W przypadku problemu komiwojagera, funkcją tą jest różnica długości ścieżki rozwiązania bieżącego i sąsiedniego. Konsekwencją tego jest konieczność każdorazowego przeszukiwania całego sąsiedztwa danego losowego rozwiązania celem znalezienia najlepszego sąsiada, stąd podejście to zwane jest stromym (**Steepest**). Istnieje zatem możliwość iż w kolejnej iteracji algorytmu może zostać wykonany ten sam ruch oferujący poprawienie rozwiązania, jednakże tylko w obszarze lokalnym, prowadząc do „utknięcia” algorytmu w tym obszarze i ograniczenia eksploracji przestrzeni wszystkich rozwiązań (zmniejszenia **dywersyfikacji**) a co za tym idzie obszarów, w których może znajdować się optimum globalne. Efekt ten może zostać złagodzony dzięki zastosowaniu **Listy Tabu**, czyli listy wykonanych ruchów, zabronionych w określonej ilości iteracji (**kadencji**) przeglądu sąsiedztwa. Możliwe jest, iż ruch znajdujący się na liście tabu prowadzi do rozwiązania lepszego niż najlepsze dotychczasowe (jest ono zapamiętywane). Spełnia ono wtedy tzw. **kryterium Aspiracji** i może zostać wykorzystane pomimo tego że znajduje się na liście tabu.

Drugim zaimplementowanym algorytmem było symulowane wyżarzanie (**Simulated Annealing**). Podobnie jak w przypadku **Tabu Search**, jego działanie polega na badaniu różnych obszarów przestrzeni rozwiązań, jednakże zastosowana została inna strategia wyboru rozwiązania sąsiedniego. W odróżnieniu od **Tabu Search**, zamiast wyboru najlepszego rozwiązania znajdującego się w sąsiedztwie, jako rozwiązanie bieżące w danym pokoleniu przyjmowany jest każdy ruch prowadzący do poprawy rozwiązania bieżącego. Stąd podejście to nazywane jest zachłannym (**Greedy**). Ponadto, możliwym jest przyjęcie rozwiązania gorszego ponieważ w jego sąsiedztwie może znajdować się o wiele lepsze a być może globalnie optymalne. Jest to realizowane dzięki zastosowaniu mechanizmu przypominającego proces wyżarzania metalu. Jego działanie polega na stopniowym ograniczaniu przyjmowania gorszych rozwiązań podczas badania sąsiedztwa danego rozwiązania początkowego. Stanowi to analogię do zmniejszającej się wraz z temperaturą ruchliwości cząsteczek wyżarzanego metalu. Podczas

przeoglądania sąsiedztwa, gorsze rozwiązanie jest przyjmowane z pewnym prawdopodobieństwem. Parametrem algorytmu, który ma wpływ na te prawdopodobieństwo jest parametr przeniesiony bezpośrednio z podstaw termodynamicznych algorytmu, czyli temperatura. Im wyższa, tym prawdopodobieństwo wyboru gorszego rozwiązania jest większe (**większa dywersyfikacja**). Im niższa, tym algorytm jest bardziej zbliżony w działaniu do typowych metod iteracyjnych (**większa intensyfikacja**). Na początku działania algorytmu temperatura jest wysoka, dzięki czemu algorytm może bardzo często zmieniać konfigurację rozwiązania, niejednokrotnie wybierając rozwiązanie gorsze. Wraz z kolejnymi iteracjami algorytmu temperatura spada i wybierane są częściej rozwiązania lepsze. Pod koniec pracy algorytmu, temperatura jest na tyle niska, że prawdopodobieństwo wyboru gorszego rozwiązania jest bliskie zeru. Algorytm zachowuje się wówczas, jak typowy algorytm iteracyjny i stara się maksymalnie ulepszyć rozwiązanie.

3. Plan eksperymentu.

Badanie efektywności zaimplementowanych algorytmów polegało na obliczeniu błędów względnych w funkcji czasu ich wykonywania dla trzech instancji problemu komiwojażera. Dla każdej instancji problemu, kryterium stopu zostało określone jako dopuszczalny czas wykonywania algorytmu i został on uzależniony od wielkości instancji. Dla pliku ftv47.atsp były to 2 minuty, ftv170.atsp 4 minuty oraz dla rbg403.atsp 6 minut. Dla obydwu algorytmów określono taki sam rodzaj sąsiedztwa, powstającego poprzez zamianę miejscami dwóch wierzchołków w rozwiązaniu początkowym. Ponadto, dla Symulowanego Wyżarzania należało dobrać współczynnik obniżania temperatury oraz określić funkcję obliczającą temperaturę początkową w zależności od danych wejściowych.

Obliczanie temperatury początkowej odbywa się poprzez 10 000 krotne próbkowanie przestrzeni rozwiązań. Próbkowanie rozumiane jest jako wygenerowanie losowego rozwiązania oraz jego sąsiada i obliczenie różnicy w jakości uzyskanych w ten sposób rozwiązań. Na podstawie pobranych próbek określana jest średnia wartość różnicy która następnie podstawiana jest do wzoru na temperaturę początkową:

$$T = \frac{-\delta}{\log(0,99)} \quad \text{gdzie } -\delta \text{ to obliczona średnia różnica}$$

Współczynnik zmiany temperatury został na potrzeby badań ustawiony na 0,99 gdyż ta wartość pozwalała uzyskać najlepsze rozwiązania dla każdej z instancji.

4. Analiza danych pomiarowych.

Poniżej zestawione zostały wyniki analizy efektywności algorytmów wraz z odpowiednimi wykresami błędu względnego w funkcji czasu wykonywania.

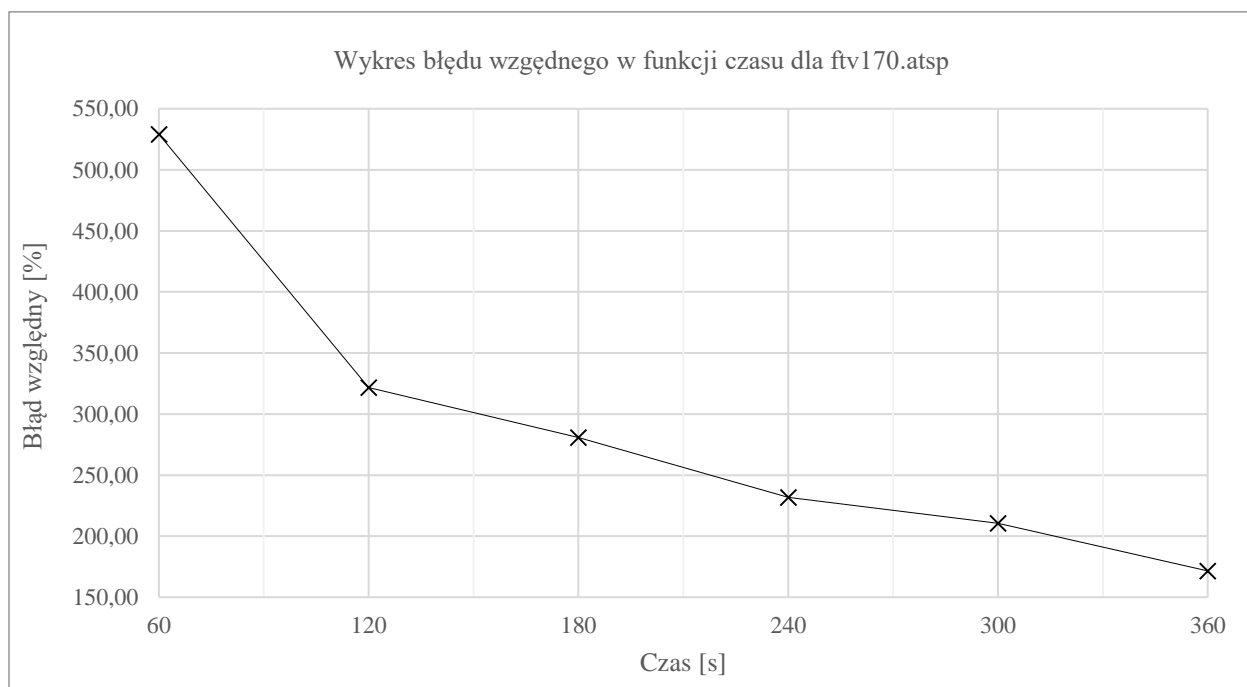
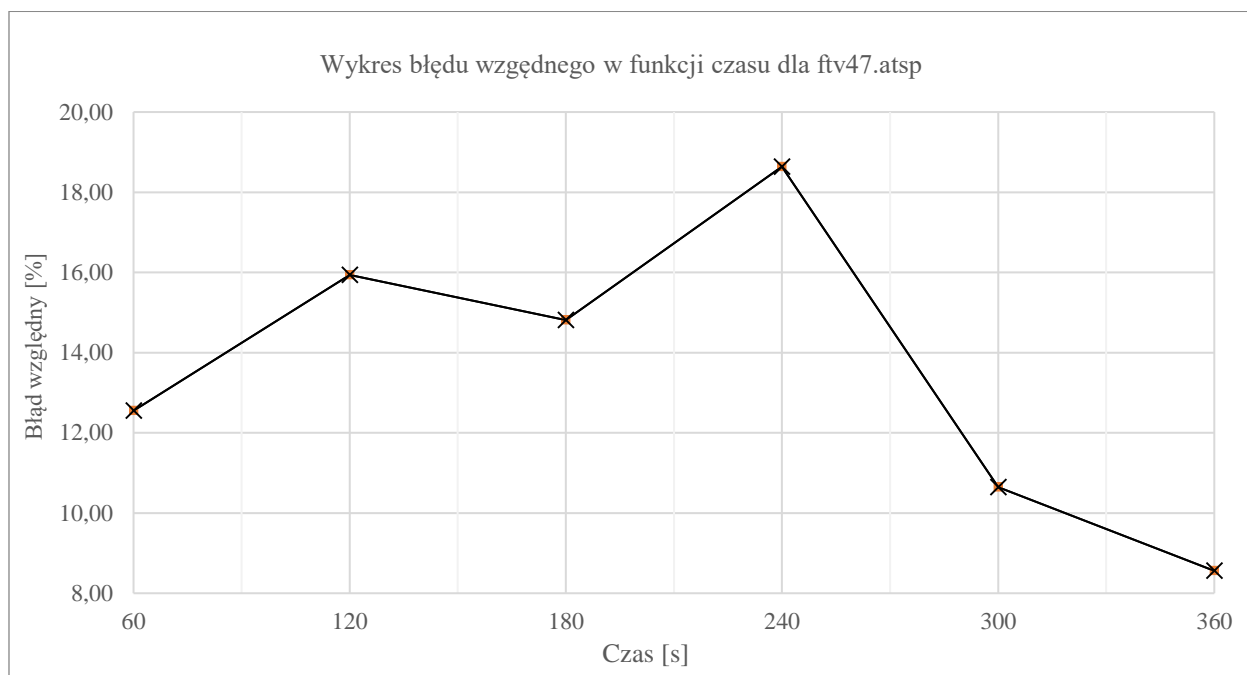
4.1. Przeszukiwanie z zakazami (Tabu Search)

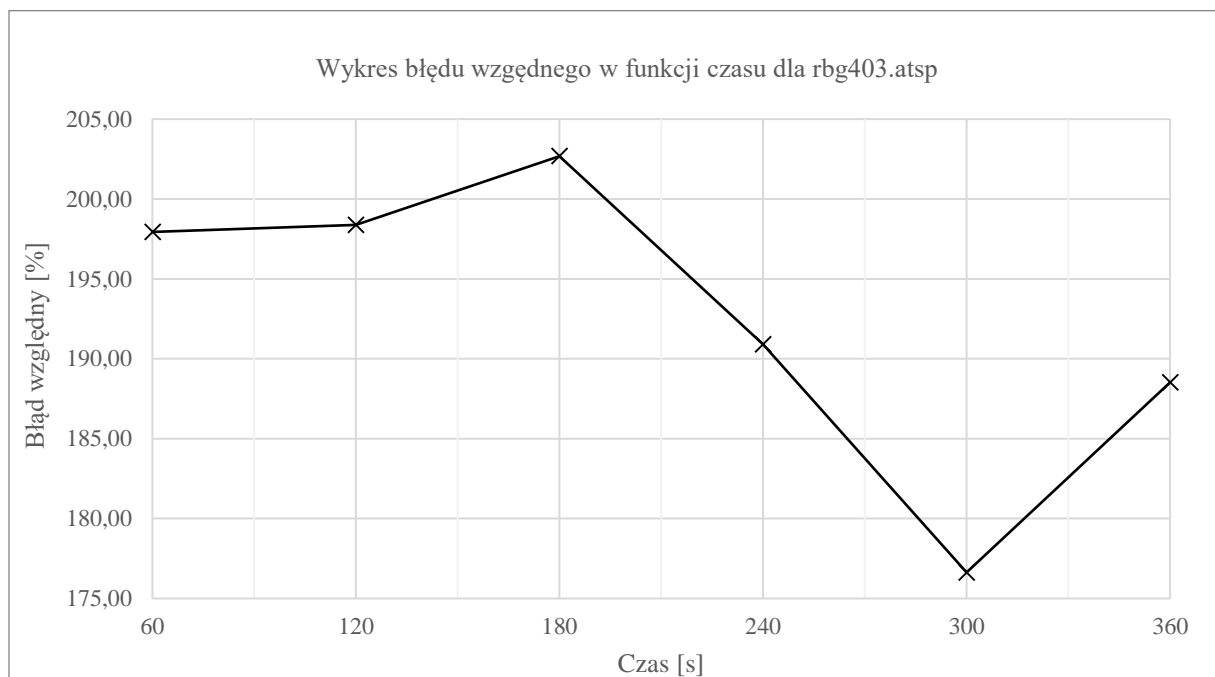
Tabela 1. Zestawienie błędów względnych oraz czasów odnalezienia

Plik	ftv47.atsp			ftv170.atsp			rbg403.atsp		
Nr	Rozw.	Błąd [%]	Czas [s]	Rozw.	Błąd [%]	Czas [s]	Rozw.	Błąd [%]	Czas [s]
1	2026	14,08	112,981	9059	228,82	238,104	7012	184,46	359,911
2	1967	10,75	116,807	9353	239,49	239,498	6863	178,42	359,263
3	2161	21,68	11,915	8724	216,66	239,83	6913	180,45	353,539
4	2167	22,02	113,92	9430	242,29	239,863	7105	188,24	347,097
5	2221	25,06	119,078	8559	210,67	239,896	6807	176,15	354,661
6	1886	6,19	117,514	8603	212,27	239,219	6900	179,92	356,131
7	2021	13,80	108,406	8891	222,72	239,571	7153	190,18	341,646
8	2068	16,44	115,769	8539	209,95	239,897	6902	180,00	359,982
9	2133	20,10	109,432	9077	229,47	239,548	6678	170,91	354,81
10	1999	12,56	110,643	9129	231,36	238,286	7064	186,57	359,18

Tabela 2. Zestawienie błędów względnych oraz maksymalnego dopuszczalnego czasu

Plik	ftv47.atsp		ftv170.atsp		rbg403.atsp	
Czas [s]	Rozw.	Błąd [%]	Rozw.	Błąd [%]	Rozw.	Błąd [%]
60	1999	12,56	17334	529,18	7344	197,93
120	2059	15,93	11619	321,74	7355	198,38
180	2039	14,81	10488	280,69	7461	202,68
240	2107	18,64	9141	231,80	7171	190,91
300	1965	10,64	8558	210,64	6819	176,63
360	1928	8,56	7481	171,54	7112	188,52





4.2. Symulowane Wyżarzanie (Simulated Annealing)

Tabela 3. Zestawienie błędów względnych oraz czasów odnalezienia

Plik	ftv47.atsp			
Nr	Rozw.	Błąd [%]	Czas [s]	Tk
1	2690	51,46	0,283	266,450
2	2585	45,55	0,437	1929,570
3	2274	28,04	0,413	36,058
4	2628	47,97	0,564	3,907
5	2604	46,62	0,344	14,823
6	2685	51,18	0,332	5,508
7	2529	42,40	0,285	11,399
8	2675	50,62	0,361	1,163
9	2723	53,32	0,267	45,038
10	2431	36,88	0,219	11,870

Tabela 4. Zestawienie błędów względnych oraz czasów odnalezienia

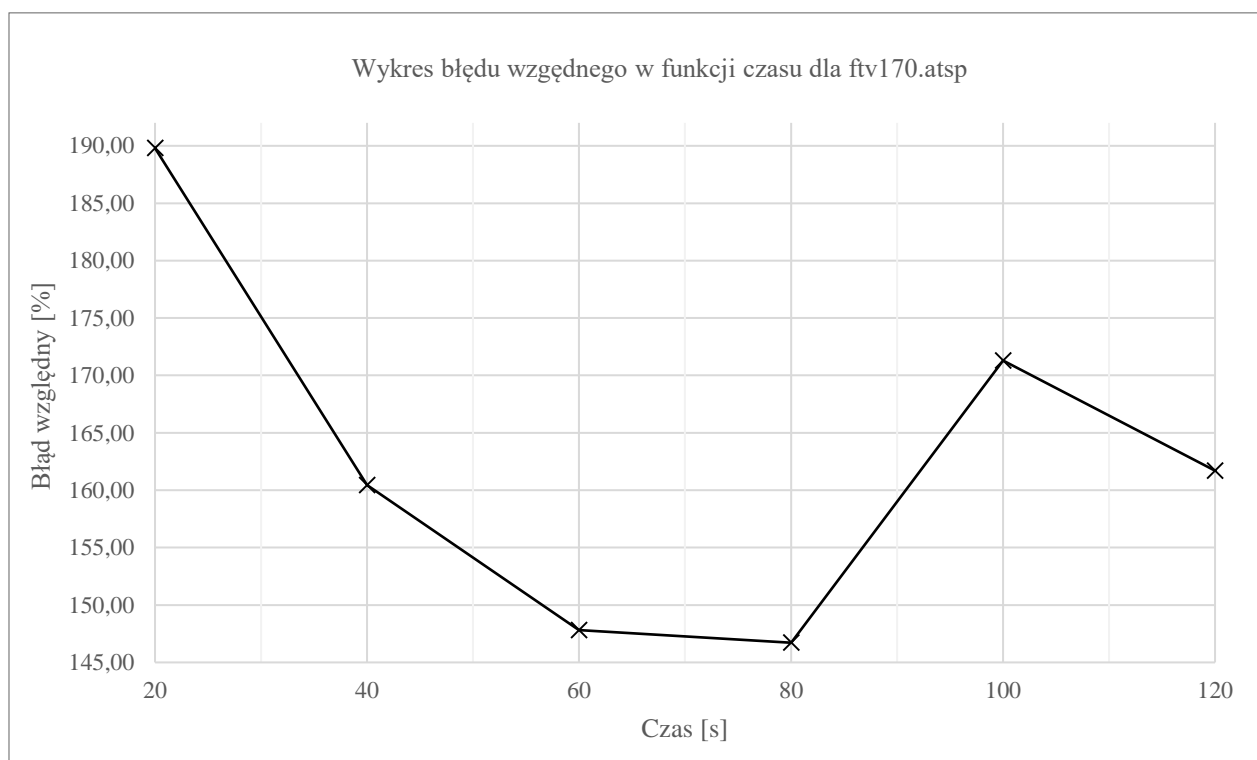
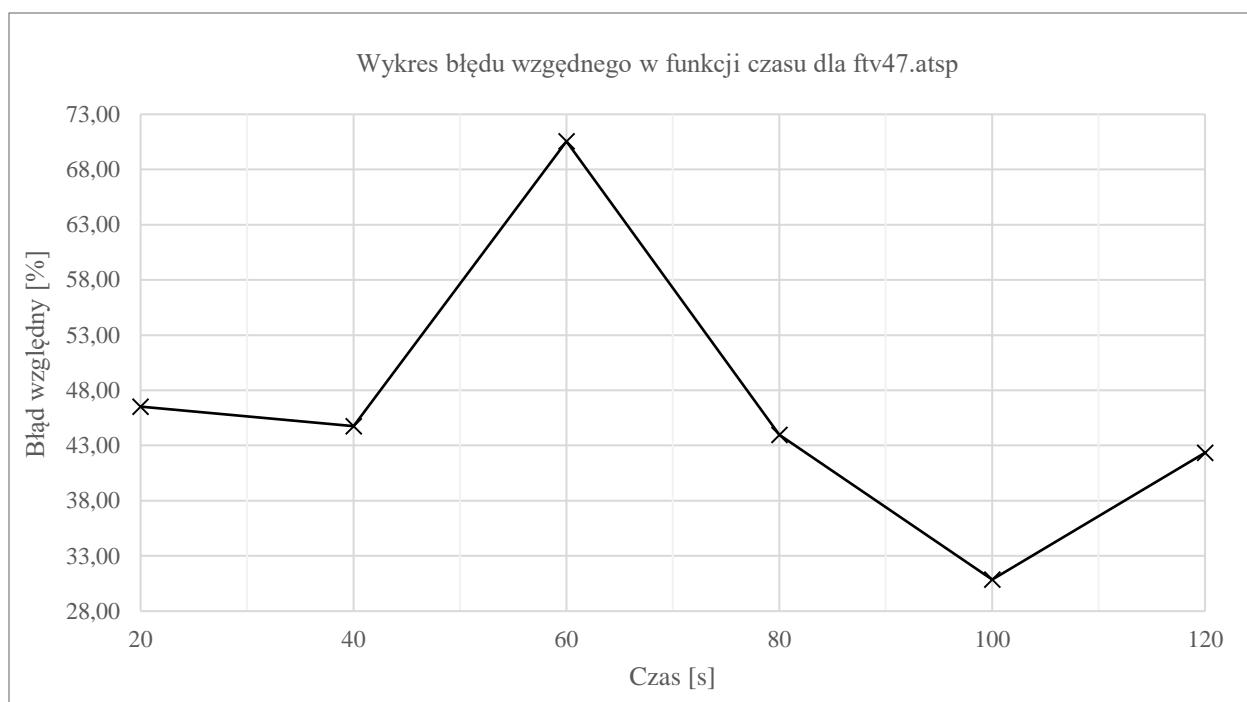
Plik	ftv170.atsp			
Nr	Rozw.	Błąd [%]	Czas [s]	Tk
1	7446	170,27	20,168	6,936
2	7457	170,67	13,426	3,219
3	7164	160,04	25,432	5,554
4	6523	136,77	13,792	463,560
5	7078	156,91	25,501	4,271
6	7047	155,79	14,755	2,064
7	6703	143,30	18,927	0,737
8	6761	145,41	13,980	4,020
9	7294	164,75	14,775	7,370
10	7257	163,41	17,917	18,666

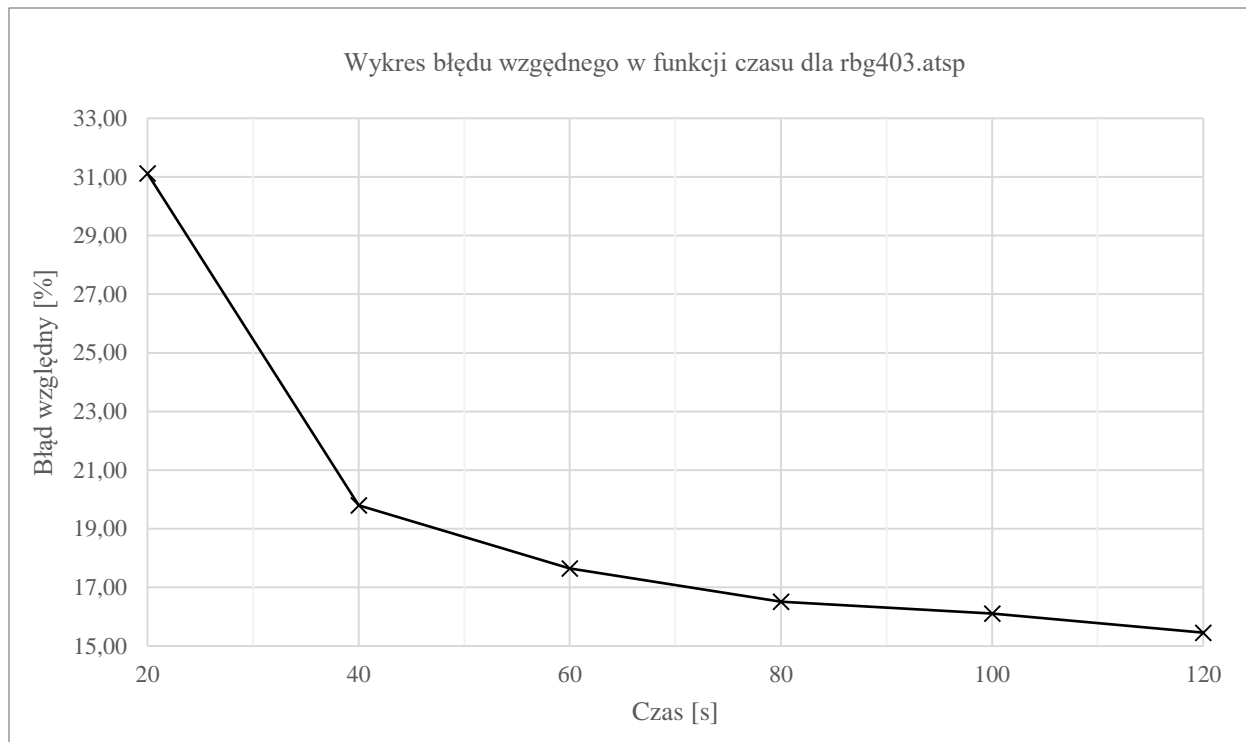
Tabela 5. Zestawienie błędów względnych oraz czasów odnalezienia

Plik	rbg403.atsp			
Nr	Rozw.	Błąd [%]	Czas [s]	Tk
1	2840	15,21	114,478	0,107
2	2855	15,82	114,866	1272,520
3	2839	15,17	107,327	1162,470
4	2832	14,89	116,300	1379,060
5	2854	15,78	118,959	0,120
6	2772	12,45	116,878	1365,270
7	2792	13,27	118,843	0,141
8	2765	12,17	116,705	1311,470
9	2809	13,96	117,845	0,109
10	2804	13,75	112,464	0,155

Tabela 6. Zestawienie błędów względnych oraz maksymalnego dopuszczalnego czasu

Plik	ftv47.atsp		ftv170.atsp		rbg403.atsp	
Czas [s]	Rozw.	Błąd [%]	Rozw.	Błąd [%]	Rozw.	Błąd [%]
20	2602	46,51	7984	189,80	3232	31,12
40	2571	44,76	7175	160,44	2953	19,80
60	3029	70,55	6827	147,80	2900	17,65
80	2557	43,98	6797	146,72	2872	16,51
100	2324	30,86	7474	171,29	2862	16,11
120	2528	42,34	7210	161,71	2846	15,46





5. Wnioski

Analiza wyników uzyskanych podczas przeprowadzonych badań pozwala w przypadku obydwu algorytmów zauważyć iż znaczącym czynnikiem wpływającym na jakość dostarczonych rozwiązań jest ilość czasu przez który mogą one działać. Zjawisko to można zaobserwować na niektórych wykresach błędu względnego. Nieregularność którą można zauważyć na pozostałej części wykresów wynika ze zbyt krótkich bądź zbyt długich przedziałów czasu dla danej instancji problemu. W przypadku Tabu Search dla małej instancji problemu, nieregularność wynika stąd, iż dobre rozwiązanie może zostać znalezione już po minucie wykonywania algorytmu, stąd najwyższy skok jakościowy mógłby zostać zaobserwowany w bardzo krótkim czasie po rozpoczęciu wykonywania algorytmu. W przypadku dużej instancji problemu miejsce miała podobna sytuacja, jednakże tym razem przeznaczony został za krótki przedział czasu aby możliwe było zaobserwowanie dużego skoku jakości rozwiązania. Opisana powyżej zależność jest lepiej widoczna w przypadku Symulowanego Wyżarzania. Algorytm ten wykazał dużą większą szybkość działania w porównaniu do przeszukiwania lokalnego, umożliwiając znalezienie dobrych rozwiązań wielokrotnie szybciej, stąd badanie błędu względnego w jego przypadku zostało przeprowadzone dla krótszych przedziałów czasowych. Dla małej instancji, przedziały te również okazały się za długie żeby zaobserwować wyraźny wzrost jakości rozwiązania. Stało się to możliwe dopiero dla większych instancji. Zaobserwowana została również dla nich dużo lepsza jakość rozwiązań.

W przypadku Tabu Search, najlepsze wyniki zostały uzyskane jedynie dla małej instacji natomiast dla większych ilości danych nastąpiła duża utrata dokładności.

6. Bibliografia

1. <https://nowosad.github.io/ahod/11-simulated-annealing.html#1>
2. https://pl.wikipedia.org/wiki/Symulowane_wy%C5%BCarzenie
3. http://155.158.112.25/~algorytmyewolucyjne/materialy/algorytm_symulowanego_wy_zarzania.pdf
4. http://www.zio.iiar.pwr.wroc.pl/pea/w5_ts.pdf
5. https://pl.wikipedia.org/wiki/Przeszukiwanie_tabu