

Podstawy efektywnych algorytmów

Prowadzący zajęcia: dr inż. Jarosław Mierzwa

Termin zajęć: Piątek, godz. 11:15

Autor projektu:

Mateusz Gawłowski 264463

Rodzaj grupy:

Późniejsza

Etap projektu:

2

Data oddania sprawozdania:

15.12.2023

Spis treści

1. Wstęp teoretyczny	2
2. Plan eksperymentu	3
3. Analiza danych pomiarowych.	4
3.1. Simulated Annealing	4
3.2. Tabu Search.....	10
4. Wnioski	16
5. Bibliografia.....	16

1. Wstęp teoretyczny

Celem drugiego zadania projektowego było zaimplementowanie oraz analiza skuteczności algorytmów wykorzystujących techniki przeszukiwania lokalnego. W ramach projektu skupiono się na dwóch algorytmach: przeszukiwaniu z zakazami (Tabu Search) oraz symulowanym wyżarzaniu (Simulated Annealing). Oba te algorytmy są heurystycznymi metodami, co oznacza, że pozwalają one na znalezienie satysfakcjonujących rozwiązań problemów przy akceptowalnym zużyciu zasobów obliczeniowych, jednakże nie dają gwarancji co do optymalności uzyskanego rozwiązania ani nie definiują jego jakości poprzez odległość od optimum globalnego.

Oba algorytmy są rozwinięciem koncepcji przeszukiwania lokalnego. W trakcie działania każdy z tych algorytmów eksploruje określone obszary w całej przestrzeni rozwiązań, które zwane są sąsiedztwem danego rozwiązania. Na początku ustawiane jest rozwiązanie startowe będące permutacją MST, powstałą w wyniku działania Algorytmu Prima. Otoczenie to będzie eksplorowane w kolejnych etapach algorytmu. Rozwiązanie sąsiednie różni się od pierwotnego rozwiązania w określony sposób. Zbiór wszystkich możliwych rozwiązań sąsiednich dla danego rozwiązania stanowi sąsiedztwo tego rozwiązania. Badane algorytmy różnią się sposobem przeglądania tego sąsiedztwa, aż do osiągnięcia warunku stopu.

Głównym założeniem przeszukiwania z zakazami jest wybór dla danego rozwiązania startowego takiego sąsiada, dla którego wartość funkcji oceny ruchu (mierząca opłacalność przyjęcia sąsiedniego rozwiązania jako aktualnego) jest najwyższa. W przypadku problemu komiwożacza, funkcją tą jest różnica długości trasy pomiędzy aktualnym a sąsiednim rozwiązaniem. Ta strategia wymaga przeglądania całego sąsiedztwa startowego rozwiązania w celu znalezienia najlepszego sąsiada, co prowadzi do stromego podejścia. Istnieje ryzyko, że algorytm może utknąć w lokalnym optimum, ograniczając eksplorację przestrzeni rozwiązań (dywersyfikację). Aby złagodzić ten efekt, stosuje się Listę Tabu, gdzie określone ruchy są zabronione przez pewną liczbę iteracji przeszukiwania. W pewnych sytuacjach ruchy z listy Tabu mogą być wykorzystane, jeśli spełniają kryterium Aspiracji, nawet jeśli są zakazane.

Drugim zaimplementowanym algorytmem był symulowane wyżarzanie (Simulated Annealing). Podobnie jak w przypadku Tabu Search, algorytm ten bada różne obszary przestrzeni rozwiązań, jednakże stosuje inną strategię wyboru rozwiązania sąsiedniego. W przeciwieństwie do Tabu Search, nie wybiera zawsze najlepszego sąsiada, lecz każdy ruch prowadzący do poprawy obecnego rozwiązania. Ta strategia jest określana jako zachłanna. Dodatkowo, algorytm ten może przyjąć gorsze rozwiązania, co pozwala na eksplorację lepszych rozwiązań, potencjalnie globalnie optymalnych. Symulowane wyżarzanie odnosi się do procesu przypominającego wyżarzanie metalu, gdzie stopniowo ogranicza się akceptację gorszych rozwiązań wraz ze zmniejszaniem temperatury.

Temperatura, będąca parametrem algorytmu, decyduje o prawdopodobieństwie przyjęcia gorszego rozwiązania - im wyższa temperatura, tym większa dywersyfikacja, a im niższa - większa intensyfikacja. Algorytm na początku pracy często zmienia rozwiązania, z czasem bardziej skupiając się na poprawie jakości. Pod koniec działania, przy niskiej temperaturze, wybierane są głównie lepsze rozwiązania, podobnie jak w standardowych metodach iteracyjnych.

2. Plan eksperymentu

Badanie efektywności zaimplementowanych algorytmów polegało na analizie błędów względnych w zależności od czasu wykonania dla trzech różnych instancji problemu komiwojażera. W każdej z tych instancji kryterium stopu było określone jako maksymalny czas trwania algorytmu, zależny od rozmiaru instancji. Dla plików `ftv55.atsp` wynosił on 60 sekund, dla `ftv170.atsp` – 120 sekund, a dla `rbg358.atsp` – 240 sekund. W przypadku algorytmu Tabu Search zdefiniowano trzy rodzaje sąsiedztwa: Insert, Invert oraz Swap, natomiast dla Symulowanego Wyżarzania rozważano jedynie sąsiedztwo typu Swap. Dodatkowo, dla Symulowanego Wyżarzania należało dobrać współczynnik obniżania temperatury oraz ustalić funkcję określającą temperaturę początkową w zależności od danych wejściowych.

Obliczanie początkowej temperatury wykonywane było poprzez 10 000-krotne próbkowanie przestrzeni rozwiązań. Próbkowanie polegało na generowaniu losowego rozwiązania i jego sąsiada, a następnie obliczeniu różnicy w jakości tych rozwiązań. Na podstawie zebranych próbek wyznaczana była średnia wartość różnic, która następnie była używana w formule do obliczenia temperatury początkowej:

$$T = \frac{-\delta}{\log(0,99)}$$

gdzie $-\delta$ 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.

3. Analiza danych pomiarowych.

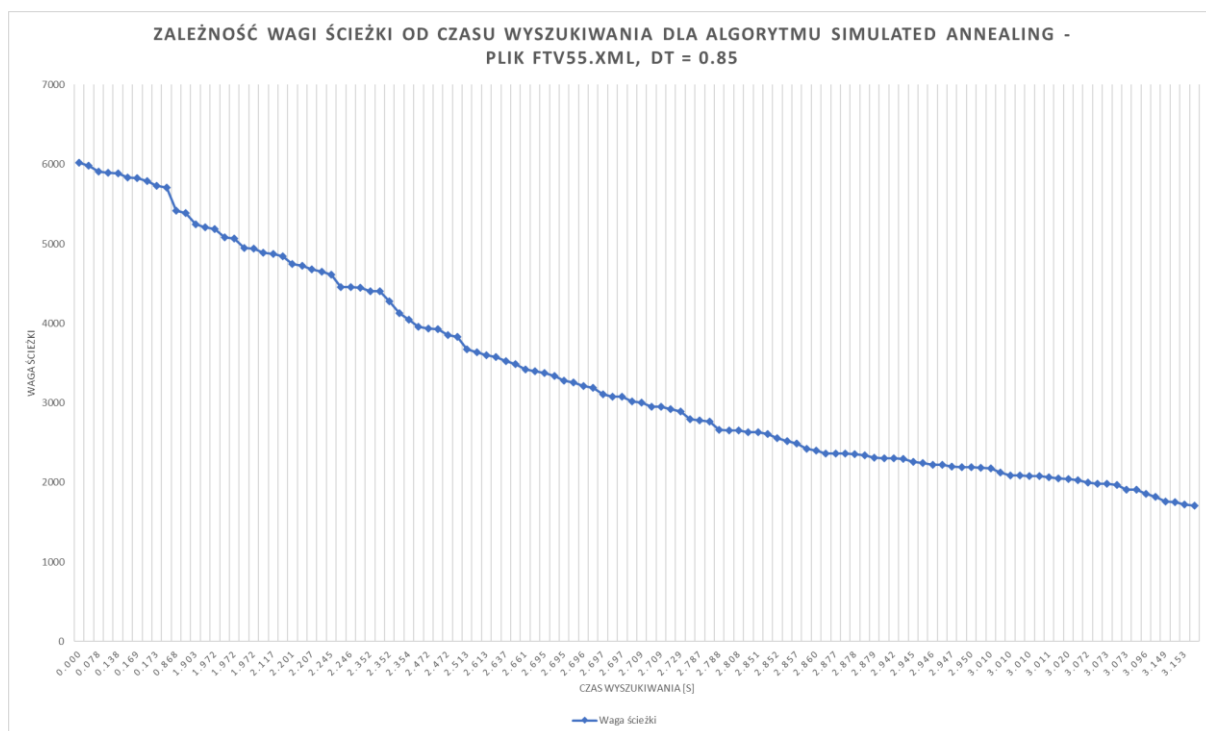
Poniżej przedstawiono zestawienie 10 najlepszych wyszukanych ścieżek, błąd względny oraz czas wyszukiwania dla trzech różnych plików.

3.1. Simulated Annealing

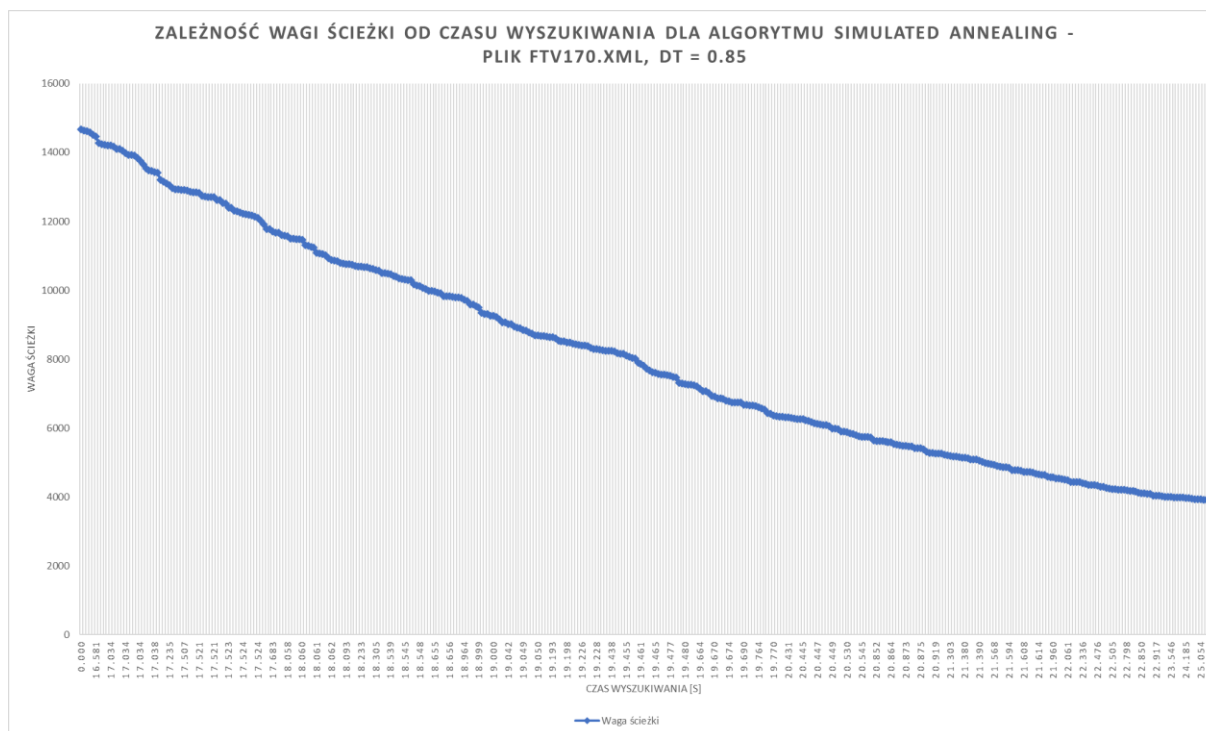
Tabela 1. Zestawienie błędów względnych oraz czasów szukania ścieżki - SA

Plik	ftv55.xml			ftv170.xml			rgb358.xml		
Nr	Najlepsza ścieżka	Błąd [%]	Czas [s]	Najlepsza ścieżka	Błąd [%]	Czas [s]	Najlepsza ścieżka	Błąd [%]	Czas [s]
1	1633	0.0155	49.541	3830	0.3902	38.679	1242	0.0679	159.895
2	1654	0.0286	52.536	3824	0.3880	45.521	1252	0.0765	167.152
3	1639	0.0193	55.768	3942	0.4309	44.547	1264	0.0868	158.647
4	1648	0.0249	49.223	3921	0.4232	37.543	1277	0.0980	162.753
5	1633	0.0155	53.631	3833	0.3913	49.534	1238	0.0645	163.163
6	1637	0.0180	45.767	3812	0.3837	47.231	1249	0.0739	152.675
7	1644	0.0224	46.113	3936	0.4287	41.574	1283	0.1032	156.654
8	1652	0.0274	41.563	3911	0.4196	39.778	1265	0.0877	171.362
9	1655	0.0292	49.986	3887	0.4109	42.765	1294	0.1126	166.531
10	1641	0.0205	51.523	3895	0.4138	38.421	1246	0.0714	160.211

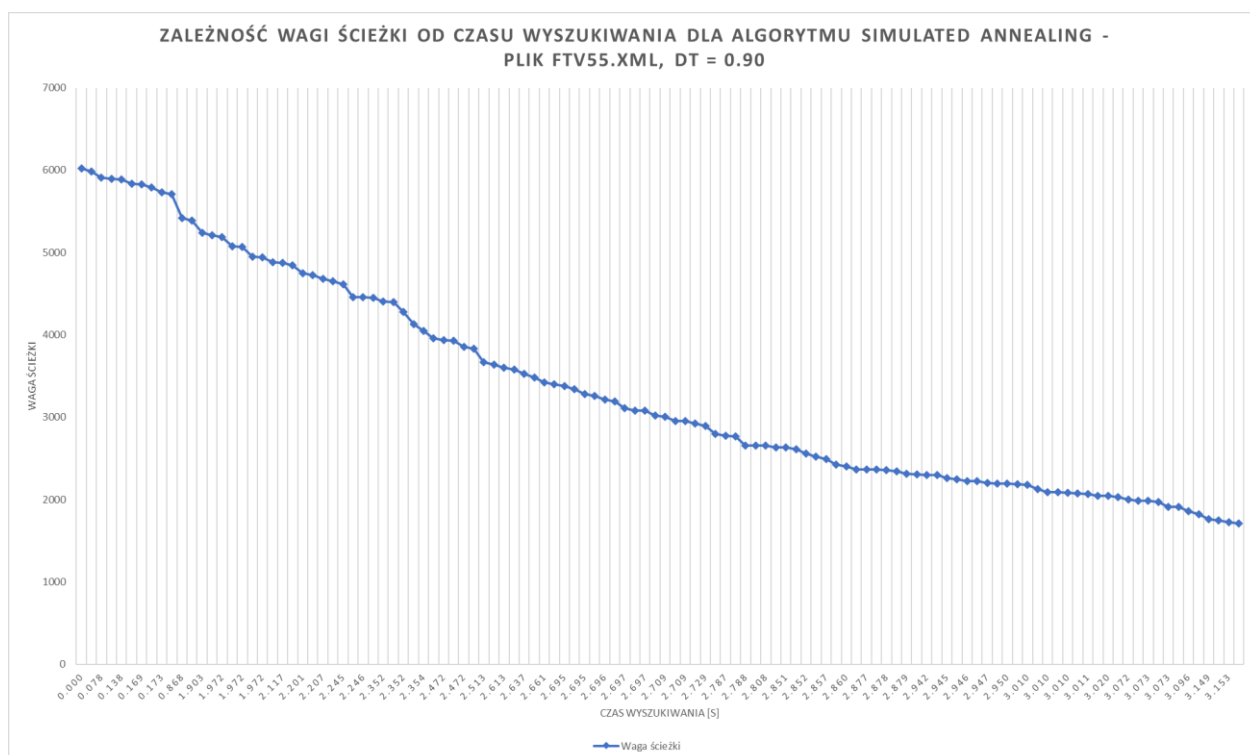
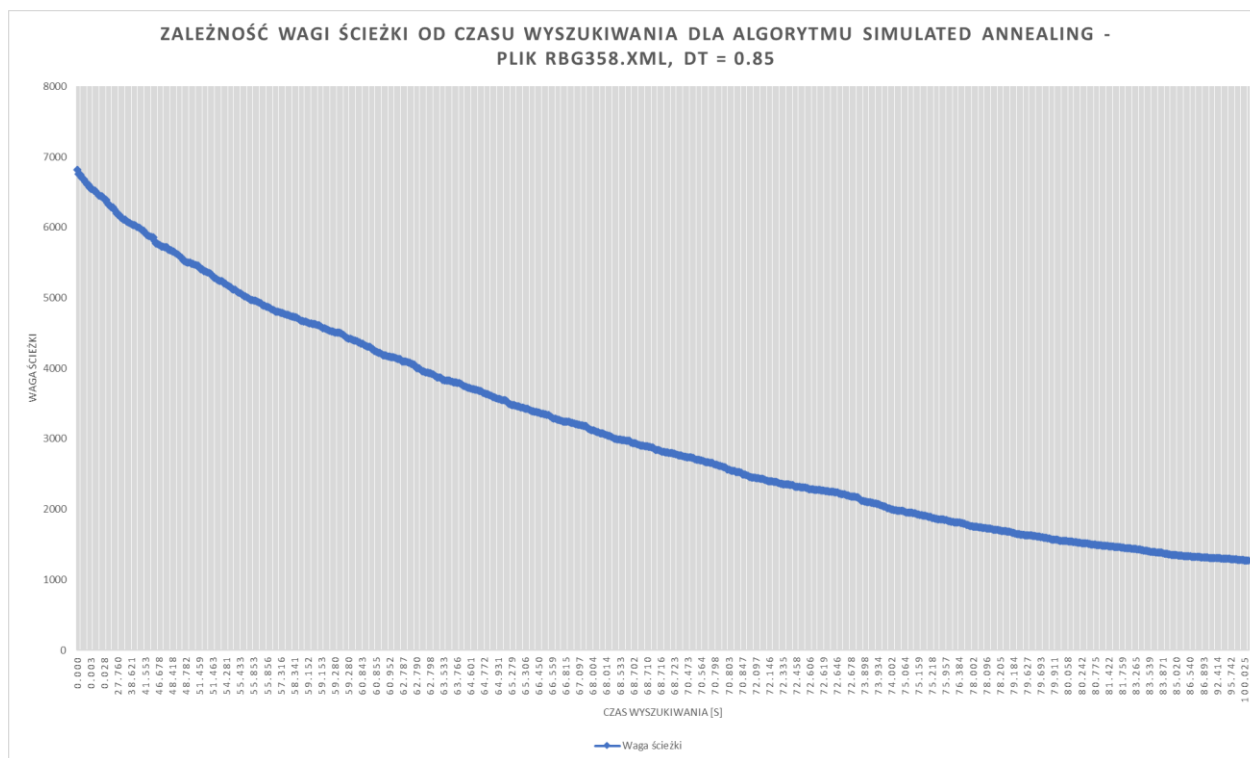
Poniżej zestawione zostały wyniki analizy wagi wyszukanej ścieżki w zależności od czasu wyszukiwania algorytmu oraz współczynnika zmiany temperatury (dT).

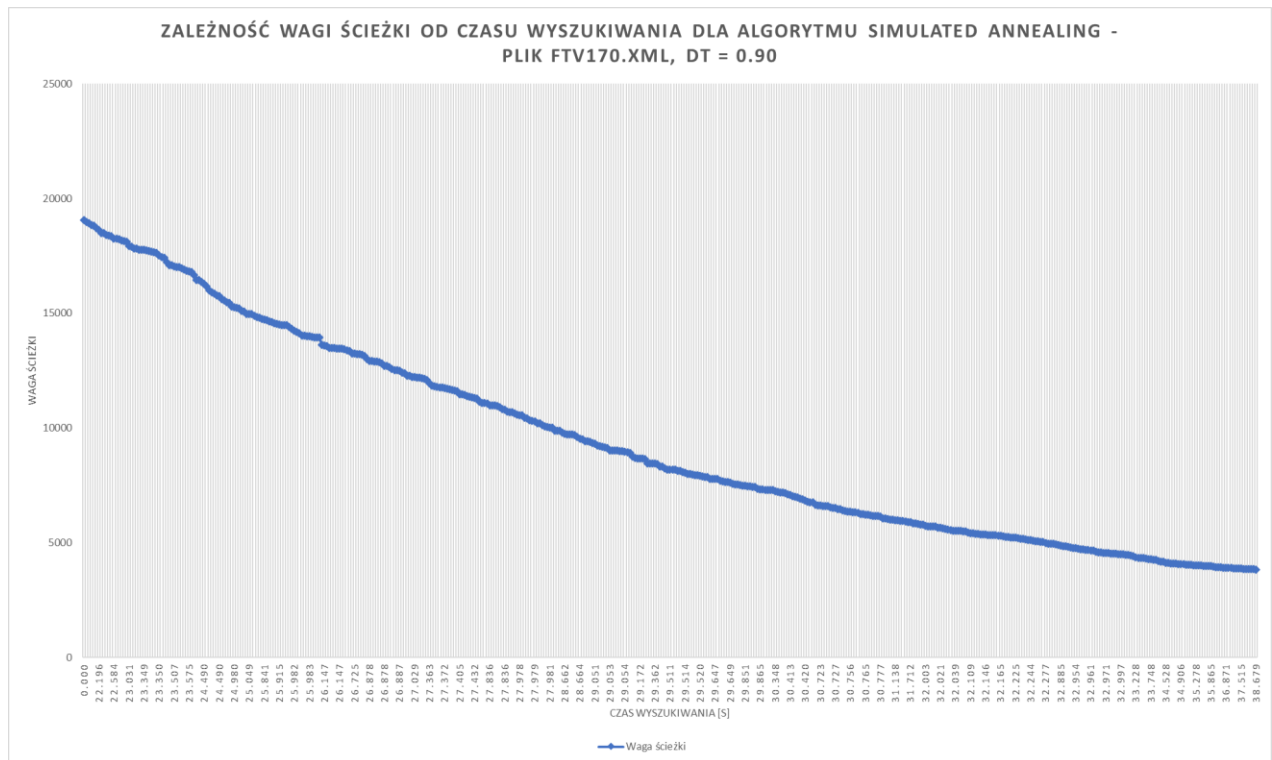


Wykres 1. Zależność wagi ścieżki od czasu wyszukiwania – SA – ftv55 – dt 0.85

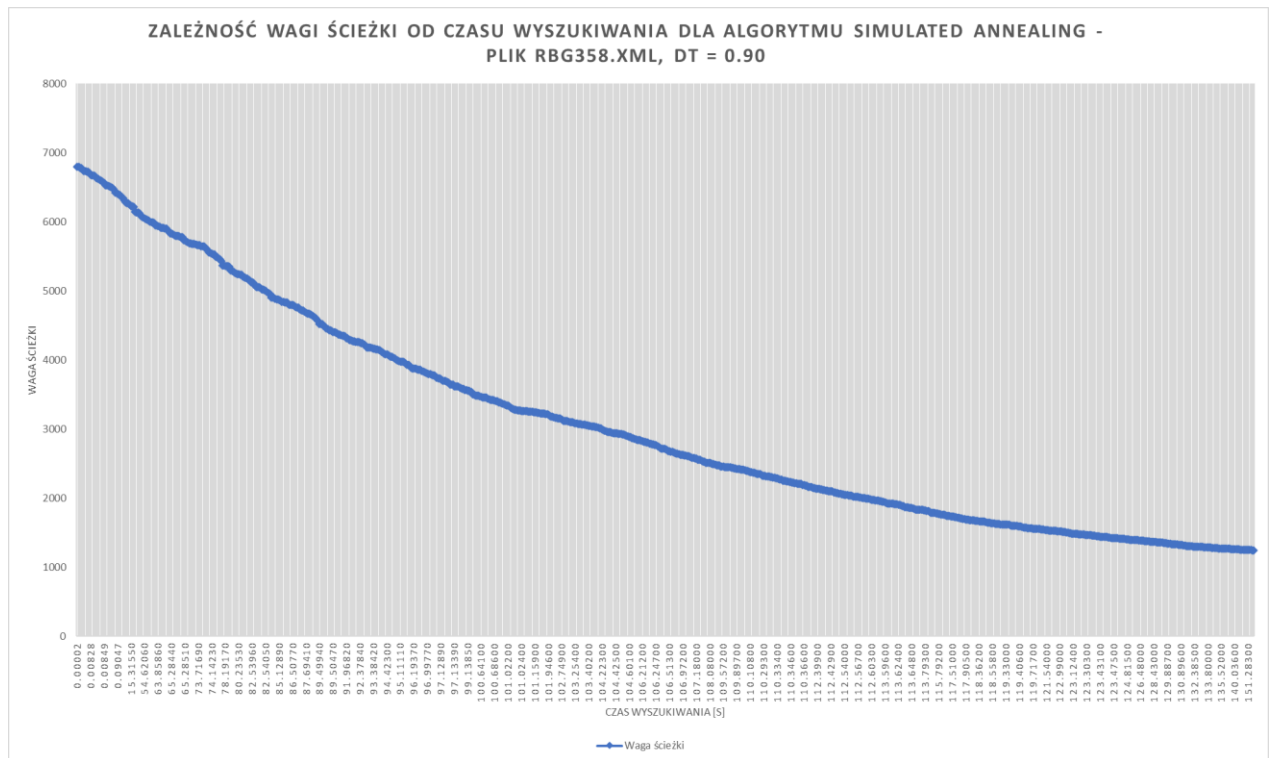


Wykres 2. Zależność wagi ścieżki od czasu wyszukiwania – SA – ftv170 – dt 0.85

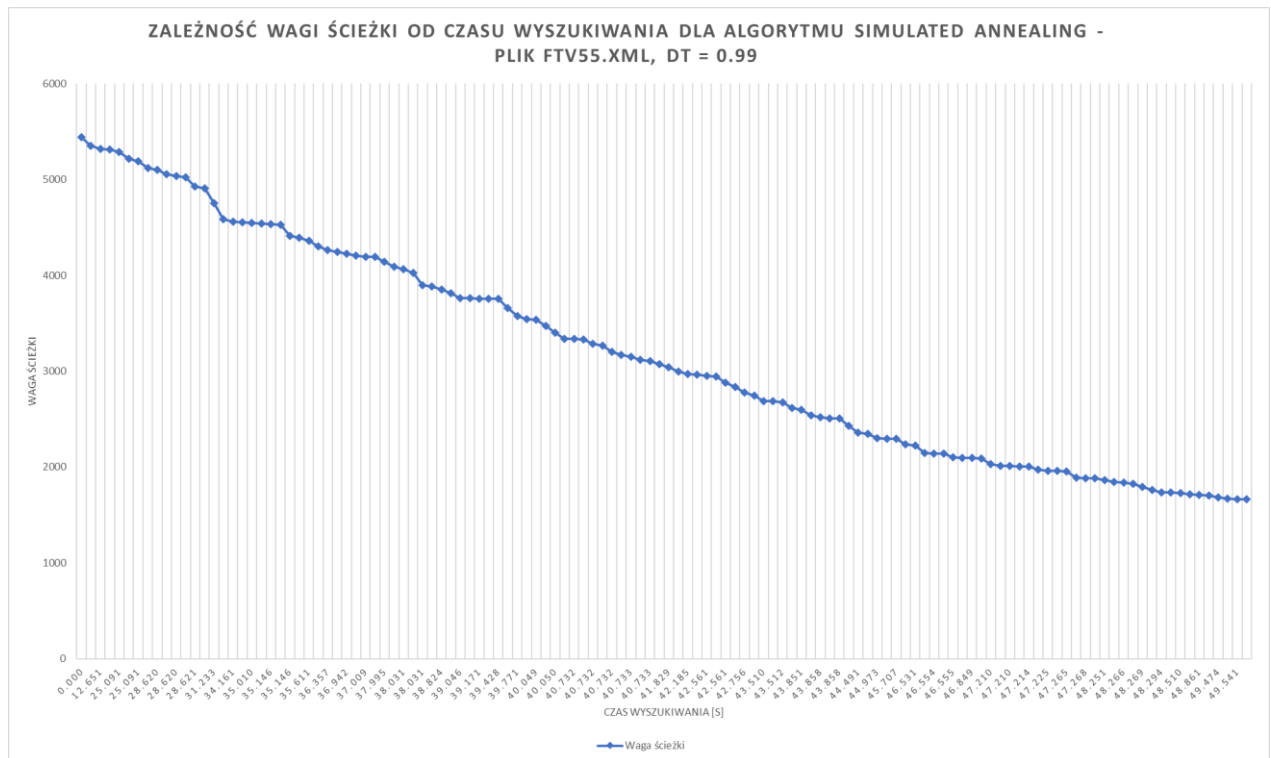




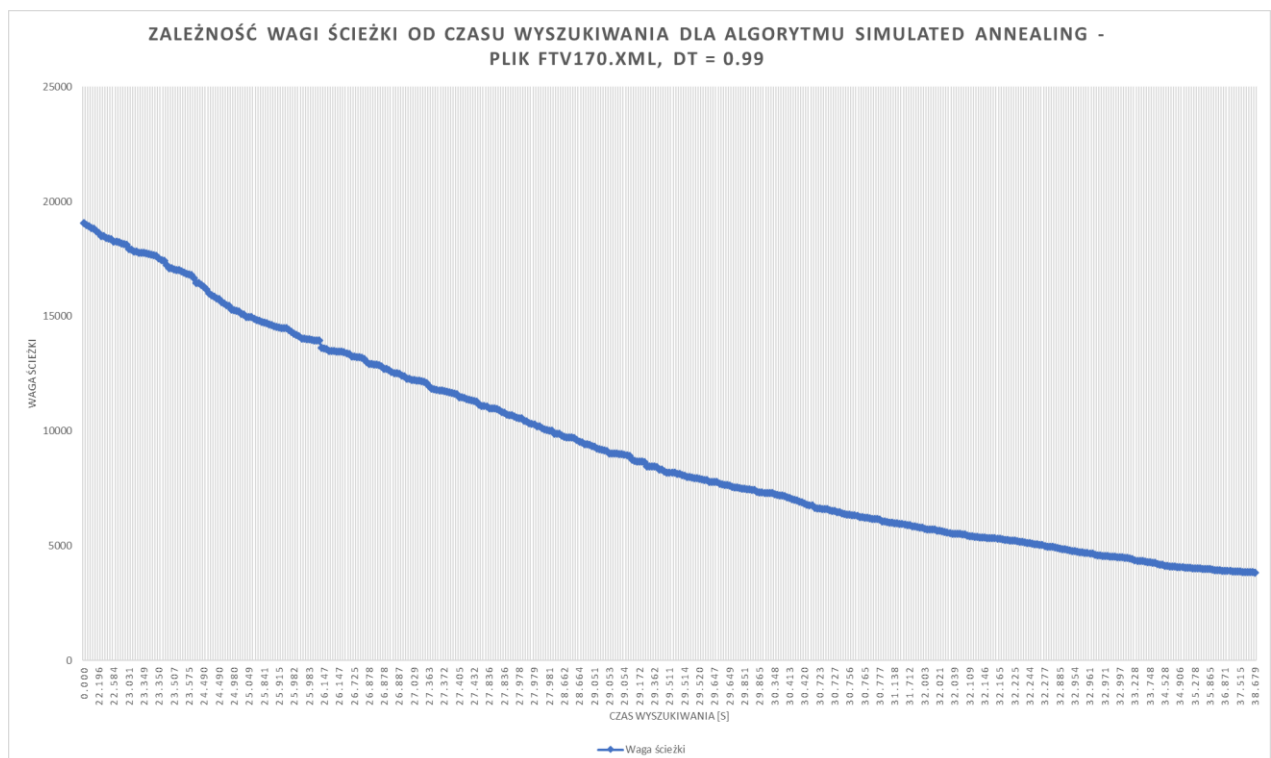
Wykres 5. Zależność wagi ścieżki od czasu wyszukiwania – SA – ftv170 – dt 0.90



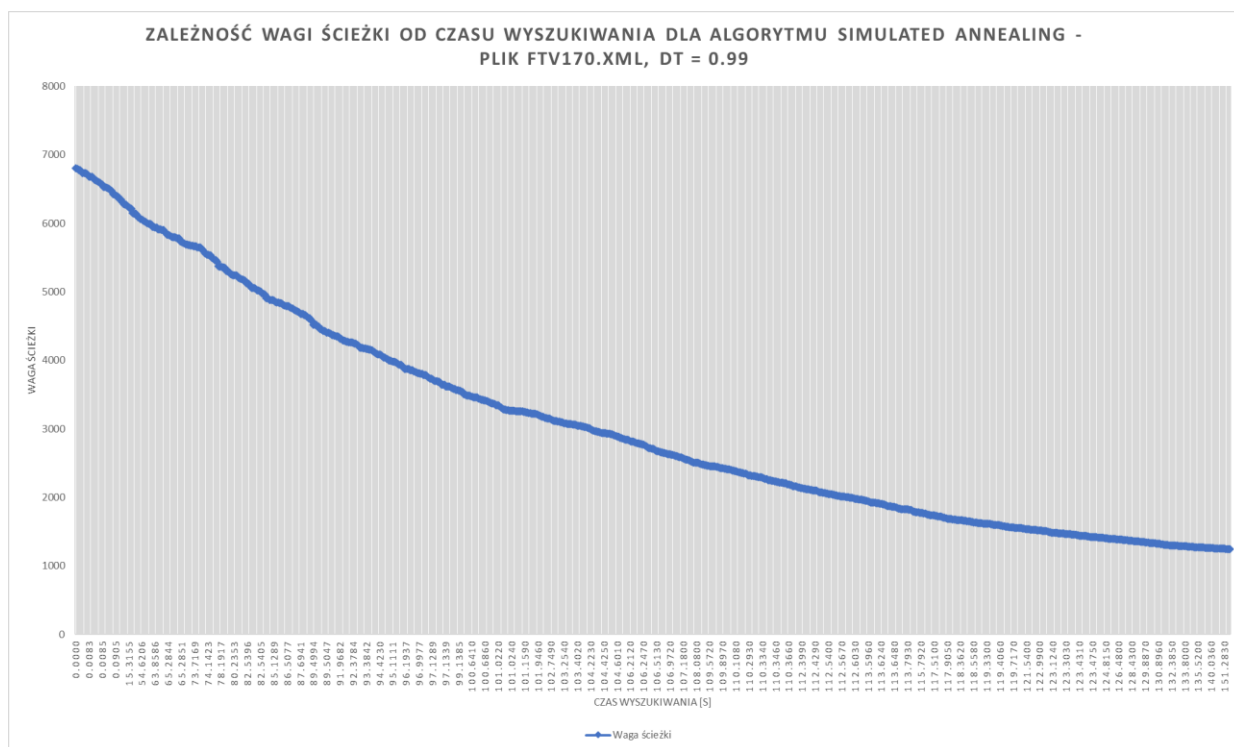
Wykres 6. Zależność wagi ścieżki od czasu wyszukiwania – SA – rbg358 – dt 0.90



Wykres 7. Zależność wagi ścieżki od czasu wyszukiwania – SA – ftv55 – dt 0.99



Wykres 8. Zależność wagi ścieżki od czasu wyszukiwania – SA – ftv170 – dt 0.99

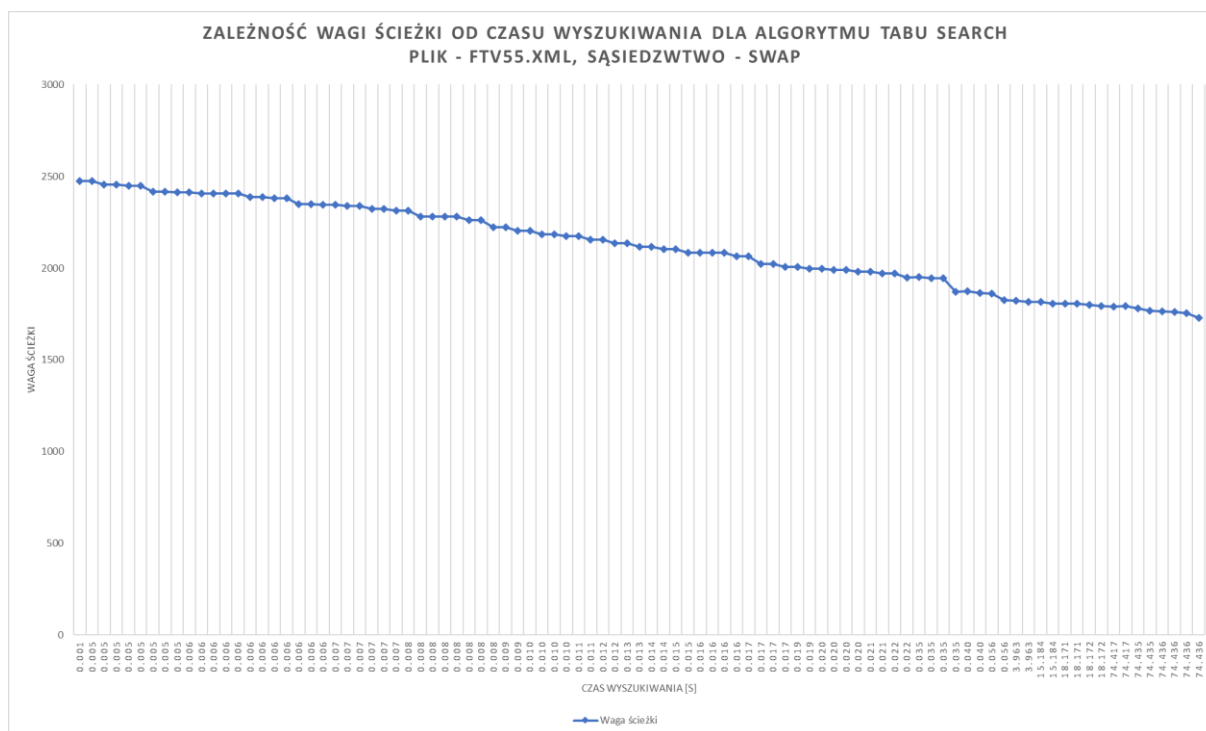


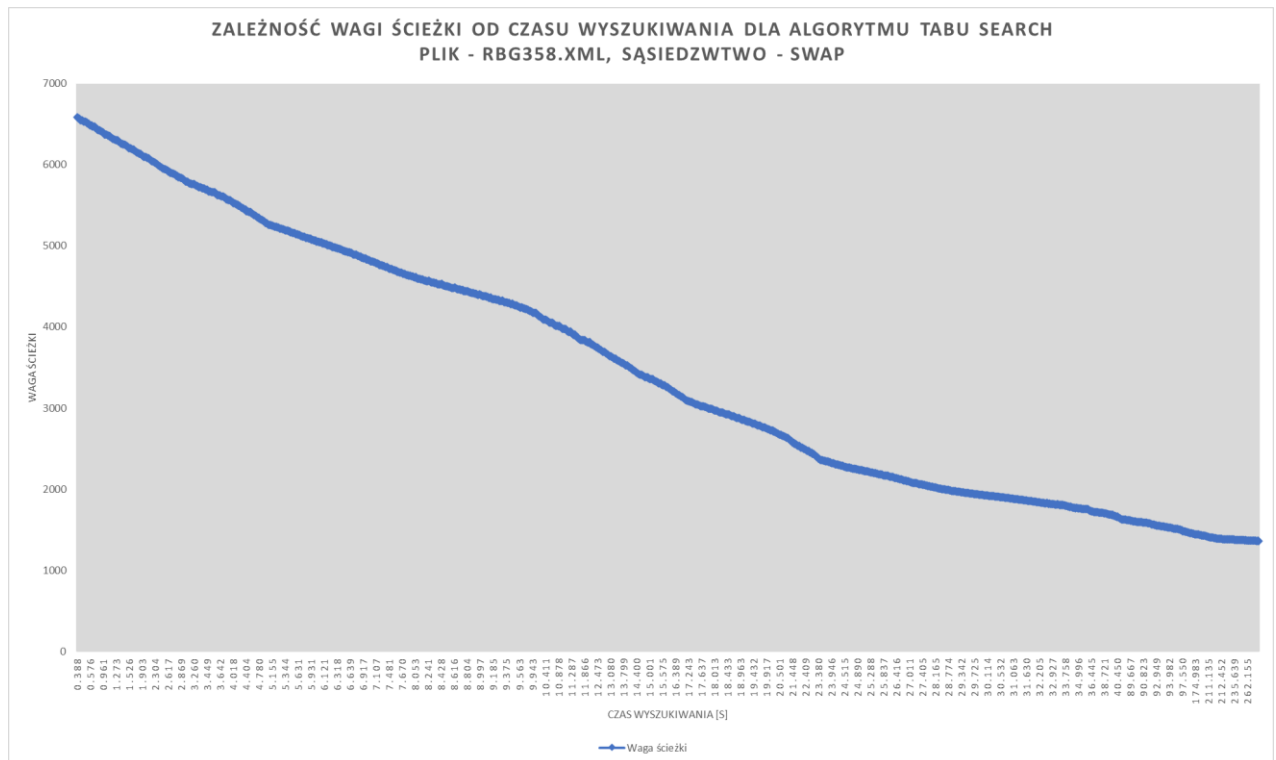
3.2. Tabu Search

Tabela 2. Zestawienie błędów względnych oraz czasów szukania ścieżki – TS

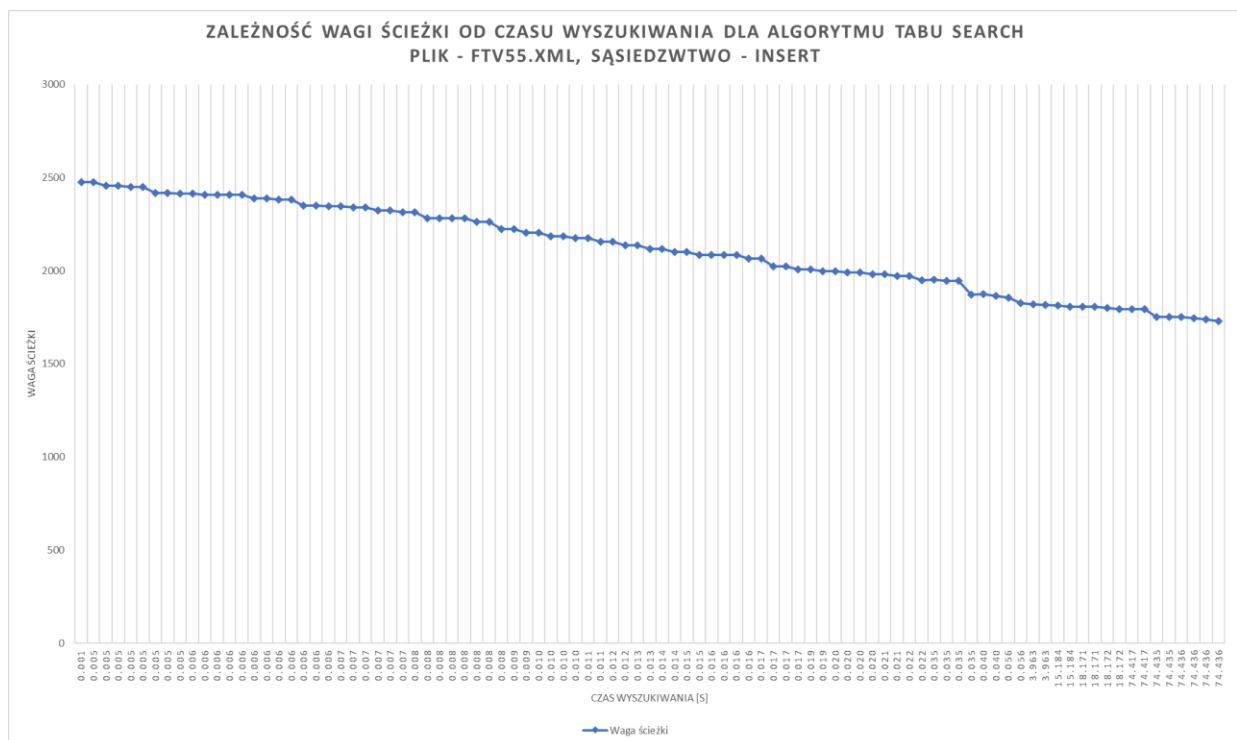
Plik	ftv55.xml			ftv170.xml			rgb358.xml		
Nr	Najlepsza ścieżka	Błąd [%]	Czas [s]	Najlepsza ścieżka	Błąd [%]	Czas [s]	Najlepsza ścieżka	Błąd [%]	Czas [s]
1	1730	0.0155	74.436	3586	0.3902	114.584	1362	0.0679	350.447
2	1737	0.0286	78.936	3591	0.388	134.853	1365	0.0765	366.352
3	1714	0.0193	83.792	3697	0.4309	131.968	1375	0.0868	347.712
4	1724	0.0249	73.958	3677	0.4232	111.219	1389	0.098	356.711
5	1708	0.0155	80.581	3594	0.3913	146.741	1347	0.0645	357.610
6	1712	0.018	68.766	3575	0.3837	139.919	1359	0.0739	334.623
7	1720	0.0224	69.285	3691	0.4287	123.160	1396	0.1032	343.344
8	1728	0.0274	62.449	3668	0.4196	117.840	1376	0.0877	375.580
9	1731	0.0292	75.105	3645	0.4109	126.689	1408	0.1126	364.991
10	1716	0.0205	77.414	3653	0.4138	113.820	1355	0.0714	351.140

Poniżej zestawione zostały wyniki analizy wagi wyszukanej ścieżki w zależności od czasu wyszukiwania algorytmu oraz typu sąsiedztwa.

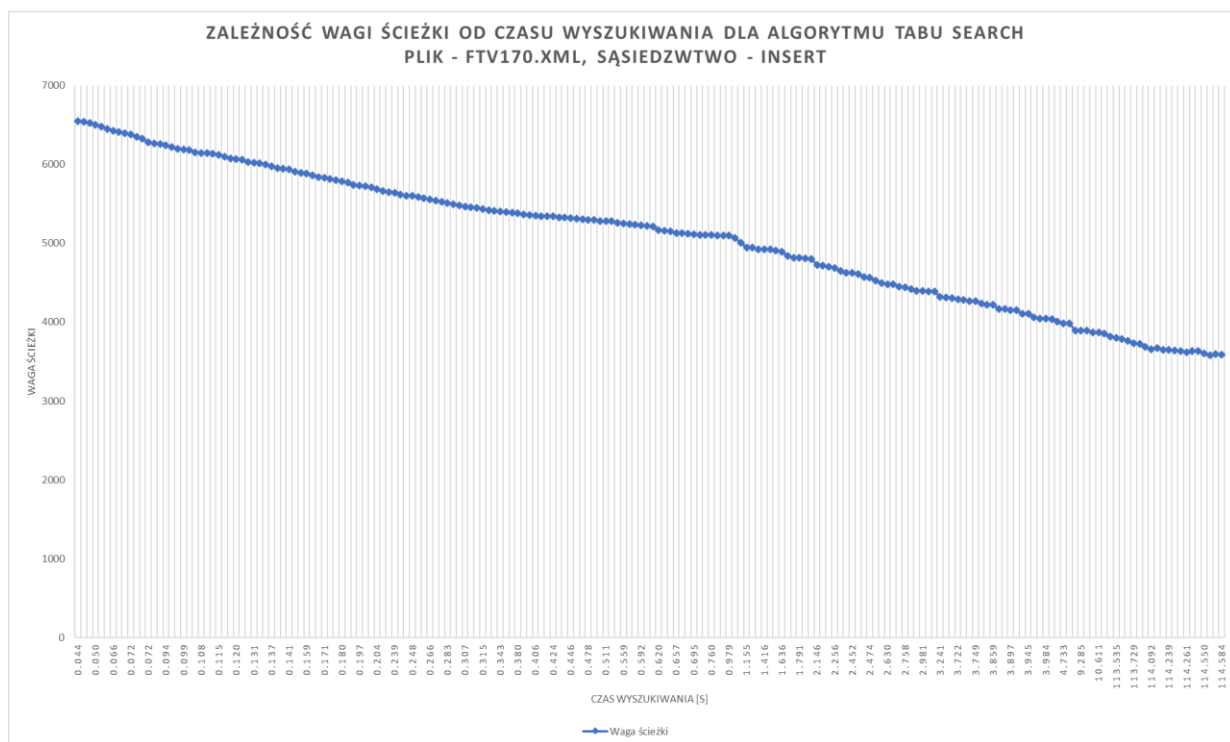




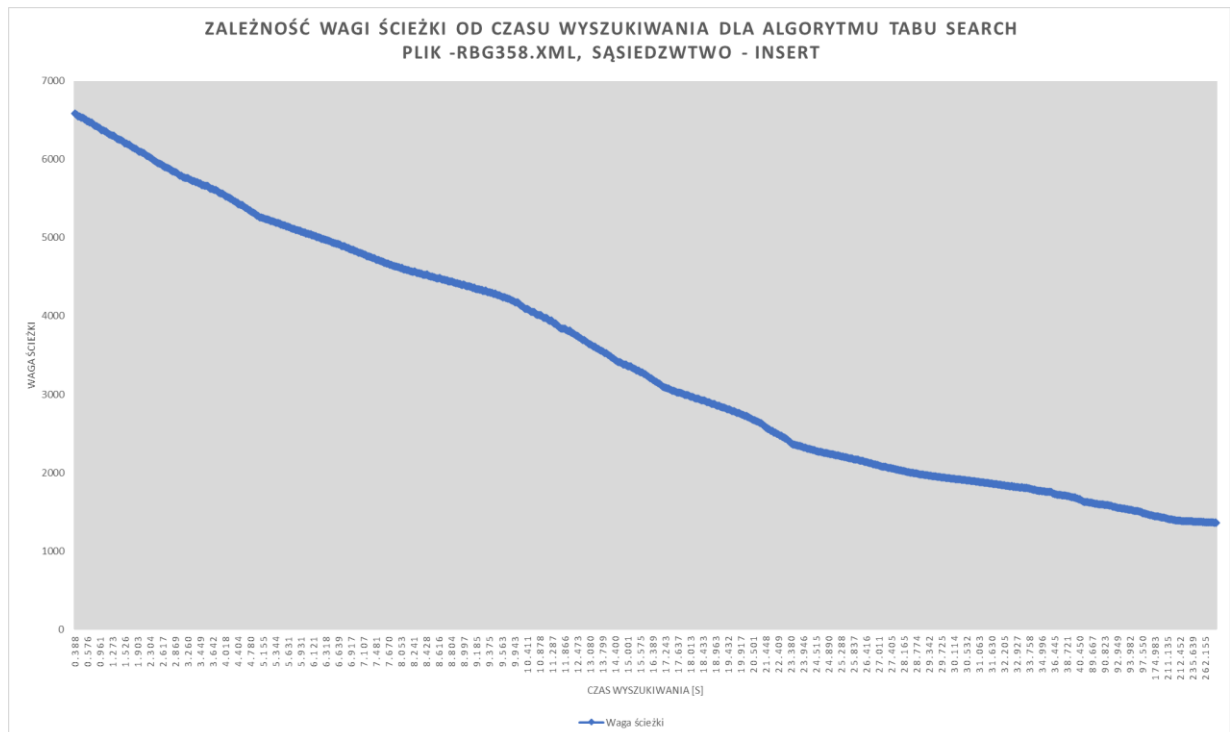
Wykres 12. Zależność wagi ścieżki od czasu wyszukiwania –TS –rbg358 – swap



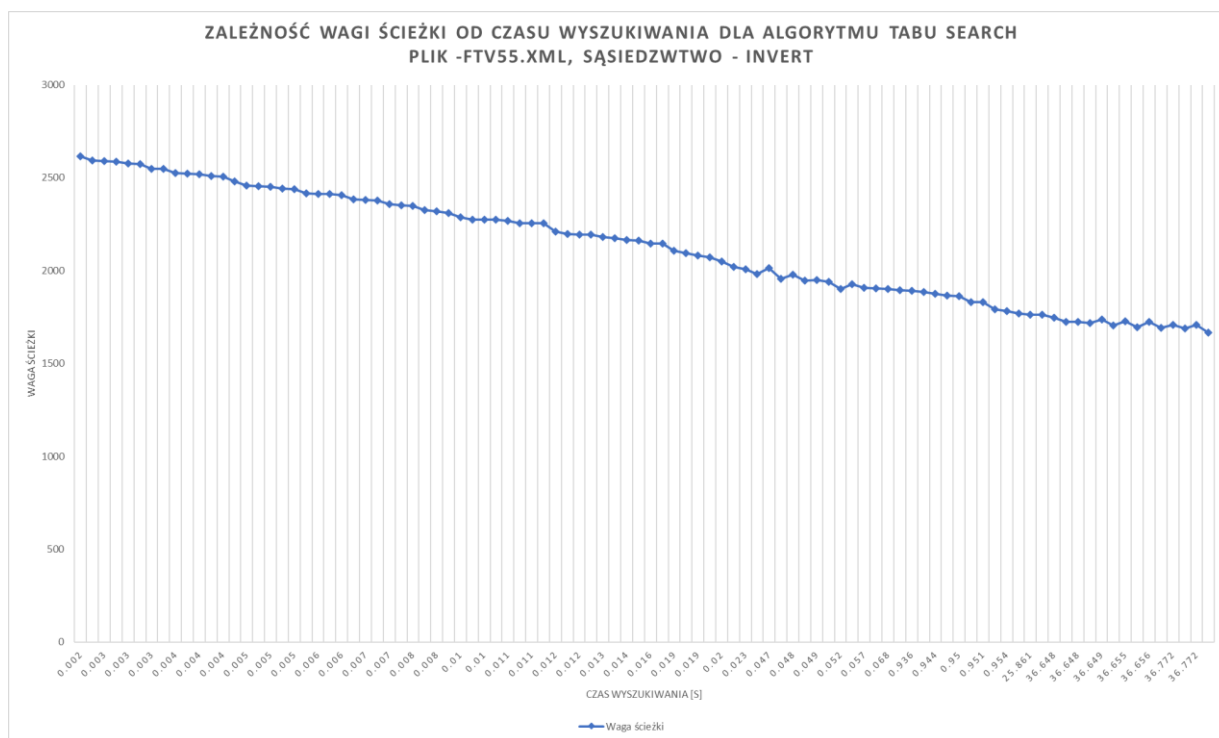
Wykres 13. Zależność wagi ścieżki od czasu wyszukiwania –TS –ftv55 – insert



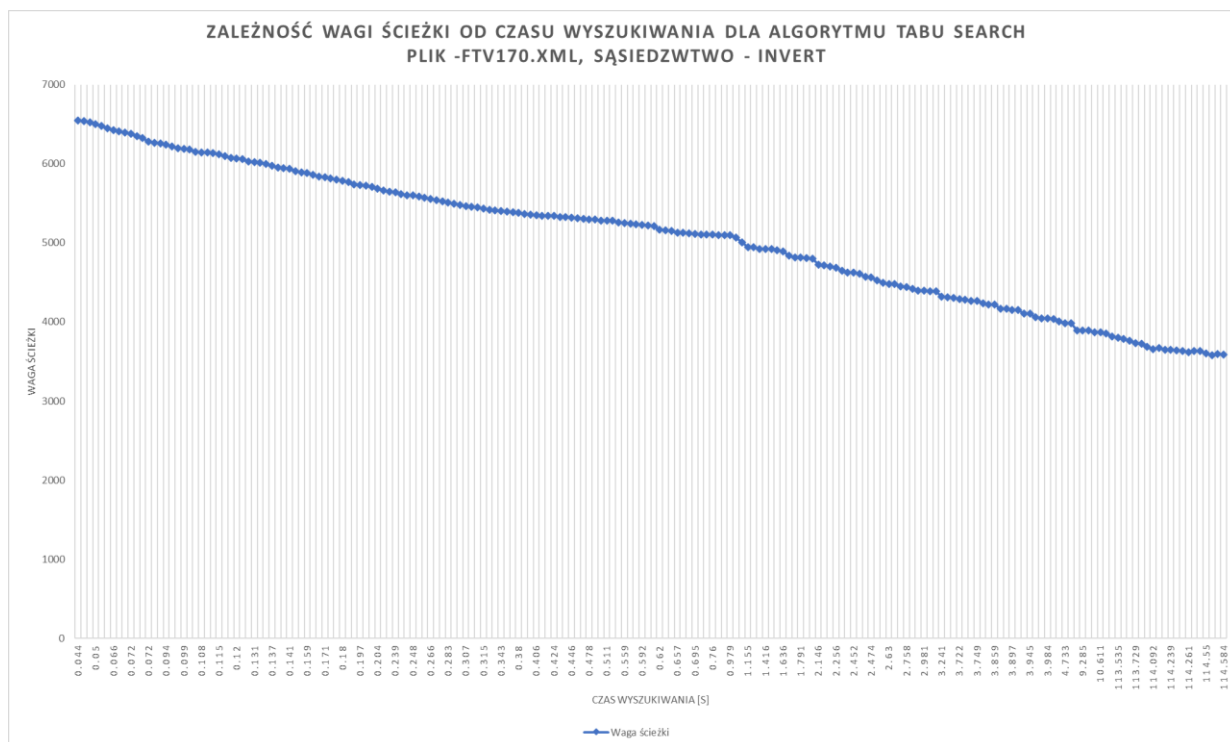
Wykres 14. Zależność wagi ścieżki od czasu wyszukiwania –TS –ftv170 – insert



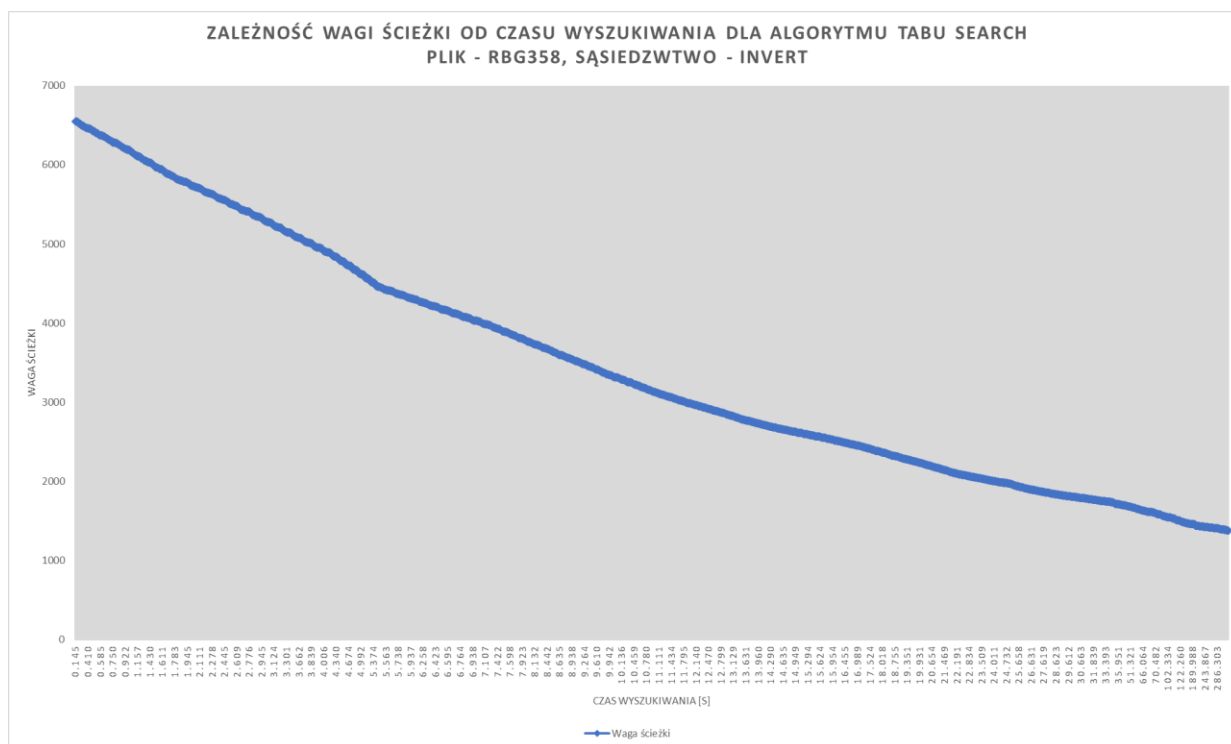
Wykres 15. Zależność wagi ścieżki od czasu wyszukiwania –TS –rbg358 – insert



Wykres 16. Zależność wagi ścieżki od czasu wyszukiwania –TS – ftv55 – invert



Wykres 17. Zależność wagi ścieżki od czasu wyszukiwania –TS – ftv170 – invert



Wykres 18. Zależność wagi ścieżki od czasu wyszukiwania –TS –rbg358 –invert

4. Wnioski

Podsumowanie analizy wyników uzyskanych w trakcie przeprowadzonych badań pozwala zauważyć istotny wpływ czasu działania na jakość rozwiązań dla obu zbadanych algorytmów. W przypadku Tabu Search dla małych instancji problemu możliwe było szybkie znalezienie satysfakcjonującego rozwiązania już po krótkim czasie, co prowadziło do wyraźnego skoku jakościowego w początkowej fazie działania algorytmu. W sytuacji większych instancji problemu, podobna dynamika była obserwowana, jednakże zbyt krótki czas działania uniemożliwiał zaobserwowanie znaczącego skoku jakości rozwiązania. Ta zależność jest bardziej wyraźna w przypadku Symulowanego Wyżarzania. Ten algorytm wykazywał szybszą efektywność w porównaniu z przeszukiwaniem lokalnym, umożliwiając szybsze znalezienie satysfakcjonujących rozwiązań. Dla małych instancji, jednakowo jak w przypadku Tabu Search, zbyt długie okresy działania utrudniały zauważenie wyraźnego wzrostu jakości rozwiązania. Dopiero dla większych instancji możliwe było zaobserwowanie istotnej poprawy jakości zaproponowanych rozwiązań. Ta obserwacja była również związana z wyraźnie lepszą jakością rozwiązań dla tych większych instancji problemu.

5. Bibliografia

- <http://www2.imm.dtu.dk/courses/02719/tabu/4tabu2.pdf>
- http://www.pi.zarz.agh.edu.pl/intObl/notes/IntObl_w2.pdf
- http://155.158.112.25/~algorytmyewolucyjne/materialy/algorytm_symulowanego_wyzarzania.pdf