

Projektowanie efektywnych algorytmów

*Implementacja algorytmów
metaheurystycznych dla problemu
komiwojażera*

Prowadzący:

Dr inż. Zbigniew Buchalski

Spis treści

1.	Wstęp teoretyczny	3
1.1.	Algorytm Tabu Search	3
1.1.1.	Struktury sąsiedztwa	3
1.1.2.	Lista tabu	3
1.1.3.	Dywersyfikacja.....	3
1.2.	Symulowane wyżarzanie	3
1.2.1.	Ważne parametry algorytmu	4
2.	Pomiary.....	4
2.1.	Wyniki pomiarów	5
2.1.1.	Problem mały	5
2.1.2.	Problem średni	6
2.1.3.	Problem duży.....	7
2.2.	Analiza pomiarów	8
2.2.1.	Problem mały	8
2.2.2.	Problem średni	10
2.2.3.	Problem duży.....	11
3.	Podsumowanie i wnioski	12
4.	Źródła.....	13

1. Wstęp teoretyczny

1.1. Algorytm Tabu Search

Przeszukiwanie tabu (ang. Tabu search - TS) – metaheurystyka stosowana do rozwiązywania problemów optymalizacyjnych. Wykorzystywana do otrzymywania rozwiązań optymalnych lub niewiele różniących się od niego dla problemów z różnych dziedzin (np. planowanie, planowanie zadań). Twórcą algorytmu jest Fred Glover.

Podstawową ideą algorytmu jest przeszukiwanie przestrzeni, stworzonej ze wszystkich możliwych rozwiązań, za pomocą sekwencji ruchów. W sekwencji ruchów istnieją ruchy niedozwolone, ruchy tabu. Algorytm unika oscylacji wokół optimum lokalnego dzięki przechowywaniu informacji o sprawdzonych już rozwiązaniach w postaci listy tabu (TL).

1.1.1. Struktury sąsiedztwa

Ze względu na przestrzeń rozwiązań należy zdefiniować relację sąsiedztwa na parach elementów tej przestrzeni, która obejmuje całą dziedzinę przestrzeni przeszukiwań. Definicja relacji zależy oczywiście od przestrzeni i formatu rozwiązań (np. rozwiązaniami mogą być wektory binarne, wektory liczb rzeczywistych, permutacje zbioru liczb naturalnych itp...). W wykonanym algorytmie uwzględnione zostały trzy różne definicje sąsiedztwa:

- $\text{swap}(i, j)$ – zamiana miejscami i -tego elementu z j -tym
- $\text{insert}(i, j)$ – przeniesienia j -tego elementu na pozycję i -tą
- $\text{invert}(i, j)$ – odwrócenie kolejności w podciągu zaczynającym się na i -tej pozycji i kończącym na pozycji j -tej

1.1.2. Lista tabu

Lista tabu jest to lista ruchów, które zamieniamy na inne w przypadku ich wygenerowania. Ta lista jest rozszerzana o wierzchołki, które zostały wylosowane w poprzedniej iteracji algorytmu. Każdy element listy jest wyłączony na określoną ilość czasu - w programie jest to czas równy ilości miast np. w przypadku gdy mamy 47 miast, dany ruch jest zakazany na 47 iteracji.

1.1.3. Dywersyfikacja

Jest to procedura pozwalająca na przeszukiwanie różnych obszarów przestrzeni stanów. W programie wykorzystana została strategia dywersyfikacji nazywana metodą zdarzeń krytycznych. Funkcja `diversification()` sprawdza przez ile kolejnych iteracji nie zostało znalezione lepsze rozwiązanie, a następnie jeśli ta ilość przekroczyła ilość krytyczną (w tym przypadku jest to dwukrotność ilości miast) to generuje nowe rozwiązanie początkowe. Algorytm ponownie rozpoczyna działanie od wygenerowanego rozwiązania

1.2. Symulowane wyżarzanie

Symulowane wyżarzanie (ang. Simulated annealing – SA) jest probabilistyczną techniką przybliżania globalnego optimum danej funkcji. W szczególności jest to metaheurystyka do przybliżenia optymalizacji globalnej w dużej przestrzeni wyszukiwania dla problemu optymalizacji. Jest ono często

używane, gdy przestrzeń wyszukiwań jest dyskretna. W przypadku problemów, w których znalezienie przybliżonego optimum lokalnego w ustalonym czasie, symulowane wyżarzanie może być lepsze.

1.2.1. Ważne parametry algorytmu

Działanie algorytmu zależy od kilku parametrów, są to:

- początkowa temperatura T
- funkcja schładzania F
- funkcja prawdopodobieństwa P

Jako temperaturę początkową przyjęto koszt wygenerowanego rozwiązania początkowego pomnożonego przez przyjęty z góry współczynnik (w programie wynosi on 10)

Funkcja schładzania jest realizowana według wzoru $T(i + 1) = a * T(i)$, gdzie a jest stałą. Do badań przyjęto a jako wartości 0.99, 0.95 oraz 0.85.

Funkcja prawdopodobieństwa zależy od aktualnej temperatury T , kosztu aktualnie najlepszego rozwiązania C oraz kosztu nowo wygenerowanego rozwiązania C' .

$$P = \exp\left(\frac{C' - C}{T}\right)$$

2. Pomiary

Znając najlepsze znane rozwiązanie dla każdego pliku, algorytmy zostały przetestowane pod względem jakości rozwiązania w danej jednostce czasu.

Dla każdego wyniku został policzony błąd względny $\frac{|f_{obl} - f_{opt}|}{f_{opt}}$, gdzie

f_{obl} – najlepsze obliczone przez algorytm rozwiązanie,

f_{opt} – najlepsze rozwiązanie dla pliku

Dla każdego z plików algorytm został uruchomiony 10 razy i każdy wynik (najlepsza znaleziona ścieżka, czas w którym została znaleziona oraz błąd względny) został zapisany w tabelach poniżej.

Algorytmy zostały przetestowane w następującym wymiarze czasowym:

- 30 sekund dla pliku małego
- 2 minuty dla pliku średniego
- 4 minuty dla pliku dużego

Pliki na których testowany był algorytm: ftv47, ftv170, rgb403.

Algorytm Tabu Search uruchomiony był z dywersyfikacją.

2.1. Wyniki pomiarów

2.1.1. Problem mały

ftv47									
TABU SEARCH									
	SWAP			INSERT			INVERT		
Lp.	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]
1.	1944	21,77	9,45	1894	1,14	6,64	3363	0,14	89,36
2.	1828	16,22	2,92	1798	2,20	1,23	2905	0,76	63,57
3.	1993	21,09	12,21	2330	0,44	31,19	3447	0,75	94,09
4.	1921	29,67	8,16	1817	3,97	2,30	3864	0,37	117,57
5.	1951	16,53	9,85	2335	0,33	31,47	3785	1,00	113,12
6.	1993	26,71	12,21	2014	13,72	13,40	3032	0,36	70,72
7.	1940	9,71	9,23	1971	0,49	10,97	3282	0,13	84,80
8.	1910	27,36	7,54	1878	1,05	5,74	3790	1,63	113,40
9.	1933	20,07	8,84	1907	5,13	7,37	3306	0,77	86,15
10.	1897	6,47	6,81	1997	2,89	12,44	3293	0,76	85,42
SYMULOWANE WYŻARZANIE									
	0.99			0.95			0.85		
Lp.	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]
1.	3145	0,16	77,08	2811	0,03	58,28	2797	0,02	57,49
2.	3193	0,36	79,79	2561	0,02	44,20	2525	0,01	42,17
3.	3081	0,15	73,48	2844	0,05	60,14	2632	0,02	48,20
4.	3324	0,15	87,16	2631	0,01	48,14	2219	0,03	24,94
5.	3177	0,15	78,89	2611	0,02	47,02	2621	0,01	47,58
6.	3305	0,41	86,09	2608	0,03	46,85	2584	0,02	45,50
7.	2998	0,78	68,81	2385	0,03	34,29	2679	0,01	50,84
8.	3604	0,14	102,93	2622	0,01	47,64	2560	0,01	44,14
9.	2793	1,10	57,26	2434	0,02	37,05	2609	0,03	46,90
10.	3177	0,88	78,89	2782	0,02	56,64	2547	0,01	43,41

2.1.2. Problem średni

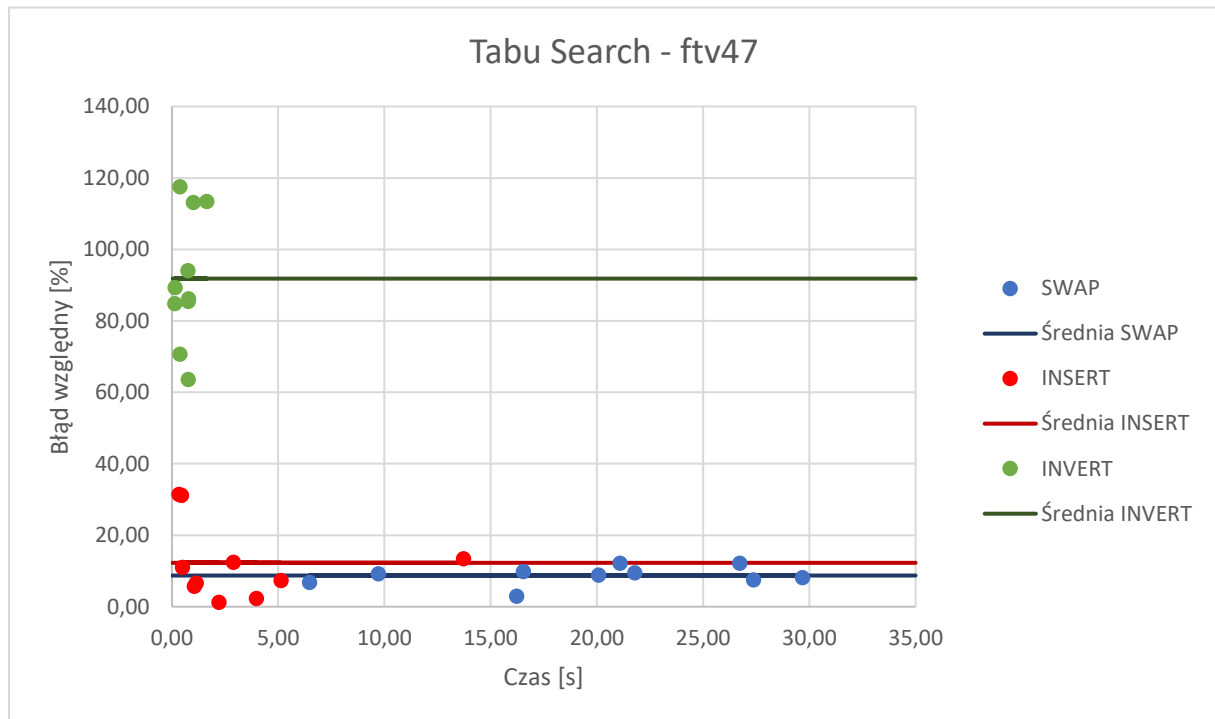
ftv170									
TABU SEARCH									
	SWAP			INSERT			INVERT		
Lp.	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]
1.	6619	120,19	140,25	6484	117,08	135,35	16390	30,17	494,92
2.	6603	120,14	139,67	6551	116,59	137,79	16160	33,66	486,57
3.	6783	94,10	146,21	6801	118,36	146,86	16192	115,59	487,73
4.	6456	118,68	134,34	6746	120,13	144,86	17102	117,42	520,76
5.	6770	116,90	145,74	6783	95,05	146,21	15870	32,47	476,04
6.	6664	113,75	141,89	6774	88,34	145,88	15277	79,92	454,52
7.	6436	110,68	133,61	6708	118,67	143,49	16666	118,08	504,94
8.	6712	75,59	143,63	6818	118,89	147,48	15952	29,48	479,02
9.	6486	109,06	135,43	6387	107,84	131,83	16760	111,31	508,35
10.	6731	119,53	144,32	6856	115,73	148,86	16480	113,81	498,18
SYMULOWANE WYŻARZANIE									
	0.99			0.95			0.85		
Lp.	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]
1.	7039	1,19	155,50	7429	1,66	169,66	7423	1,45	169,44
2.	7183	1,74	160,73	7489	3,46	171,83	7275	1,36	164,07
3.	7474	1,00	171,29	7369	1,98	167,48	7610	2,48	176,23
4.	7593	2,36	175,61	7032	2,13	155,25	7689	1,91	179,09
5.	7975	0,87	189,47	7610	2,25	176,23	7377	2,31	167,77
6.	7363	1,65	167,26	8290	1,35	200,91	7257	1,46	163,41
7.	7277	1,14	164,14	7470	2,26	171,14	7287	2,46	164,50
8.	7252	1,65	163,23	7591	2,11	175,54	7665	2,43	178,22
9.	7336	1,67	166,28	7344	3,06	166,57	7050	2,47	155,90
10.	7551	2,32	174,08	7183	3,14	160,73	7219	1,36	162,03

2.1.3. Problem duży

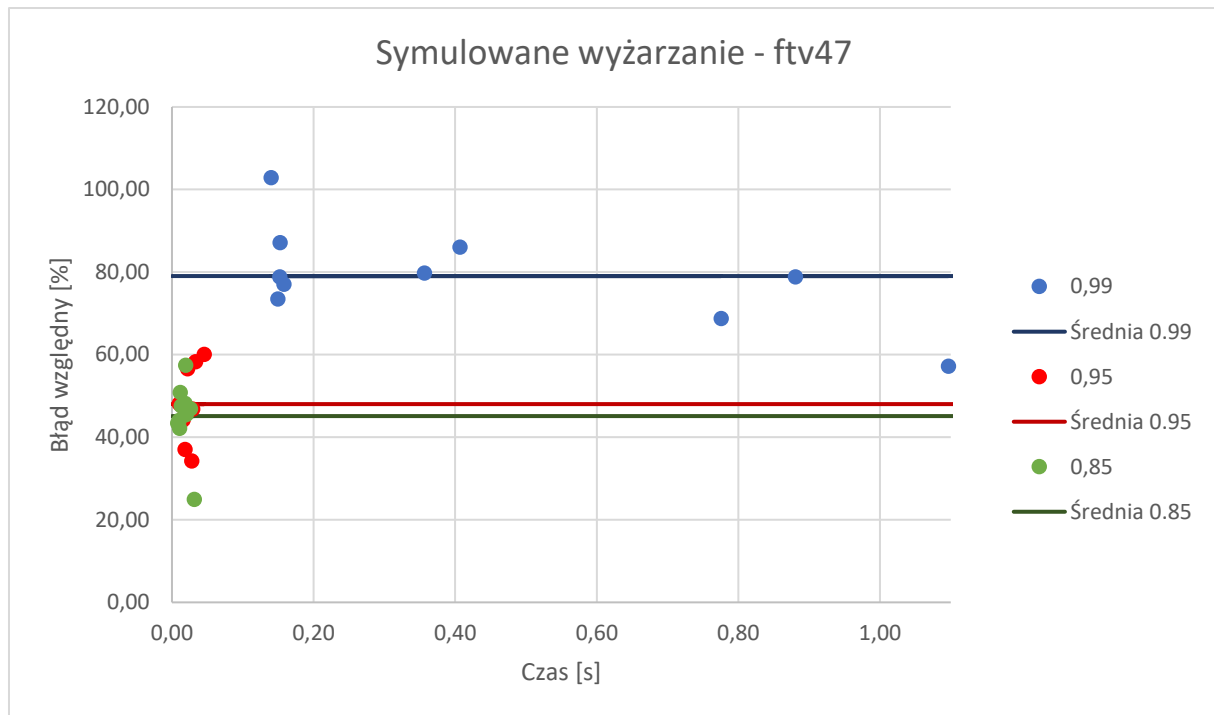
fbg403									
TABU SEARCH									
	SWAP			INSERT			INVERT		
Lp.	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]
1.	5566	241,94	125,80	5652	246,14	129,29	6170	245,14	150,30
2.	5245	243,75	112,78	5787	240,75	134,77	5985	245,08	142,80
3.	5328	244,39	116,15	5603	246,37	127,30	5948	241,17	141,30
4.	5168	244,50	109,66	5615	241,42	127,79	6073	246,14	146,37
5.	4900	240,97	98,78	5432	243,91	120,37	5643	240,08	128,93
6.	4079	242,22	65,48	4996	242,89	102,68	5782	244,25	134,56
7.	4058	242,20	64,62	4925	242,85	99,80	5646	244,23	129,05
8.	4059	240,38	64,67	4737	242,84	92,17	5806	244,32	135,54
9.	4082	241,88	65,60	4834	243,00	96,11	5546	241,98	124,99
10.	4379	240,94	77,65	5001	242,71	102,88	5528	241,65	124,26
SYMULOWANE WYŻARZANIE									
	0.99			0.95			0.85		
Lp.	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]	Koszt	Czas [s]	Błąd [%]
1.	2746	17,41	11,40	2777	18,82	12,66	2771	16,20	12,41
2.	2721	14,52	10,39	2768	16,17	12,29	2729	12,68	10,71
3.	2825	10,73	14,60	2718	15,54	10,26	2761	14,93	12,01
4.	2714	9,64	10,10	2723	21,96	10,47	2803	14,94	13,71
5.	2785	9,17	12,98	2742	12,14	11,24	2764	7,90	12,13
6.	2746	7,26	11,40	2734	22,34	10,91	2832	6,74	14,89
7.	2719	11,87	10,30	2805	15,45	13,79	2755	21,78	11,76
8.	2755	18,34	11,76	2763	9,65	12,09	2763	11,89	12,09
9.	2730	13,74	10,75	2762	14,20	12,05	2741	7,34	11,20
10.	2778	10,41	12,70	2731	15,65	10,79	2713	7,83	10,06

2.2. Analiza pomiarów

2.2.1. Problem mały



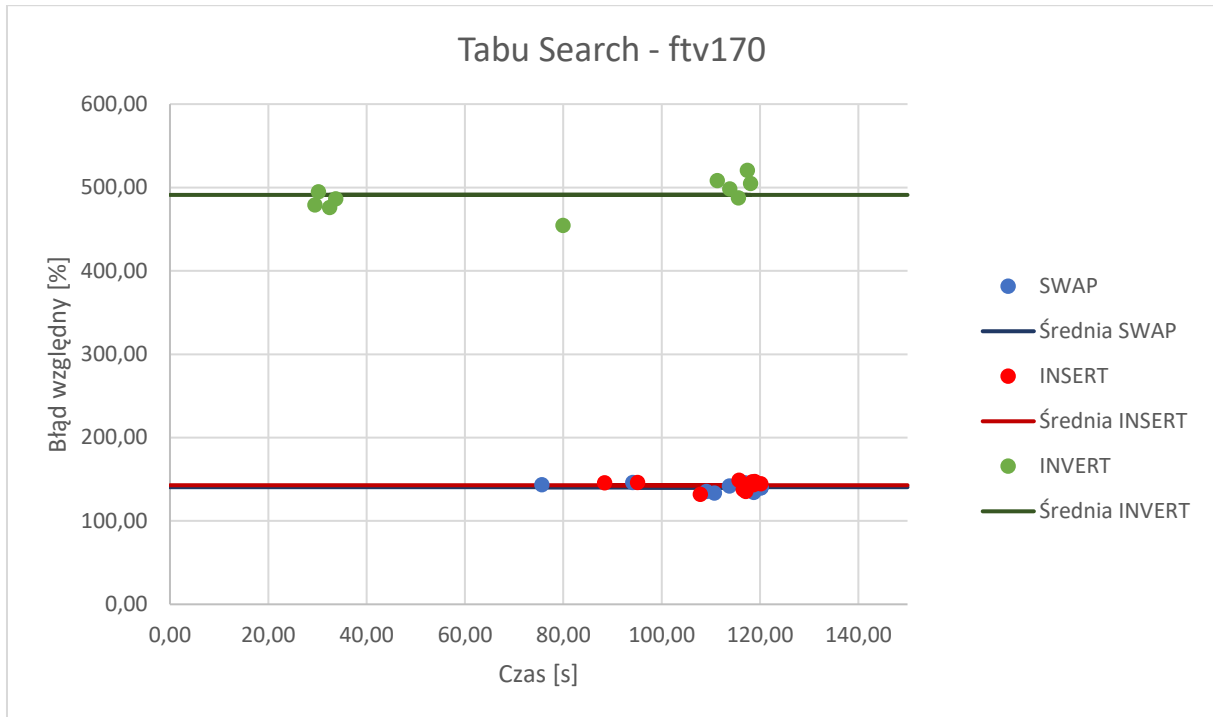
Z powyższych pomiarów wynika, że w algorytmie przeszukiwań tabu dla definicji sąsiedztwa SWAP oraz INSERT można uzyskać wyniki zbliżone do optymalnego. Sąsiedztwo INVERT wypadło w tym przypadku zdecydowanie najgorzej, uzyskując wyniki niemal dwukrotnie większe od najlepszego. Warto zauważyć, że wynik INSERT wypadł nieco gorzej od SWAP, jednak rozwiązania zostały znalezione w średnio 3 sekundy w tej pierwszej definicji, natomiast około 20 sekund w drugiej.



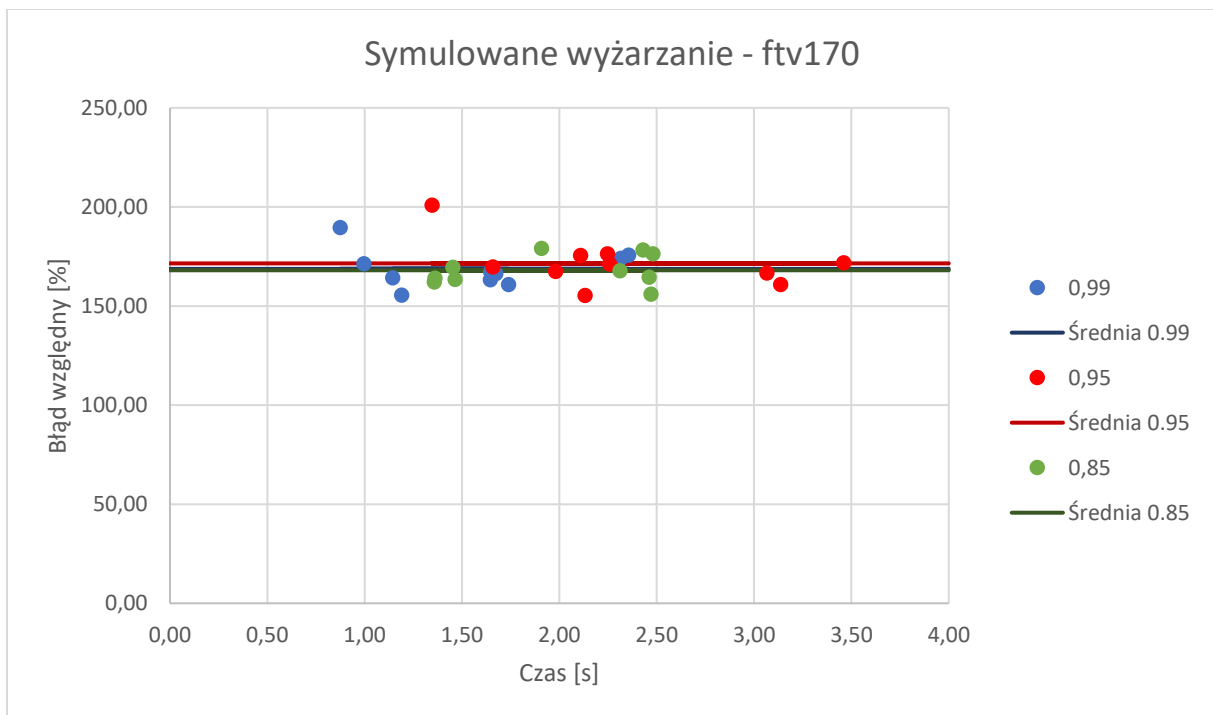
W przypadku algorytmu symulowanego wyżarzania, gdzie zwycięzcami można określić współczynniki chłodzenia 0.95 oraz 0.85, które zwracały najlepsze wyniki prawie natychmiast. Błędy współczynnika 0.99 były średnio dwukrotnie większe, w dodatku rezultat był zwracany około 20 razy później.

Porównując oba algorytmy dla małych problemów, wybór jednej z metod jest zależna od tego, czy wyniki chcemy uzyskać bardziej dokładne, czy chcemy je uzyskać szybciej. Najwcześniej pojawiały się rezultaty dla metody symulowanego wyżarzania dla współczynnika 0.85, gdzie średnia czasu była równa 0,02 sekundy, natomiast najlepsze rozwiązanie otrzymano dla algorytmu przeszukiwań tabu dla metody sąsiedztwa SWAP, gdzie błąd względny wynosił zaledwie 8,72 w około 20 sekund.

2.2.2. Problem średni



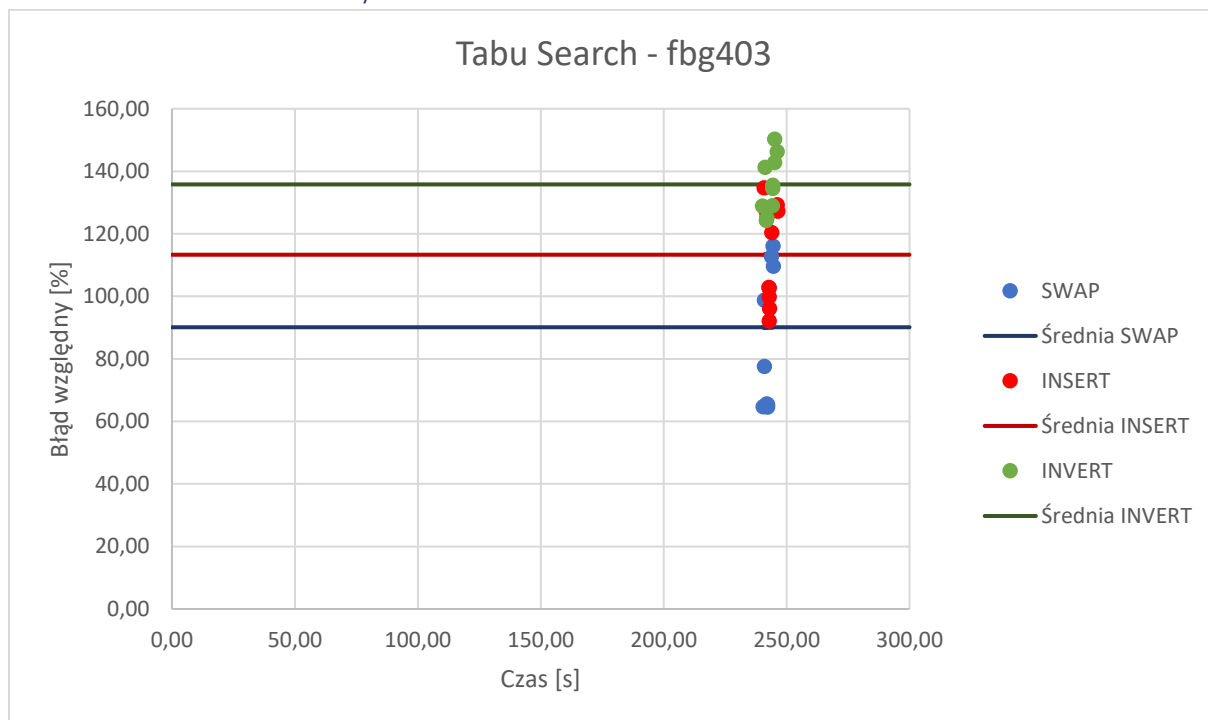
Wyniki w przypadku algorytmu tabu search były bardzo podobne do poprzednich. Większość wyników skupiała się przy określonym limicie czasu działania algorytmu, więc bardzo prawdopodobne jest, że możliwe do uzyskania jest jeszcze lepsze rozwiązanie.



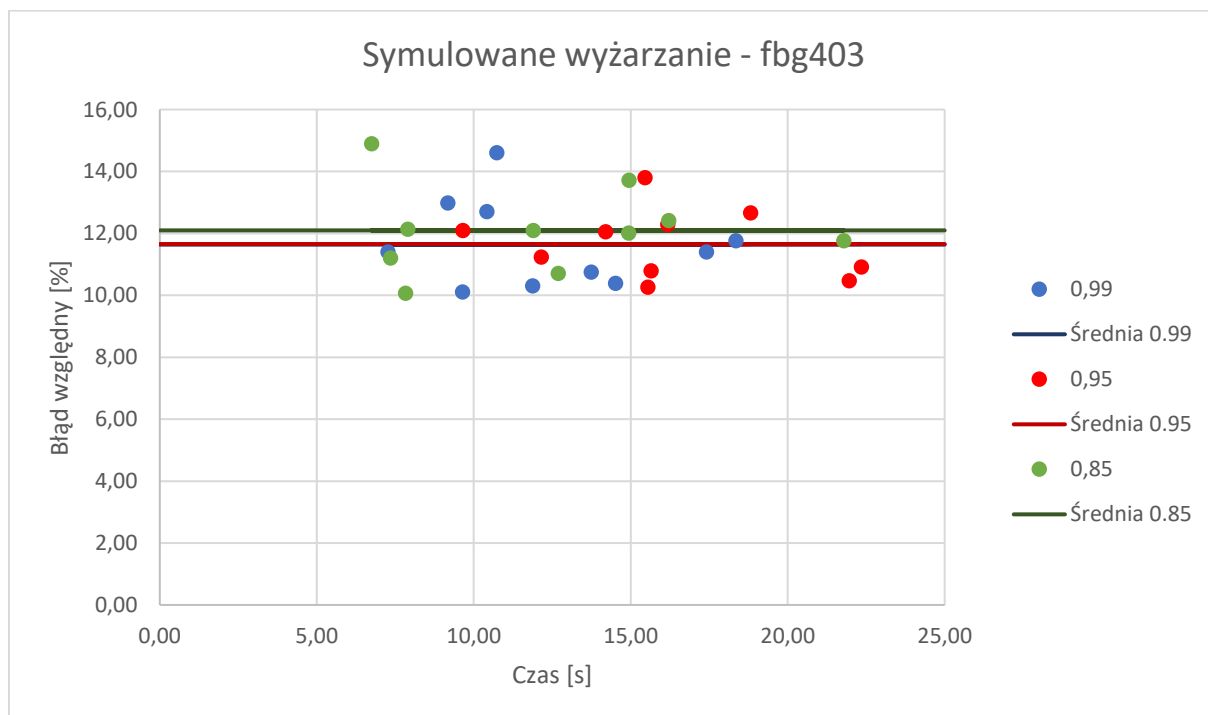
Wszystkie współczynniki wychładzania brane pod uwagę w przypadku algorytmu symulowanego wyżarzania otrzymywały bardzo zbliżone wyniki. Jedyną zauważalną różnicą jest średnia czasu dla najlepszych rozwiązań, które różniły się od siebie o około 0,5 sekundy. Najlepiej wypadł 0.99, następnie 0.85 i na końcu 0.95.

Podobnie jak w poprzednim przypadku, algorytm symulowanego wyżarzania zwracał wyniki dużo szybciej od swojego poprzednika, jednak jego wyniki są znacznie gorsze.

2.2.3. Problem duży



Dla dużego problemu widać znaczną różnicę między definicjami sąsiedztwa. Podobnie jak dla problemu średniego, średnia czasu dla każdej metody jest zbliżona do limitu czasu działania programu. Jako zwycięzcę, można określić definicję SWAP, która otrzymywała najlepsze wyniki.



Algorytm symulowanego wyżarzania po raz kolejny zwracał wyniki znacznie szybciej od algorytmu tabu search oraz tym razem wyniki są bardzo dokładne (można je porównać do wyników algorytmu

tabu dla metod INSERT oraz SWAP dla problemu małego). Dla każdego współczynnika chłodzenia rezultat ponownie jest bardzo zbliżony.

3. Podsumowanie i wnioski

W działaniu algorytmu przeszukiwań tabu dużą rolę odgrywa definicja sąsiedztwa wierzchołków. Metoda INVERT zwracała zdecydowanie najgorsze wyniki, następnie porównywalne metody SWAP (bardziej dokładna, znacznie wolniejsza) oraz INSERT (nieco mniej dokładna, błyskawiczna). Związane jest to z tym, że INVERT wprowadza najwięcej zmian w aktualnej ścieżce, co utrudnia znalezienie lokalnego minimum.

Algorytm symulowanego wyżarzania wyniki otrzymywał bardzo szybko, co związane jest ze współczynnikiem chłodzenia, który z każdą iteracją zmniejsza aktualną temperaturę. Wraz ze spadkiem ciepła, szansa na wybór innej (losowej) ścieżki z określonym prawdopodobieństwem maleje bardzo szybko, aż w pewnym momencie jest wręcz nieosiągalna.

Dla małego problemu najlepszy okazał się algorytm przeszukiwań tabu z metodą sąsiedztwa INSERT, który po krótkiej chwili zwracał bardzo dokładne wyniki.

W przypadku problemu średniego, wybór algorytmu zależałby czy ważniejsza jest jakość otrzymanego rozwiązania, czy czas ich otrzymania. Jeśli interesuje nas lepsze rozwiązanie, powinniśmy wybrać algorytm przeszukiwań tabu, natomiast rezultaty algorytmu symulowanego wyżarzania pojawiały się znacznie wcześniej.

4. Źródła

- dr hab. Piotr Zieliński, „Metody przeszukiwania lokalnego”, 2013, <https://cs.pwr.edu.pl/zielinski/lectures/om/localsearch.pdf>, 17.12.2018
- dr hab. Urszula Boryczka, „Algorytm symulowanego wyżarzania”, http://155.158.112.25/~algorytmyewolucyjne/materialy/algorytm_symulowanego_wyjarzania.pdf, 17.12.2018
- Autor nieznany, „Tabu Search (Poszukiwanie z zakazami)”, 2016, http://www.zio.iia.pwr.wroc.pl/pea/w5_ts.pdf, 18.12.2018
- dr Maciej Hapke, „Kryteria aspiracji”, 2002, <http://www.cs.put.poznan.pl/mhapke/TO-TS2.pdf>, 18.12.2018