Jakub Balcerzak

Projektowanie Efektywnych Algorytmów

Termin zajęć: czwartek godz. 7:30

Prowadzący:

Zadanie projektowe 2

- sprawozdanie

0. Opis zadania projektowego:

Celem projektu była implementacja oraz analiza efektywności algorytmu metaheurystycznego. Wybranym algorytmem jest symulowane wyżarzanie zastosowane do rozwiązania problemu komiwojażera.

1. Wstęp teoretyczny:

W metalurgii zauważono, że w powolnym i stopniowym procesie ochładzania i stygnięcia metalu, cząsteczki jego ciała oddając energię, rozkładają się w sposób bardziej systematyczny, co wpływa na utworzenie równomiernej struktury. Z kolei, gdy spadek temperatury dla danego metalu jest zbyt szybki, to te same cząsteczki rozkładają się chaotycznie. Wzorem, który opisuje zaprezentowane zjawisko jest wzór:

$$P(E) = e^{\frac{-E}{kT}}, \text{ gdzie:}$$

k – stała Boltzmanna
T – temperatura

Przełożenia zaobserwowanego zjawiska na podstawowy algorytm symulowanego wyżarzania dokonał Metropolis w 1953 r. Algorytm ten był rozwinięciem metod iteracyjnych, które polegały na ciągłym ulepszaniu rozwiązaniu, aż do momentu, gdy nie dało się go już bardziej poprawić. Główną wadą tychże metod był fakt, że mogły one „utknąć" w miejscu pseudo-optymalnym, stanowiącym jedynie minimum lokalne. Nie posiadały one możliwości na wyjście z minimum lokalnego, aby kontynuować poszukiwania globalnego minimum.

Algorytm symulowanego wyżarzania (ang. simmulated annealing [SA]) posiada możliwość wyboru rozwiązania gorszego. Dzieje się to oczywiście z pewnym prawdopodobieństwem. Pozwala to właśnie na wyjście z minimum lokalnego, co niemożliwe było w przypadku wspomnianych metod iteracyjnych, a tym samym umożliwia dalszą eksplorację w poszukiwaniu rozwiązania optymalnego globalnie.

Na prawdopodobieństwo, które określa możliwość zaakceptowania rozwiązania gorszego wpływa bezpośrednio parametr przeniesiony z termodynamiki – temperatura. Im większa, tym p-p przyjęcia rozwiązania gorszego jest większe. Im niższa, tym algorytm bardziej będzie przypominał typowe metody iteracyjne. Zatem, jak widać, kolejnym istotnym elementem algorytmu jest proces ochładzania.

Na początku działania algorytmu, gdy temperatura jest wysoka, rozwiązanie ulega częstym zmianom. Etap ten nazywany jest eksploracją. Wraz z kolejnymi iteracjami, a tym samym spadkiem temperatury dochodzi do momentu, gdy nowe rozwiązania wybierane są coraz rzadziej i rzadziej. Algorytm SA zaczyna wtenczas działać, jak algorytm iteracyjny, a jego celem będzie maksymalne ulepszenie rozwiązania. Rozpoczyna się etap eksploatacji.

Schemat algorytmu SA w podstawowej wersji przedstawia się następująco:

```
1. wyznacz rozwiązanie początkowe s
2. wyznacz temperaturę początkową t
3. while(kryterium stopu)
4.   for i = 0 to L
5.        wyznacz losowo rozwiązanie sąsiednie s' ∈ N(s)
6.        Δ = E(s') – E(s)
7.        if (Δ<0)
8.             s = s'
9.        else
10.            wylosuj liczbę 'x' z zakresu [0,1)
11.            if ( x < e^(-Δ/t) )
12.                 s = s'
13.  t = α(t)
14. return s
```

Jak widać w powyższym schemacie mamy do czynienia z 5 parametrami:

- rozwiązanie początkowe s – istnieje wiele sposobów wyznaczania rozwiązania początkowego m. in. iteracyjny lub zachłanny. Sposób ten dobierany jest doświadczalnie,
- temperatura początkowa t – na wyznaczenie temperatury początkowej istnieje wiele sposobów. Jednym z nich (najprostszym) jest pomnożenie wartości kosztu rozwiązania początkowego przez z góry przyjęty współczynnik. Współczynnik ten staje się kolejnym parametrem, którym należy doświadczalnie moderować,
- długość epoki L – wyznaczana eksperymentalnie. Im dłuższa epoka tym algorytm działa dokładnie, lecz zwiększa to czas jego wykonywania.
- kryterium stopu – w zależności od problemu oraz od kontekstu mogą być nim m. in. osiągnięcie temperatury minimalnej, przekroczenie czasu wykonywania się algorytmu. W obu przypadkach oznacza to wprowadzenie kolejnego parametru,
- współczynnik schładzania α – najprostszą metodą schładzania jest schładzanie gradientowe, które prezentuje zależność $T = \alpha T$.

Dodatkowo definiuje się trzy podstawowe sposoby wyznaczania nowego sąsiedniego rozwiązania: sąsiedztwo typu insert (przestawienie elementu z pozycji i-tej na j-tą), typu swap (zamiana elementów na pozycjach i-tej i j-tej) oraz typu invert (odwrócenie elementów pomiędzy elementem i-tym i j-tym włącznie z elementami brzegowymi).

2. Opis implementacji:

W celu wprowadzenia kryterium stopu, jako czasu wykonywania się algorytmu zaimplementowano klasę Timer. Obiekt tej klasy po wywołaniu metody startTime() rozpoczyna odliczanie (analogicznie, jak stoper). Metoda getTime() pozwala uzyskać dokładny czas wyrażony w sekundach. Metoda getIntTime() jest szczególnym przypadkiem powyższej metody, zwraca ona czas również wyrażony w sekundach, lecz bez dodatkowych miejsc po przecinku. Funkcja ta została użyta przy wyświetlaniu wyników bieżących, co sekundę, w trakcie wykonywania się algorytmu SA.

Algorytm symulowanego wyżarzania dla problemu komiwojażera jest zaimplementowany w klasie SimmulatedAnnealingTSP. Użytkownik ma dostęp do następujących metod publicznych:

- readGraphFromFile(filename) – funkcja wczytująca graf z pliku o nazwie filename,
- setParameters() – funkcja wywołująca menu modyfikujące parametry algorytmu,
- setNeighbourhood() – funkcja wywołująca menu wybory typu sąsiedztwa,
- simmulateAnneal() – funkcja realizująca algorytm symulowanego wyżarzania,
- getBestRoute(d)/getCurrentRoute(d) – funkcje zwracające odpowiednio najlepsze znalezione rozwiązanie oraz rozwiązanie przyjęte przed osiągnieciem kryterium stopu – rozwiązanie bieżące.

Zmienne prywatne zawarte wewnątrz klasy potrzebne do prawidłowego działania algorytmu SA to:

- bestOrder/bestOrderDistance – odpowiednio: wyznaczone najlepsze rozwiązanie oraz jego koszt. Przez najlepsze rozwiązanie rozumie się tu, że jest to rozwiązanie, które zostało odnalezione podczas działania algorytmu, aż do osiągnięcia kryterium stopu,
- currentOrder/currentOrderDistance – odpowiednio: bieżące rozwiązanie oraz jego koszt. Przez bieżące rozwiązanie rozumie się tu, że jest to rozwiązanie które zostało wyznaczone na danym etapie działania algorytmu,
- newOrder/newOrderDistance – odpowiednio: nowe rozwiązanie oraz jego koszt. Nowe rozwiązanie wyznaczane jest przez wybrany typ sąsiedztwa (insert, swap lub invert).
- phi – temperatura wyznaczana jest wg wzoru: t = phi * (koszt rozwiązania początkowego)

Metody prywatne wykorzystywane przez SA:

- setInitialOrder() – funkcja wyznaczająca rozwiązanie początkowe. Wyznaczane jest ono strategią zachłanną,
- setInitialTemperature() – funkcja wyznaczająca temperaturę początkową.

W programie zaimplementowano trzy typy sąsiedztwa oraz funkcję liczącą długość trasy dla wyznaczonego nowego porządku:

- insert() – funkcja wyznaczająca nowe rozwiązanie dla sąsiedztwa typu insert,
- swap() – funkcja wyznaczająca nowe rozwiązanie dla sąsiedztwa typu swap,
- invert() – funkcja wyznaczająca nowe rozwiązanie dla sąsiedztwa typu invert,
- countRouteLength(order) – funkcja obliczająca koszt podanego rozwiązania order.

Pozostałe niewymienione wyżej zmienne oraz metody są odpowiednio okomentowane w kodzie źródłowym programu.

Należy również dodać, że obok czasu, jako kryterium stopu wprowadzono w programie dodatkowy warunek końcowy, jakim jest temperatura końcowa. Aby algorytm nie osiągał jej zbyt szybko musi być ona odpowiednio niska. Temperatura końcowa została zaszyta w kodzie programu i ma wartość: 1^{-15}.

3. Pomiary:

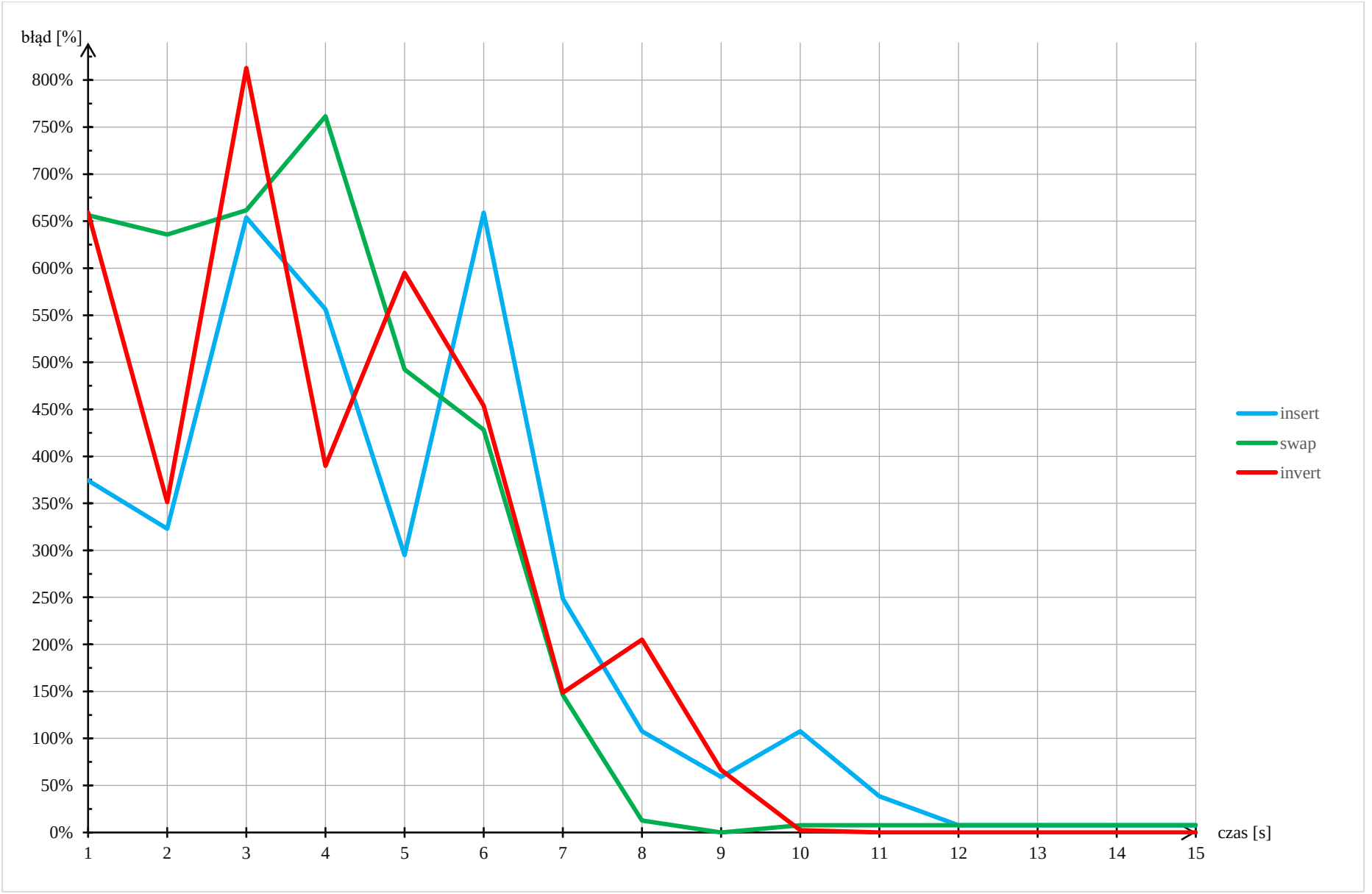
Pomiary przeprowadzono dla trzech plików: tsp_17.txt, tsp_56.txt, tsp_171.txt. Cyfra w nazwie wskazuje na rozmiar instancji problemu. Dla każdego z plików dobierano zestaw parametrów, a następnie obserwowano, jak pomiary będą różnić się w zależności od przyjętej definicji sąsiedztwa. Tabele zawarte w kolejnych punktach zestawiają te różnice prezentując wartości zwracane przez zaimplementowany algorytm, a także błąd względny liczony w kontekście znanego rozwiązania optymalnego. Wykresy ilustrują zmiany błędu względnego (wyrażonego w %) w zależności od czasu.

3.1. tsp_17.txt:

Parametry:

- phi: 100 000
- alpha: 0.99
- L = 40 000
- T = 15 s
- rozw. optymalne = 39

	insert		swap		invert	
czas [s]	koszt	błąd [%]	koszt	błąd [%]	koszt	błąd [%]
1	185	374,36%	295	656,41%	296	658,97%
2	165	323,08%	287	635,90%	176	351,28%
3	294	653,85%	297	661,54%	356	812,82%
4	256	556,41%	336	761,54%	191	389,74%
5	154	294,87%	231	492,31%	271	594,87%
6	296	658,97%	206	428,21%	216	453,85%
7	136	248,72%	96	146,15%	97	148,72%
8	81	107,69%	44	12,82%	119	205,13%
9	62	58,97%	39	0,00%	65	66,67%
10	81	107,69%	42	7,69%	40	2,56%
11	54	38,46%	42	7,69%	39	0,00%
12	42	7,69%	42	7,69%	39	0,00%
13	42	7,69%	42	7,69%	39	0,00%
14	42	7,69%	42	7,69%	39	0,00%
15	42	7,69%	42	7,69%	39	0,00%


błąd [%]
800%
750%
700%
650%
600%
550%
500%
450%
400%
350%
300%
250%
200%
150%
100%
50%
0%
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
czas [s]
insert
swap
invert

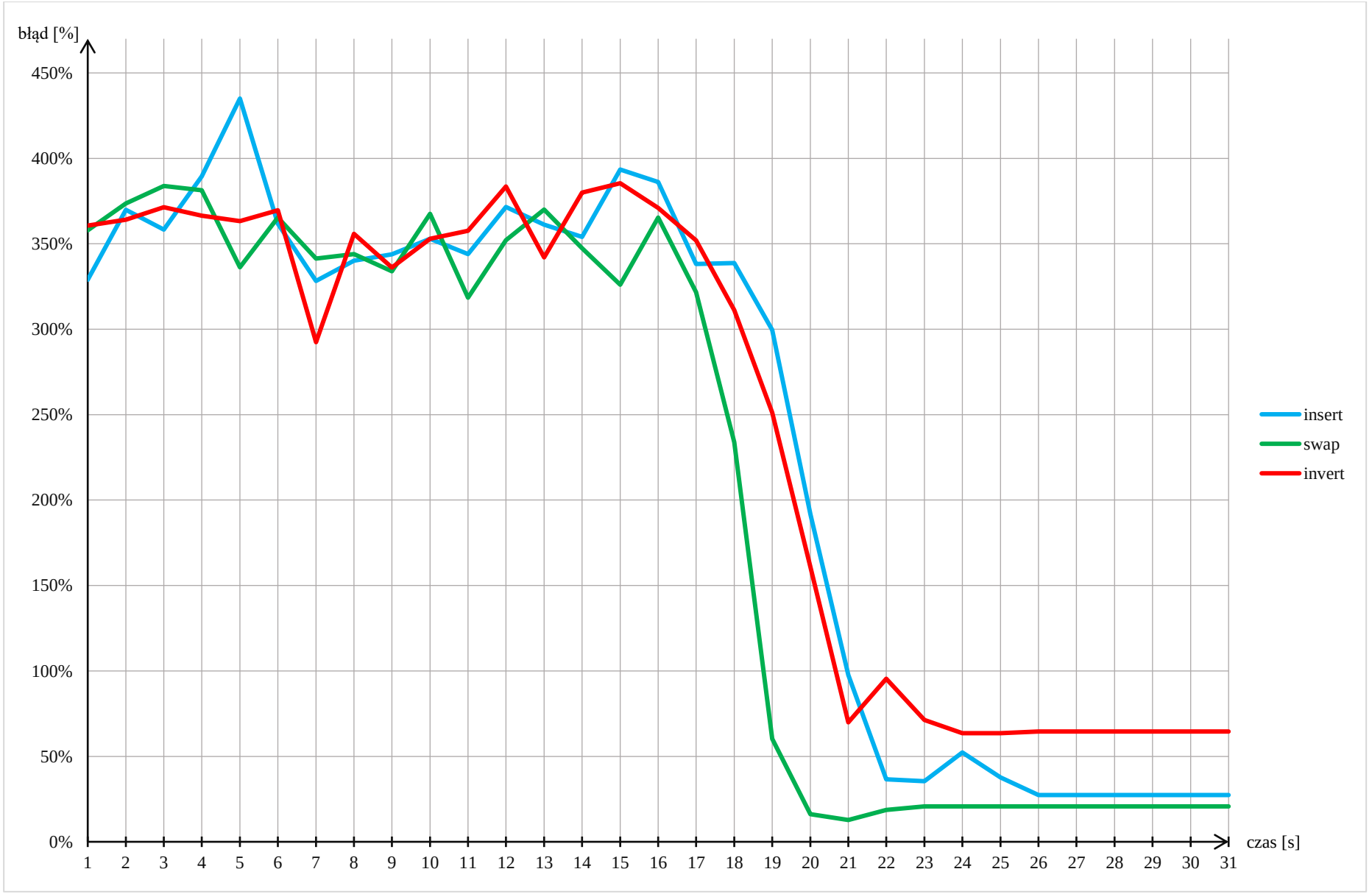
3.2. tsp_56.txt:

Parametry:

- phi: 1 000 000
- alpha: 0.98
- L = 120 000
- T = 31 s
- rozw. optymalne = 1608

	insert		swap		invert	
czas [s]	koszt	błąd [%]	koszt	błąd [%]	koszt	błąd [%]
1	6898	328,98%	7362	357,84%	7407	360,63%
2	7556	369,90%	7615	373,57%	7462	364,05%
3	7370	358,33%	7780	383,83%	7581	371,46%
4	7870	389,43%	7740	381,34%	7500	366,42%
5	8603	435,01%	7014	336,19%	7450	363,31%
6	7424	361,69%	7482	365,30%	7552	369,65%
7	6886	328,23%	7097	341,36%	6309	292,35%
8	7076	340,05%	7138	343,91%	7329	355,78%
9	7136	343,78%	6977	333,89%	7012	336,07%
10	7284	352,99%	7518	367,54%	7283	352,92%
11	7139	343,97%	6729	318,47%	7358	357,59%
12	7583	371,58%	7269	352,05%	7775	383,52%
13	7416	361,19%	7558	370,02%	7109	342,10%
14	7302	354,10%	7195	347,45%	7718	379,98%
15	7934	393,41%	6850	326,00%	7806	385,45%
16	7818	386,19%	7482	365,30%	7574	371,02%
17	7047	338,25%	6781	321,70%	7268	351,99%
18	7055	338,74%	5367	233,77%	6611	311,13%
19	6426	299,63%	2577	60,26%	5647	251,18%
20	4691	191,73%	1869	16,23%	4195	160,88%
21	3178	97,64%	1813	12,75%	2733	69,96%

22	2197	36,63%	1908	18,66%	3142	95,40%
23	2179	35,51%	1942	20,77%	2754	71,27%
24	2448	52,24%	1942	20,77%	2630	63,56%
25	2213	37,62%	1942	20,77%	2630	63,56%
26	2048	27,36%	1942	20,77%	2646	64,55%
27	2048	27,36%	1942	20,77%	2646	64,55%
28	2048	27,36%	1942	20,77%	2646	64,55%
29	2048	27,36%	1942	20,77%	2646	64,55%
30	2048	27,36%	1942	20,77%	2646	64,55%
31	2048	27,36%	1942	20,77%	2646	64,55%


błąd [%]
450%
400%
350%
300%
250%
200%
150%
100%
50%
0%
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
czas [s]
insert
swap
invert

3.3. tsp_171.txt:

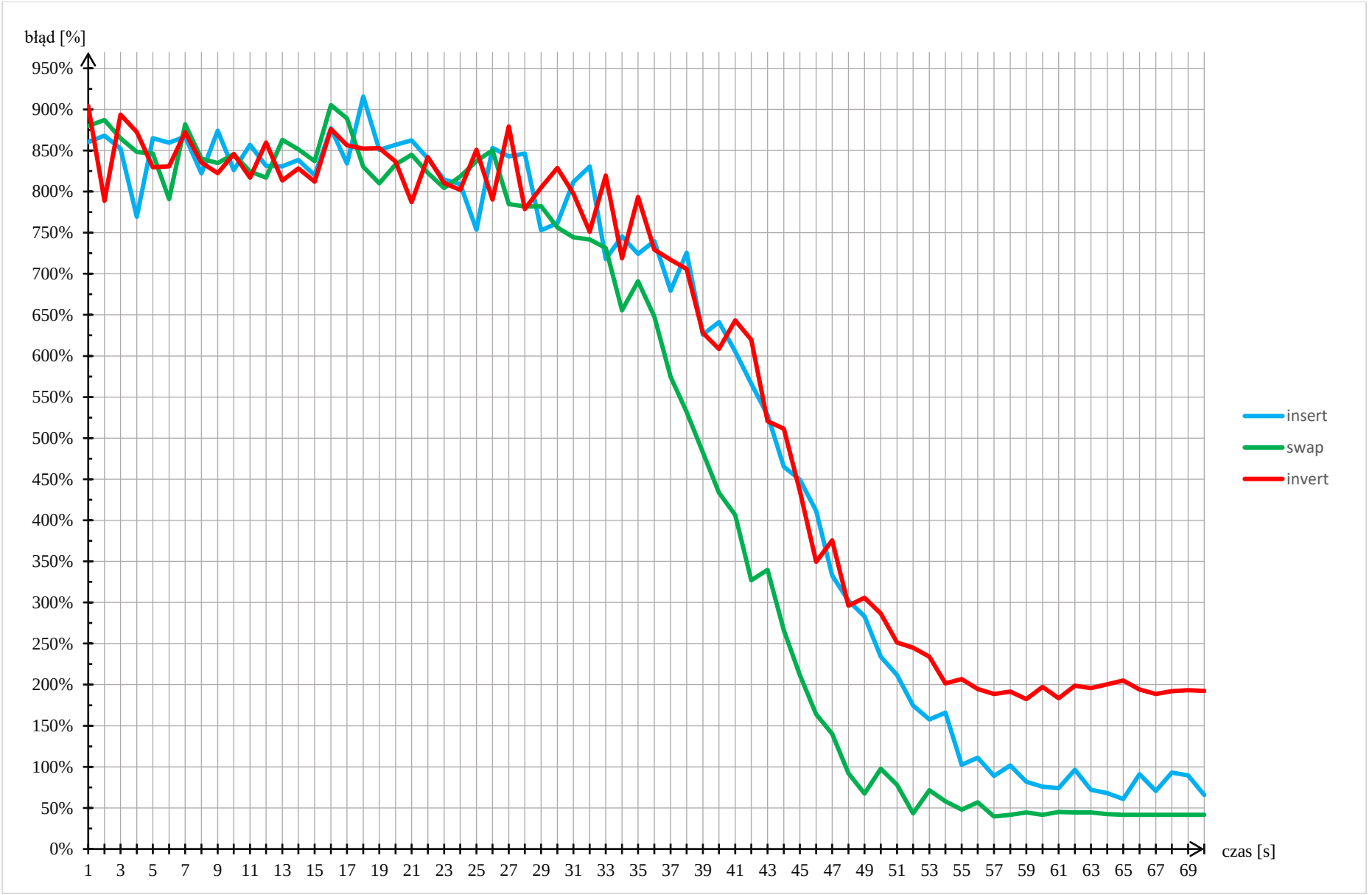
Parametry:

- phi: 10
- alpha: 0.9999
- L = 2 000
- T = 70 s
- rozw. optymalne = 2755

	insert		swap		invert	
czas [s]	koszt	błąd [%]	koszt	błąd [%]	koszt	błąd [%]
1	26457	860,33%	26991	879,71%	27663	904,10%
2	26670	868,06%	27191	886,97%	24488	788,86%
3	26232	852,16%	26586	865,01%	27372	893,54%
4	23947	769,22%	26126	848,31%	26791	872,45%
5	26581	864,83%	26067	846,17%	25612	829,66%
6	26431	859,38%	24541	790,78%	25642	830,74%
7	26635	866,79%	27054	882,00%	26789	872,38%
8	25402	822,03%	25899	840,07%	25769	835,35%
9	26840	874,23%	25754	834,81%	25412	822,40%
10	25515	826,13%	26048	845,48%	26056	845,77%
11	26361	856,84%	25468	824,43%	25260	816,88%
12	25660	831,40%	25266	817,10%	26444	859,85%
13	25644	830,82%	26532	863,05%	25166	813,47%
14	25856	838,51%	26213	851,47%	25567	828,02%
15	25338	819,71%	25820	837,21%	25130	812,16%
16	26915	876,95%	27694	905,23%	26889	876,01%
17	25735	834,12%	27246	888,97%	26348	856,37%
18	27977	915,50%	25625	830,13%	26234	852,23%
19	26199	850,96%	25076	810,20%	26251	852,85%
20	26359	856,77%	25703	832,96%	25804	836,62%
21	26515	862,43%	26033	844,94%	24442	787,19%

22	25904	2840,25%	25421	822,72%	25951	841,96%
23	25194	814,48%	24916	804,39%	25078	810,27%
24	25049	809,22%	25301	818,37%	24854	802,14%
25	23516	753,58%	25820	837,21%	26197	850,89%
26	26261	853,21%	26181	850,31%	24527	790,27%
27	25977	842,90%	24376	784,79%	26983	879,42%
28	26070	846,28%	24297	781,92%	24210	778,77%
29	23500	752,99%	24309	782,36%	24931	804,94%
30	23733	761,45%	23596	756,48%	25583	828,60%
31	25118	811,72%	23263	744,39%	24724	797,42%
32	25631	830,34%	23192	741,81%	23450	751,18%
33	22533	717,89%	22904	731,36%	25335	819,60%
34	23296	745,59%	20820	655,72%	22555	718,69%
35	22707	724,21%	21791	690,96%	24617	793,54%
36	23132	739,64%	20602	647,80%	22850	729,40%
37	21474	679,46%	18585	574,59%	22509	717,02%
38	22744	725,55%	17409	531,91%	22199	705,77%
39	20003	626,06%	16046	482,43%	20060	628,13%
40	20421	641,23%	14700	433,58%	19516	608,38%
41	19428	605,19%	13951	406,39%	20474	643,16%
42	18353	566,17%	11764	327,01%	19830	619,78%
43	17307	528,20%	12112	339,64%	17097	520,58%
44	15571	465,19%	10086	266,10%	16844	511,40%
45	15133	449,29%	8587	211,69%	14754	435,54%
46	14095	411,62%	7275	164,07%	12384	349,51%
47	11921	332,70%	6618	140,22%	13103	375,61%
48	11080	302,18%	5293	92,12%	10912	296,08%
49	10556	283,16%	4614	67,48%	11181	305,84%
50	9203	234,05%	5447	97,71%	10648	286,50%
51	8589	211,76%	4908	78,15%	9678	251,29%

52	7568	174,70%	3946	43,23%	9504	244,97%
53	7095	157,53%	4726	71,54%	9204	234,08%
54	7325	165,88%	4354	58,04%	8309	201,60%
55	5580	102,54%	4074	47,88%	8450	206,72%
56	5814	111,03%	4324	56,95%	8115	194,56%
57	5209	89,07%	3845	39,56%	7951	188,60%
58	5555	101,63%	3904	41,71%	8035	191,65%
59	5007	81,74%	3977	44,36%	7781	182,43%
60	4841	75,72%	3900	41,56%	8190	197,28%
61	4798	74,16%	3998	45,12%	7807	183,38%
62	5409	96,33%	3979	44,43%	8226	198,58%
63	4742	72,12%	3979	44,43%	8147	195,72%
64	4629	68,02%	3927	42,54%	8277	200,44%
65	4432	60,87%	3901	41,60%	8407	205,15%
66	5265	91,11%	3901	41,60%	8100	194,01%
67	4698	70,53%	3901	41,60%	7954	188,71%
68	5319	93,07%	3901	41,60%	8046	192,05%
69	5219	89,44%	3901	41,60%	8081	193,32%
70	4568	65,81%	3901	41,60%	8053	192,30%


błąd [%]
950%
900%
850%
800%
750%
700%
650%
600%
550%
500%
450%
400%
350%
300%
250%
200%
150%
100%
50%
0%
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69
czas [s]
insert
swap
invert

4. Wnioski:

Pierwszy wniosek wiąże się ściśle z metaheurystyczną naturą algorytmu symulowanego wyżarzania i dotyczy on parametrów. Jak wcześniej wspomniano należy rozważyć 5 parametrów: temperaturę początkową – zależną od ustawionego parametru phi, współczynnik chłodzenia (α), długość dekady (L), czas wykonywania lub/i temperaturę końcową będącymi kryterium stopu. Ponieważ algorytm ten jest metaheurystyczny nie ma idealnego „przepisu" na wysterowanie parametrów. Parametry, które zostały wybrane do prezentacji w niniejszym sprawozdaniu dają pewne rozwiązania, które wydają się nie być, aż tak złymi rozwiązaniami, lecz nie wolno tych parametrów nazywać optymalnymi, gdyż, jak widać, dla każdego rozmiaru instancji, dla każdego typu sąsiedztwa, a także dla różnych możliwości sprzętowych będą one inne.

Można zauważyć jednak pewne prawidłowości i tak:

- Współczynnik chłodzenia nie może być zbyt mały, gdyż musi umożliwiać powolne obniżanie temperatury. Niektóre źródła podają dla niego zakres [0.85, 0.99]. Jak widać w projekcie stosowane były nawet wyższe wartości.
- Nie istnieje sposób na wyznaczenie optymalnej lub dobrej temperatury początkowej, a także długości dekady. Parametry te są dobierane doświadczalnie. Należy jednak pamiętać, że wzrost temp. pocz. oraz L pociąga za sobą dłuższy czas wykonywania się algorytmu – dłużej trwa proces wychładzania.
- Tak samo czas, w jakim ma działać SA zależy od parametrów wymienionych powyżej. Jeżeli proces ochładzania trwa długo, wówczas i czas musi wzrosnąć, by algorytm miał szansę przejść z eksploracji do eksploatacji.
- Temperatura końcowa z kolei nie może być za wysoka, aby algorytm mógł swobodnie eksploatować, ale nie może być również za niska, gdyż od pewnego momentu dalsze działanie algorytmu nie będzie przynosić wyraźnego polepszenia wyników, a wręcz stanie on na pewnym rozwiązaniu z b. małą szansą na wyjście z niego.

Drugi wniosek dotyczy wykresów. Dla ustawionych parametrów można obserwować na nich oczekiwane zachowanie. Dla każdego wykresu można wyróżnić trzy fazy:

- faza 1: w tej fazie błąd względny otrzymywanych rozwiązań jest bardzo duży, rzędu kilku setek %. Obserwujemy w tej fazie „szamotanie" się rozwiązań oraz bardzo częste zmiany z rozwiązań lepszych na gorsze. Faza ta inaczej określana jest, jako eksploracja.
- faza 2: gdy temperatura spadnie odpowiednio, a tym samym zmniejszy się prawdopodobieństwo wyborów gorszych, to rozpoczyna się faza eksploatacji. Na wykresach objawia się ona, jako nagły bardzo intensywny spadek wartości błędu względnego (strome zbocze).
- faza 3: gdy temperatura jest już bardzo niska, a rozwiązanie zaczyna oscylować wokół pewnego wyniku.

Co się tyczy otrzymywanych rozwiązań, to widać, że błąd względny wzrastał wraz z rozmiarem instancji problemu. W przypadku n = 17 udało się zejść do zera, czyli otrzymane rozwiązanie było równe rozwiązaniu optymalnemu. Dla n = 56 wyniki uległy pogorszeniu,

najlepszy utrzymywał się na poziomie 20,77 %, a najgorszy 64,55% wartości błędu. Kolejne zwiększenie rozmiaru instancji problemu od 171 przyniosło dalsze pogorszenie otrzymywanych rozwiązań. Najlepszemu udało osiągnąć się 41,60% błędu, podczas, gdy najgorszemu aż 192,30%.

Istotnym czynnikiem jakości rozwiązań okazało się nie tylko odpowiednie wysterowanie parametrów, lecz również wybór sąsiedztwa. Dla najmniejszego rozmiaru problemu najlepszym wyborem okazało się sąsiedztwo typu invert. Najwolniejszym z kolei sąsiedztwo typu insert. Sąsiedztwo typu invert sprawdziło się jednak wyłącznie wówczas, gdy rozmiar problemu był mały. W kolejnych przypadkach tj. 56, 171 dla tego typu sąsiedztwa otrzymywane wyniki charakteryzowały się największym błędem względnym. Najmniejsze błędy przy dużych instancjach pozwalało uzyskać sąsiedztwo typu swap, a po nim typu insert, lecz zdecydowanie od niego wolniejsze, co widać na odpowiednich wykresach.

Ponieważ w trakcie swojego działania algorytm SA może dojść do rozwiązania najlepszego, a następnie utracić je, podstawowy schemat algorytmu na potrzeby implementacji został rozbudowany o przechowywanie rozwiązania najlepszego (bestOrder) na całej rozpiętości czasu działania algorytmu. Zwrócone po zakończeniu się wykonywania koszty rozwiązań najlepszych (bestOrderDistance) pokazały, że algorytmowi często udawało się osiągnąć rozwiązania jeszcze lepsze, niż prezentowane na wykresach. Nieuchwycenie ich na przebiegu czasowym wynika ze sposobu próbkowania rozwiązania co 1 s. Czasem jednak rozwiązania te były rozwiązaniem inicjacyjnym, wyznaczanym strategia zachłanną, jako rozwiązania początkowe. Powyższe uwagi oraz wyniki prezentuje poniższa tabelka:

	typ sąsiedztwa					
plik	insert		swap		invert	
	koszt	błąd [%]	koszt	błąd [%]	koszt	błąd [%]
tsp_56	1773	10,26	1731	7,65	2012*	25,12
tsp_171	3923*	42,40	3618	31,32	3923*	42,40

* - koszt rozwiązania inicjacyjnego (początkowego)