

Implementacja i analiza efektywności algorytmu  
Symulowanego Wyżarzania  
dla asymetrycznego problemu komiwojażera.

Jakub Grzegocki (264009)

grudzień, 2023

Sprawozdanie z drugiego zadania projektowego kursu "Projektowanie efektywnych algorytmów"

# Spis treści

<b>1</b>	<b>Wstęp teoretyczny</b>	<b>3</b>
<b>2</b>	<b>Opis klas</b>	<b>3</b>
2.1	Klasa SymulowaneWyzarzanie . . . . .	3
2.2	Klasa Graph . . . . .	4
<b>3</b>	<b>Plan eksperymentu</b>	<b>5</b>
<b>4</b>	<b>Wyniki badań</b>	<b>5</b>
<b>5</b>	<b>Wykresy</b>	<b>11</b>
<b>6</b>	<b>Wnioski</b>	<b>16</b>
<b>7</b>	<b>Źródła</b>	<b>17</b>

# 1 Wstęp teoretyczny

Symulowane wyżarzanie (Simulated Annealing) to zaawansowany algorytm optymalizacji, który znalazł szerokie zastosowanie w rozwiązywaniu problemów przeszukiwania przestrzeni rozwiązań. Jest to podejście oparte na strategii przeszukiwania sąsiedztwa rozwiązań. Symulowane Wyżarzanie stosuje innowacyjną strategię wyboru rozwiązania sąsiedniego, wybieraną podobnie do zachowania zachłannego.

W kontekście Symulowanego Wyżarzania, rozwiązanie bieżące w danym pokoleniu nie jest wybierane na podstawie kryterium "najlepszego sąsiada", lecz przyjmuje się każdą możliwą nową permutację prowadzącą do poprawy aktualnego rozwiązania. Ta elastyczność w podejściu sprawia, że algorytm ma zdolność akceptowania rozwiązań gorszych, co może okazać się kluczowe, gdy istnieje potencjał znalezienia globalnie optymalnego rozwiązania w okolicy rozwiązania gorszego.

Jednym z kluczowych elementów Symulowanego Wyżarzania jest analogia do procesu wyżarzania metalu, co stanowi istotną różnicę w porównaniu do innych metod optymalizacyjnych. Proces wyżarzania metalu polega na stopniowym obniżaniu temperatury, co prowadzi do zmniejszania ruchliwości cząsteczek metalu i osiągnięcia bardziej stabilnej struktury. W przypadku algorytmu, temperatura pełni rolę kluczowego parametru, wpływającego na proces wyboru rozwiązania w danym etapie.

Na początku działania algorytmu temperatura jest wysoka, co umożliwia częste zmiany konfiguracji rozwiązania, nawet na korzyść tych gorszych. To stadium charakteryzuje się dużą dywersyfikacją, gdzie algorytm bada szeroki zakres możliwych rozwiązań. W miarę postępu iteracji temperatura systematycznie maleje, co sprawia, że algorytm zaczyna preferować wybór rozwiązań lepszych. Ostatecznie, w fazie końcowej, temperatura jest na tyle niska, że prawdopodobieństwo akceptacji rozwiązania gorszego zbliża się do zera, a algorytm staje się bardziej zbliżony do konwencjonalnych metod iteracyjnych, skupiając się na maksymalnym ulepszaniu znalezionej konfiguracji.

Warto zauważyć, że temperatura jest kluczowym regulatorem balansującym między intensyfikacją a dywersyfikacją. Wyższa temperatura sprzyja eksploracji rozwiązań, podczas gdy niższa skupia algorytm na eksploatacji obszaru wokół już zidentyfikowanego rozwiązania. Symulowane Wyżarzanie stanowi zatem elegancką równowagę pomiędzy eksploracją i eksploatacją, co sprawia, że jest to skuteczna metoda optymalizacji w praktyce.

Aby uniknąć utknięcia w lokalnym minimum przestrzeni rozwiązań warto stosować dywersyfikację opartą na losowych permutacjach zamiast zamiany dwóch losowych miast co wzbogaca algorytm Symulowanego Wyżarzania, o większą elastyczność, różnorodność i zdolność do efektywnego przeszukiwania przestrzeni rozwiązań, co może przyczynić się do uzyskania lepszych wyników optymalizacyjnych. Algorytm generując losową permutację co zadaną ilość iteracji zmienia miejsce przestrzeni i eksploruje sąsiadów dla nowego miejsca.

## 2 Opis klas

### 2.1 Klasa SymulowaneWyżarzanie

Klasa `SymulowaneWyżarzanie` jest implementacją algorytmu symulowanego wyżarzania, służącego do rozwiązania problemu komiwojażera.

- Metoda `rozpocznij()` pełni kluczową rolę w tym algorytmie, rozpoczynając proces przeszukiwania przestrzeni permutacji w poszukiwaniu optymalnej ścieżki komiwojażera. Inicjalizuje

permutację początkową, mierzy czas wykonania i iteruje poprzez różne permutacje, starając się minimalizować koszt ścieżki. W trakcie tego procesu schładza temperaturę, akceptując nowe permutacje na podstawie pewnego prawdopodobieństwa.

- Metoda *wypiszWyniki(constvector < int > najlepsza, intwynik, doubleznalezionyCzas, doubletemperatura)* odpowiada za wyświetlanie wyników algorytmu na standardowym wyjściu. Przedstawia koszt najlepszej znalezionej ścieżki, czas jej znalezienia oraz temperaturę w chwili przerwania algorytmu. Dodatkowo prezentuje wyrażenie związane z temperaturą w procesie wyżarzania.
- Kolejna metoda, *permutacjaPocatkowa(intwielkosc)*, generuje początkową permutację dla problemu komiwojażera. Wybiera pierwsze miasto, a następnie dodaje kolejne, wybierając zawsze najbliższe dostępne miasto.
- Metoda *znajdzNajblizszeMiasto(intmiasto, constvector < bool > odwiedzone)* lokalizuje najbliższe nieodwiedzone miasto dla danego miasta na podstawie macierzy odległości.
- Metoda *permutacjaLosowa(intwielkosc)* generuje losową permutację dla problemu komiwojażera poprzez losową permutację numerów miast. *liczTemperature()* oblicza początkową temperaturę algorytmu, opierając się na próbkowaniu losowych permutacji i różnicach w jakości rozwiązań.
- *liczSciezke(vector < int > sciezka)* oblicza koszt ścieżki dla danej permutacji, sumując odległości między kolejnymi miastami. Metoda *zwrocPrawdopodobienstwo(int roznica, double temperatura)* zwraca prawdopodobieństwo akceptacji nowej permutacji w zależności od różnicy w jakości rozwiązań i temperatury.
- Metoda *zapiszSciezke(constvector < int > sciezka1, conststringnazwaPliku)* zapisuje ścieżkę do pliku o podanej nazwie, a *odczytajSciezke(const string nazwaPliku)* odczytuje ścieżkę z pliku.

## 2.2 Klasa Graph

Klasa Graph została zaprojektowana do reprezentacji i manipulacji grafu w kontekście rozwiązywania problemu komiwojażera. Oto krótki opis poszczególnych metod w tej klasie:

- Metoda *ustawSciezke(vector<int> sciezka)* służy do ustawienia ścieżki wewnętrznej zmiennej pomocniczej, co umożliwia jej późniejsze wykorzystanie.
- Metoda *odczytajSciezke(const string nazwaPliku)* wczytuje ścieżkę z pliku o podanej nazwie. Odczytane dane zostają zapisane zarówno w zmienną lokalną, jak i w zmiennej pomocniczej.
- Metoda *liczSciezke(vector<int> Najlepsza)* oblicza koszt danej ścieżki komiwojażera na podstawie macierzy odległości między miastami.
- Metoda *wczytajGraf(string plik)* wczytuje dane o grafie z pliku, inicjalizując macierz odległości oraz liczbę miast. Jeśli wczytane dane są nieprawidłowe, zwraca false.

### 3 Plan eksperymentu

Aby ocenić skuteczność zastosowanych algorytmów, przeprowadzono analizę, w której badano błędy względne w zależności od czasu wykonania dla trzech różnych instancji problemu komiwojażera. W każdej z tych instancji, kryterium zakończenia działania algorytmu zostało zdefiniowane jako dopuszczalny czas trwania, który był uzależniony od rozmiaru konkretnej instancji problemu. Do badań użyty został współczynnik zmiany temperatury na poziomie 0,99 aby uzyskać najlepsze wyniki dla każdej instancji. Dla pliku ftv55.atstp ustalono czas na 40 sekund, dla ftv170.astp - minuta, a dla rbg358.atstp - minuta i 20 sekund.

Określanie temperatury początkowej w procesie odbywa się poprzez przeprowadzenie dużej ilości próbek w przestrzeni rozwiązań. Próbkowanie oznacza tutaj generowanie losowego rozwiązania  $f(i)$  i jego sąsiada  $f(j)$ , a następnie obliczenie różnicy  $\Delta(f)$  w jakości między tymi dwoma rozwiązaniami. Na podstawie zebranych próbek oblicza się średnią wartość różnicy, a uzyskany wynik jest następnie używany w formule do wyznaczenia temperatury początkowej. Wstępne prawdopodobieństwo  $p$  zostało określone na 0.99 - wartość ta powinna być wysoka i bliska wartości 1.

Wzór na wyliczenie temperatury początkowej został zaczerpnięty ze źródła pod nr 5 i przyjmuje on postać:

$$T_0 = \frac{-\Delta f}{\ln(p)}$$

### 4 Wyniki badań

plik	ftv55.atstp			
	koszt rozw.	czas [s]	temp k	błąd [%]
ftp55_1	2033	39,754	9.24235	26,4303483
ftp55_2	2129	38,267	151.931	32,4004975
ftp55_3	1967	37,813	34.733	22,3258706
ftp55_4	1981	39,26	3332.7	23,1965174
ftp55_5	1992	37,128	5042.94	23,880597
ftp55_6	1981	39,278	62.08	23,1965174
ftp55_7	2006	34,19	7570.00	24,7512438
ftp55_8	<b>1889</b>	<b>36,016</b>	<b>99.2007</b>	17,4751244
ftp55_9	2005	32,879	181.428	24,6890547
ftp55_10	2096	33,582	24.485	30,3482587

Rysunek 1: Zestawienie błędów względnych oraz czasów odnalezienia dla ftv55.atstp

plik	ftv170.atsp			
	koszt rozw.	czas [s]	temp k	błąd [%]
ftv170_1	3878	59,143	116.26	40,7622505
ftv170_2	3888	58,195	1.610	41,1252269
ftv170_3	3822	59,653	16.3275	38,7295826
ftv170_4	3822	59,137	190.485	38,7295826
ftv170_5	3962	57,069	1561.07	43,8112523
ftv170_6	4056	43,360	3461.74	47,2232305
ftv170_7	<b>3822</b>	<b>55,077</b>	<b>37.729</b>	38,7295826
ftv170_8	4013	52,877	0.316	45,6624319
ftv170_9	3929	59,875	7693.17	42,6134301
ftv170_10	3919	58,790	10.1439	42,2504537

Rysunek 2: Zestawienie błędów względnych oraz czasów odnalezienia dla ftv170.atsp

plik	rgb358.atsp			
	koszt rozw.	czas [s]	temp k	błąd [%]
rgb358_1	1378	77,543	0.567	18,486672
rgb358_2	1368	78.953	8.19559	17,626827
rgb358_3	1402	60.324	50.4259	20,550301
rgb358_4	1420	79.324	0.1455	22,098022
rgb358_5	1384	78.453	0.4986	19,00258
rgb358_6	1440	79.324	81.104	23,817713
rgb358_7	1364	77.940	924.229	17,282889
rgb358_8	1413	77,69	34.9641	21,496131
rgb358_9	1427	78.121	3.489	22,699914
rgb358_10	<b>1361</b>	<b>79.132</b>	<b>0.3332</b>	17,024936

Rysunek 3: Zestawienie błędów względnych oraz czasów odnalezienia dla rgb358.atsp

Najlepsza ścieżka znaleziona dla ftv55.atasp:

33 → 2 → 13 → 35 → 4 → 6 → 5 → 47 → 31 → 46 → 55 → 34 → 1 → 3 → 48 → 29 → 27 → 49  
→ 45 → 44 → 28 → 53 → 43 → 30 → 26 → 25 → 24 → 42 → 21 → 50 → 23 → 54 → 22 → 41  
→ 40 → 20 → 19 → 18 → 39 → 10 → 51 → 14 → 52 → 17 → 16 → 15 → 12 → 32 → 8 → 7 →  
36 → 9 → 37 → 11 → 38

Najlepsza ścieżka znaleziona dla ftv170.atasp:

1 → 2 → 77 → 73 → 170 → 49 → 50 → 51 → 52 → 53 → 43 → 55 → 54 → 58 → 59 → 60 → 61  
→ 68 → 67 → 167 → 70 → 87 → 69 → 85 → 86 → 84 → 71 → 66 → 63 → 64 → 56 → 57 → 62 →  
65 → 88 → 153 → 154 → 89 → 90 → 91 → 94 → 96 → 97 → 99 → 98 → 95 → 92 → 93 → 166 →  
108 → 107 → 106 → 105 → 165 → 163 → 100 → 102 → 103 → 117 → 118 → 119 → 120 → 121 →  
122 → 123 → 162 → 101 → 104 → 114 → 109 → 113 → 164 → 127 → 126 → 125 → 124 → 129  
→ 128 → 130 → 131 → 132 → 133 → 134 → 141 → 6 → 7 → 8 → 9 → 10 → 76 → 74 → 75 → 11  
→ 12 → 18 → 19 → 20 → 158 → 32 → 23 → 26 → 27 → 28 → 29 → 30 → 31 → 33 → 34 → 156  
→ 40 → 39 → 38 → 37 → 35 → 36 → 157 → 41 → 155 → 42 → 45 → 44 → 46 → 47 → 48 → 168  
→ 72 → 78 → 82 → 79 → 80 → 81 → 3 → 4 → 5 → 169 → 111 → 110 → 83 → 22 → 16 → 17  
→ 21 → 24 → 25 → 150 → 160 → 151 → 152 → 142 → 143 → 144 → 140 → 149 → 148 → 147  
→ 137 → 136 → 138 → 135 → 139 → 14 → 13 → 15 → 159 → 161 → 115 → 116 → 146 → 145 → 112

Najlepsza ścieżka znaleziona dla rgb358.atasp:

1 → 21 → 69 → 305 → 40 → 4 → 44 → 91 → 10 → 27 → 16 → 17 → 13 → 3 → 164 → 218 →  
321 → 260 → 55 → 34 → 9 → 263 → 35 → 18 → 352 → 64 → 36 → 29 → 337 → 137 → 37 → 28  
→ 24 → 187 → 205 → 180 → 47 → 79 → 45 → 25 → 12 → 235 → 11 → 15 → 246 → 6 → 81 →  
284 → 247 → 174 → 56 → 23 → 57 → 49 → 78 → 97 → 208 → 63 → 114 → 233 → 19 → 104 →  
98 → 5 → 154 → 124 → 20 → 135 → 39 → 58 → 243 → 51 → 67 → 95 → 52 → 171 → 53 → 59  
→ 125 → 195 → 66 → 14 → 249 → 118 → 146 → 190 → 297 → 99 → 93 → 26 → 112 → 103 →  
72 → 172 → 315 → 183 → 173 → 214 → 219 → 196 → 291 → 77 → 42 → 198 → 76 → 106 →  
181 → 328 → 61 → 86 → 165 → 299 → 144 → 123 → 117 → 89 → 192 → 8 → 100 → 357 → 273  
→ 92 → 275 → 191 → 339 → 230 → 276 → 182 → 207 → 70 → 32 → 31 → 68 → 22 → 87 → 288  
→ 119 → 267 → 75 → 212 → 333 → 122 → 303 → 200 → 126 → 202 → 102 → 210 → 265 → 226  
→ 127 → 131 → 170 → 128 → 293 → 285 → 129 → 185 → 41 → 50 → 227 → 130 → 111 → 101  
→ 132 → 316 → 224 → 134 → 82 → 253 → 136 → 85 → 84 → 33 → 320 → 270 → 329 → 290 →  
138 → 38 → 139 → 271 → 310 → 140 → 110 → 304 → 142 → 109 → 96 → 286 → 107 → 193 →  
153 → 65 → 168 → 266 → 43 → 215 → 308 → 209 → 255 → 176 → 319 → 257 → 155 → 115 →  
344 → 156 → 30 → 197 → 231 → 148 → 158 → 149 → 241 → 189 → 342 → 282 → 274 → 159 →  
121 → 120 → 161 → 152 → 335 → 147 → 162 → 133 → 143 → 163 → 2 → 54 → 151 → 349 →  
141 → 167 → 145 → 289 → 169 → 348 → 343 → 313 → 240 → 46 → 108 → 116 → 175 → 307 →  
258 → 177 → 150 → 216 → 178 → 184 → 223 → 179 → 356 → 331 → 186 → 346 → 312 → 206  
→ 204 → 268 → 234 → 244 → 211 → 262 → 332 → 272 → 194 → 73 → 280 → 203 → 74 → 259  
→ 281 → 324 → 279 → 338 → 213 → 222 → 278 → 251 → 217 → 201 → 166 → 80 → 322 → 340  
→ 301 → 296 → 311 → 245 → 236 → 345 → 94 → 220 → 250 → 113 → 317 → 221 → 199 → 71  
→ 314 → 229 → 309 → 325 → 350 → 300 → 225 → 294 → 298 → 188 → 228 → 327 → 7 → 330  
→ 261 → 242 → 306 → 336 → 60 → 295 → 326 → 277 → 351 → 157 → 248 → 254 → 318 → 323  
→ 302 → 238 → 353 → 287 → 237 → 355 → 105 → 256 → 160 → 269 → 264 → 341 → 62 → 239  
→ 0 → 252 → 292 → 283 → 347 → 83 → 88 → 232 → 48 → 354 → 334

czas[s]	ftv55	
	rozw.	błąd[%]
20	2033	26,4303483
40	2006	24,7512438
60	1992	23,880597
80	1981	23,1965174
100	1940	20,6467662
120	1967	22,3258706

Rysunek 4: Zestawienie błędów względnych najlepszych rozwiązań i maksymalnego dopuszczalnego czasu



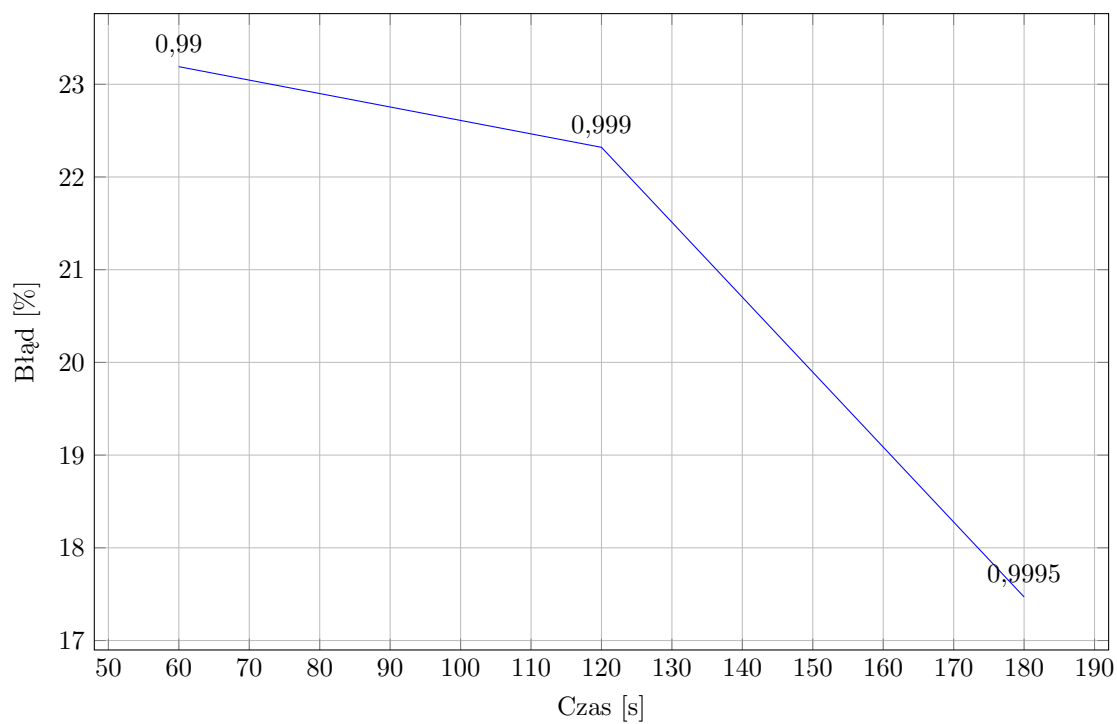
czas[s]	ftv170	
	rozw.	błąd[%]
20	4013	45,6624319
40	3919	42,2504537
60	3878	40,7622505
80	3888	41,1252269
100	3929	42,6134301
120	3822	38,7295826

Rysunek 5: Zestawienie błędów względnych najlepszych rozwiązań i maksymalnego dopuszczalnego czasu

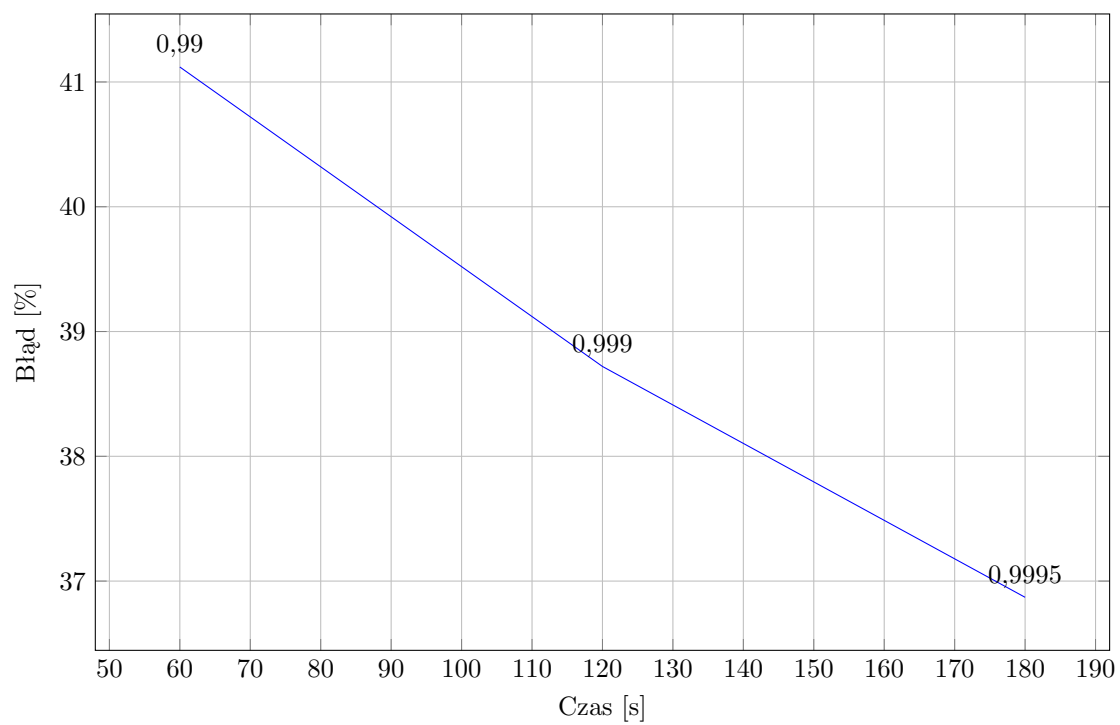
czas[s]	rgb358	
	rozw.	błąd[%]
20	1420	22,0980224
40	1440	23,8177128
60	1427	22,699914
80	1368	17,6268272
100	1378	18,4866724
120	1364	17,2828891

Rysunek 6: Zestawienie błędów względnych najlepszych rozwiązań i maksymalnego dopuszczalnego czasu

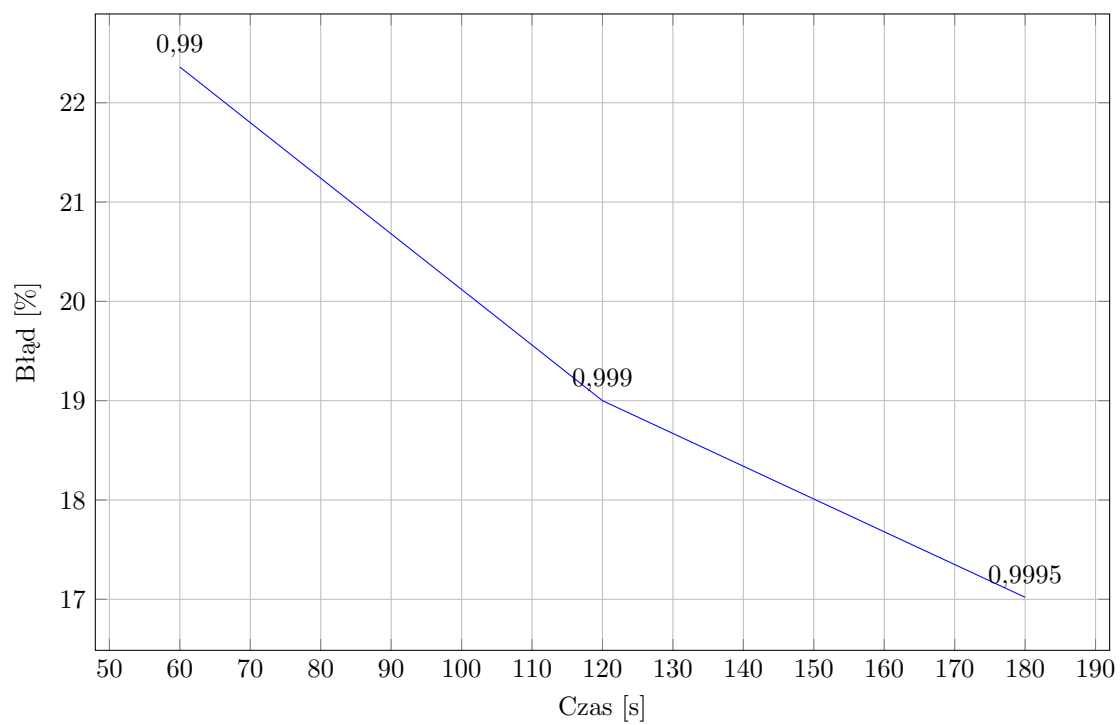
## 5 Wykresy



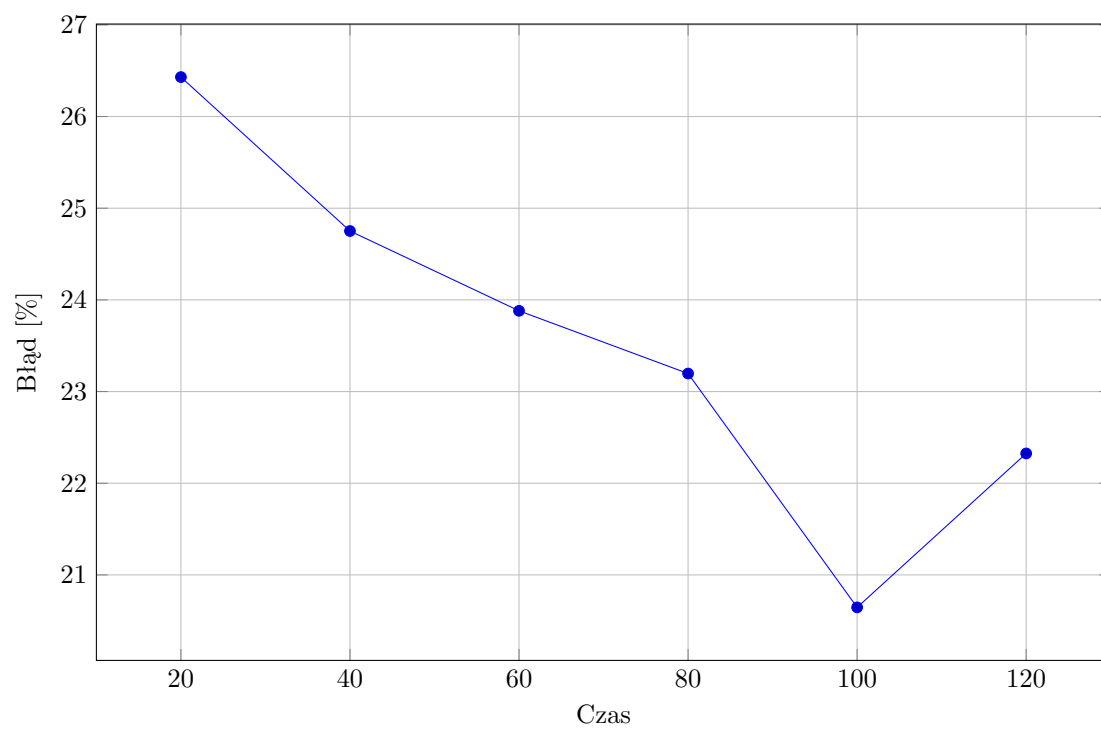
Rysunek 7: Zależność rozwiązania od czasu dla różnych wartości współczynnika  $ftv55$ .



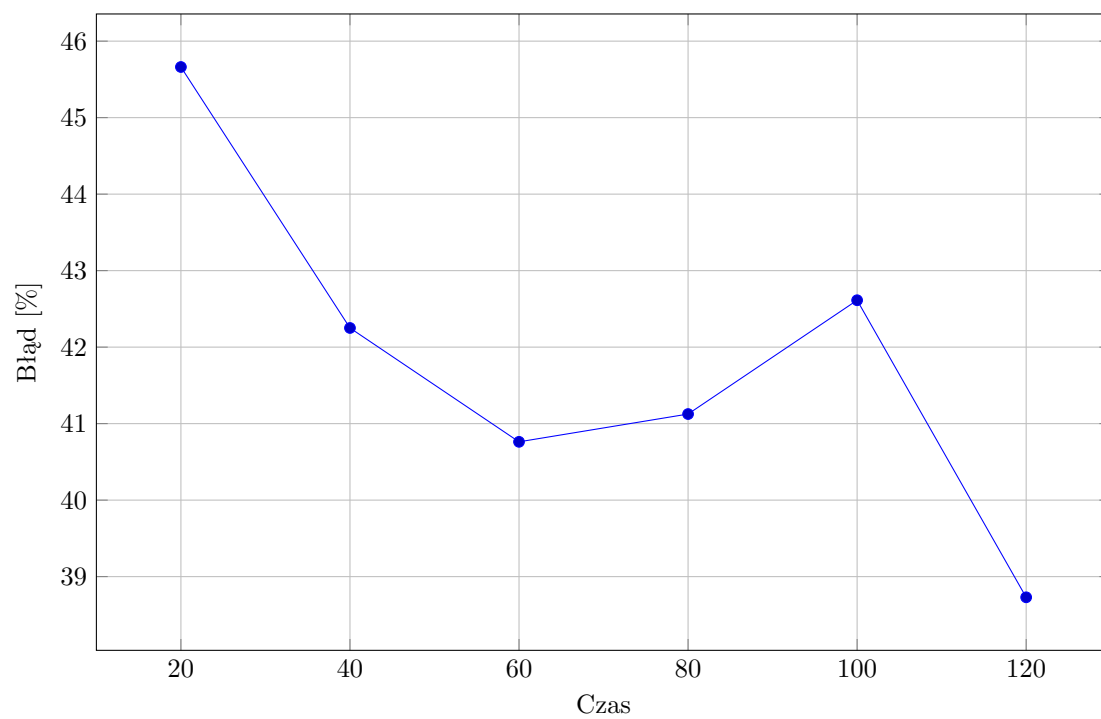
Rysunek 8: Zależność rozwiązania od czasu dla różnych wartości współczynnika  $ftv170$ .



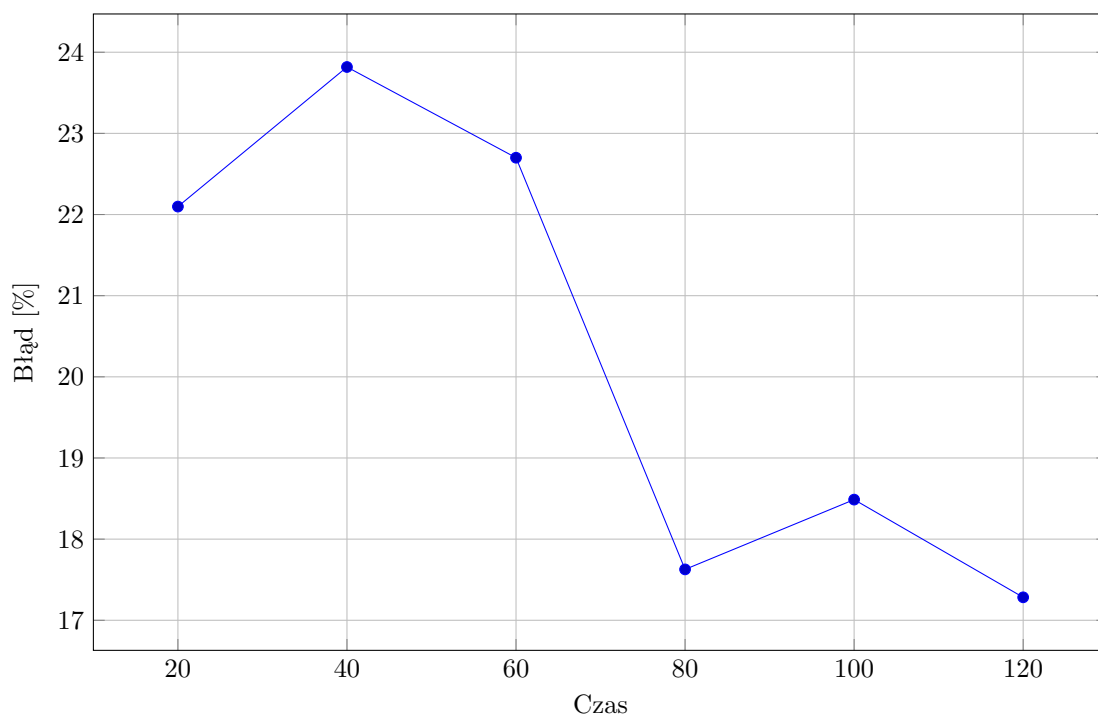
Rysunek 9: Zależność rozwiązania od czasu dla różnych wartości współczynnika rgb358.



Rysunek 10: Wykres liniowy błędu w zależności od czasu dla ftv55.atsp



Rysunek 11: Wykres liniowy błędu w zależności od czasu dla ftv170.atsp



Rysunek 12: Wykres liniowy błędu w zależności od czasu dla rgb358.atsp

## 6 Wnioski

Analiza wyników uzyskanych w trakcie badań Symulowanego Wyżarzania ujawnia istotny wpływ czasu wykonania na jakość dostarczonych rozwiązań. Wynika to z obserwacji wykresów błędu względnego, na których można dostrzec pewne nieregularności. Te nieregularności wynikają z dostosowania przedziałów czasowych do konkretnych instancji problemów. W przypadku małych instancji, algorytm Symulowanego Wyżarzania może szybko znaleźć dobre rozwiązanie już po krótkim czasie działania, co prowadzi do gwałtownego wzrostu jakości rozwiązania. Z kolei dla dużych instancji, krótki przedział czasowy uniemożliwia zaobserwowanie istotnego skoku jakości rozwiązania, ponieważ algorytmowi brakuje czasu na przeszukanie innych obszarów rozwiązań. Ogólnie rzecz biorąc, Symulowane Wyżarzanie prezentuje się jako efektywna metoda o dużym tempie działania, szczególnie dla problemów o większych rozmiarach. Dostosowanie czasu wykonania do konkretnych charakterystyk instancji problemu jest kluczowe dla skutecznej analizy oraz uzyskania pełnej wartości informacyjnej z wyników badań. Możemy z pewnością powiedzieć, że poszukiwanie wystarczająco dobrego rozwiązania w przypadku dużych instancji problemu ATSP jest często lepszym pomysłem, niż przeszukiwanie wszystkich możliwych rozwiązań. Skutkuje to zredukowanym czasem rozwiązywania problemu co dla problemów NP-trudnych o złożoności rzędu  $O(n!)$  jest znaczącym usprawnieniem.



## 7 Źródła

1. <https://www.cs.ubc.ca/~hutter/previous-earg/EmpAlgReadingGroup/TSP-JohMcg97.pdf>
2. [http://www.pi.zarz.agh.edu.pl/int0bl/notes/Int0bl\\_w2.pdf](http://www.pi.zarz.agh.edu.pl/int0bl/notes/Int0bl_w2.pdf)
3. [https://www.youtube.com/watch?v=gX-X85dCib0&t=843s&ab\\_channel=MaciejKomosi%C5%84ski%5Bpl%5D](https://www.youtube.com/watch?v=gX-X85dCib0&t=843s&ab_channel=MaciejKomosi%C5%84ski%5Bpl%5D)
4. [https://www.youtube.com/watch?v=GiDsjiB0VoA&t=693s&ab\\_channel=Reducible](https://www.youtube.com/watch?v=GiDsjiB0VoA&t=693s&ab_channel=Reducible)
5. <https://jakubnowosad.com/ahod/11-simulated-annealing.html#15>
6. Z.Michalkiewicz, D.B.Vogel, Jak to rozwiązać czyli nowoczesna heurystyka, WNT 2006