

Sprawozdanie – Projektowanie Efektywnych Algorytmów – Projekt

*Projekt nr 2 – Implementacja i analiza efektywności algorytmu Tabu Search i/lub Symulowanego
Wyżarzania dla problemu komiwojażera*

Autor:	Adam Czekalski
Nr indeksu:	264488
Termin:	Piątek, godz. 13:15 – 15:00
Prowadzący:	dr inż. Jarosław Mierzwa
Data oddania:	15.12.2023 r.

1. Wstęp teoretyczny

1.1. Pojęcie symulowanego wyżarzania

Nazwa algorytmu „symulowane wyżarzanie” nawiązuje do obróbki termicznej metali, polegającej na nagrzaniu go do konkretnej temperatury, przetrzymaniu go w tej temperaturze, a następnie powolnym schładzaniu. Gdy ciało ma wysoką temperaturę, wtedy cząsteczki mają dużo energii, drgają w sposób chaotyczny i jeśli je zaczniemy powoli schładzać, cząsteczki drgają coraz to słabiej i ustawiają się w optymalnych pozycjach w przestrzeni. Wtedy sieć połączeń staje się dosyć regularna, co może oznaczać, że ciało wyżarzonego materiału może zachowywać się w zupełnie inny sposób.

1.2. Poszczególne elementy i kroki algorytmu

Pierwszym krokiem w tym algorytmie jest wygenerowanie rozwiązania początkowego x_0 . W projekcie zaimplementowano algorytm zachłanny (ang. *Greedy*), który polega na tym, że rozpoczynamy od wierzchołka nr 0 i szukamy nieodwiedzonego miasta najbliższego wierzchołka. Następnie z uprzednio znalezionej wierzchołka znów szukamy najbliższego nieodwiedzonego miasta, i tak dalej. Na końcu następuje powrót do wierzchołka początkowego.

Kolejnym krokiem jest wyznaczenie temperatury początkowej T_0 . Można to zrobić na wiele sposobów, jednak w programie temperatura ta to różnica maksymalnej i minimalnej długości ścieżki w grafie, na podstawie artykułu [5] z literatury. W macierzy wyszukiwane jest minimum oraz maksimum, a następnie jest obliczana ich różnica.

Algorytm SA działa dopóki nie osiągnie tzw. kryterium zatrzymania. Może skończyć swoje działanie, kiedy np. osiągnie określoną liczbę iteracji, nastąpi brak poprawy po określonej liczbie iteracji, liczba zaakceptowanych gorszych rozwiązań będzie poniżej ustalonego poziomu czy osiągnie akceptowalny poziom błędu. W projekcie, kryterium zatrzymania jest czas w sekundach określony przez użytkownika. Kiedy je osiągnie, algorytm jest przerywany i wypisywane jest najlepsze znalezione rozwiązanie wraz z funkcją celu.

Algorytm szuka rozwiązań w danej temperaturze tak długo, dopóki nie osiągnie ustalonej liczby epok. W projekcie, liczba epok uzależniona jest od liczby wierzchołków, w celu proporcjonalnego przeszukiwania sąsiedztwa. Wynosi ona $\frac{n^2}{2}$.

Następny krok to wyznaczenie nowego stanu y w sąsiedztwie x . Polega to na wylosowaniu 2 wierzchołków ze ścieżki i ich zamianę (*swap()*). Potem obliczana jest Δ - różnica między funkcją celu ścieżki powstałej w wyniku *swap'a()* oraz aktualnego najlepszego lokalnego rozwiązania. Jeśli $\Delta \leq 0$ (nowo znalezione rozwiązanie jest lepsze od najlepszego lokalnego rozwiązania), wtedy podmieniamy aktualne lokalne rozwiązanie na lepsze i aktualizujemy lokalną funkcję celu. Dodatkowo jeśli aktualne lokalne rozwiązanie jest lepsze od ogólnie najlepszego rozwiązania, to także je podmieniamy. W algorytmie istnieje też losowy mechanizm prawdopodobieństwa akceptacji gorszego rozwiązania. Jeśli $e^{-\frac{\Delta}{T}} \geq \text{random}(0; 1)$, wtedy także nowo znalezione rozwiązanie jest przyjmowane, mimo że jest gorsze. Pozwala to na „cofnięcie się” do mniej efektywnego rozwiązania, w celu znalezienia bardziej optymalnego, ryzykując przy tym jakość aktualnego najlepszego rozwiązania.

Z każdą epoką, zmniejszana jest temperatura. W projekcie redukowana jest geometrycznie, ze wzoru:

$$T_{k+1} = a * T_k$$

Gdzie:

a – współczynnik zmiany temperatury

T_k – temperatura z poprzedniego kroku

T_{k+1} – temperatura z następnego kroku

2. Opis najważniejszych klas w projekcie

2.1. Klasa SimulatedAnnealing

Pola klasy:

- `std::vector<int> bestSolution` – wektor przechowujący ciąg wierzchołków najlepszego możliwego rozwiązania (bez zapętlenia ścieżki)
- `int bestObjectiveFunction` – zmienna przechowująca wartość funkcji celu najlepszego możliwego rozwiązania (włączając w to koszt powrotu do wierzchołka początkowego)
- `std::vector<int> currentSolution` – wektor przechowujący ciąg wierzchołków rozwiązania zaakceptowanego z pewnym prawdopodobieństwem (bez zapętlenia ścieżki)
- `int currentObjectiveFunction` – zmienna przechowująca wartość funkcji celu ścieżki zapisanej w powyższym wektorze (włączając w to koszt powrotu do wierzchołka początkowego)
- `std::vector<int> newSolution` – wektor, do którego w każdej iteracji kopiowane jest najlepsze lokalne rozwiązanie, a następnie podmieniane 2 losowe wierzchołki (bez zapętlenia ścieżki)
- `int newObjectiveFunction` – zmienna przechowująca wartość funkcji celu nowo znalezionej ścieżki (włączając w to koszt powrotu do wierzchołka początkowego)
- `double T_k` – przechowuje aktualną temperaturę
- `double whenFound` – zmienna przechowująca czas w którym znaleziono ostatnie najlepsze rozwiązanie

Metody klasy:

- `int findMin()` – znajduje minimalną wartość krawędzi w macierzy i ją zwraca
- `int findMax()` – znajduje maksymalną wartość krawędzi w macierzy i ją zwraca
- `void launch(Timer)` – uruchamia algorytm, do metody przekazywany jest obiekt typu `Timer` w celu kontroli, czy algorytm osiągnął kryterium stopu
- `void generateBegSolutionGreedy(std::vector<int>&, int&)` – metoda znajdująca rozwiązanie zachłanne, rozpoczynając od wierzchołka nr 0 i wybierając najkrótsze możliwe krawędzie
- `int calculateRoute()` – oblicza funkcję celu dla nowo znalezionej ścieżki (`newSolution`)

2.2. Klasa Timer

Czas mierzony jest w mikrosekundach za pomocą klasy `Timer`, za pomocą biblioteki `<windows.h>`. Po wywołaniu metody `startTimer()` zostaje odczytana zostaje liczba „tiknięć” od ostatniego restartu systemu. Następnie po wywołaniu metody `stopTimer()`, znowu odczytana zostaje liczba „tiknięć” a następnie zostaje ona odjęta od wartości odczytanej z wywołaniem metody `startTimer()` i podzielona przez wartość `frequency = 10MHz`.

3. Testy

Na czas testów tworzone są cztery, 10-elementowe tablice, do których wpisywany jest czas w którym znaleziono najlepsze rozwiązanie, wartość najlepszej funkcji celu (ścieżki), współczynnik $e^{(-1/T_k)}$ oraz temperatura końcowa T_k , dla każdego ukończonego działania algorytmu. Na koniec testów, wszystkie dane wypisywane są do pliku z rozszerzeniem `.txt`. Nazwa pliku `.txt` ma następujący format: „`typAlgorytmu_nazwaGrafu.txt`”. Oprócz tego w trakcie testów w osobnym pliku z nazwą

„numer.txt” zapisywane są czasy za każdym razem, kiedy została znaleziona lepsza wartość funkcji celu oraz wartość funkcji celu w tamtym momencie oraz w jeszcze innym pliku o nazwie „numer_s.txt” ścieżki.

Wykonano testy dla 3 kolejnych plików: *ftv55.atsp*, *ftv170.atsp* i *rbg358.atsp*, dla każdego dobranego współczynnika zmiany temperatury α oraz grafu uruchomiono algorytm 10 razy. Dla pierwszego pliku przyjęto kryterium stopu równe 2 minuty (mały graf), dla drugiego – 4 minuty (graf o średnim rozmiarze), a dla trzeciego 6 minut (graf o dużym rozmiarze). Po każdorazowym zakończeniu algorytmu zapisywano najlepszą znalezioną funkcję celu (ścieżkę) oraz czas (w sekundzie), w którym została ona znaleziona.

- Graf *ftv55.atsp*
 - $\alpha = 0.999985$ (zbyt szybki schemat schładzania)

nr	Czas [s]	Błąd względny [%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	44,9112	3,36%	1662	4,60259E-06	0,0813743
2	46,0487	4,04%	1673	2,67406E-05	0,0949728
3	47,1014	4,35%	1678	4,74194E-06	0,0815723
4	120	4,79%	1685	3,36693E-05	0,0970975
5	46,0466	3,54%	1665	4,13777E-05	0,0990808
6	47,8938	3,17%	1659	9,79E-05	0,108329
7	53,049	3,92%	1671	4,79629E-06	0,0816482
8	45,3546	3,11%	1658	8,21013E-06	0,0853961
9	120	5,60%	1698	5,56388E-06	0,08265
10	47,4921	4,66%	1683	1,32342E-05	0,0890257

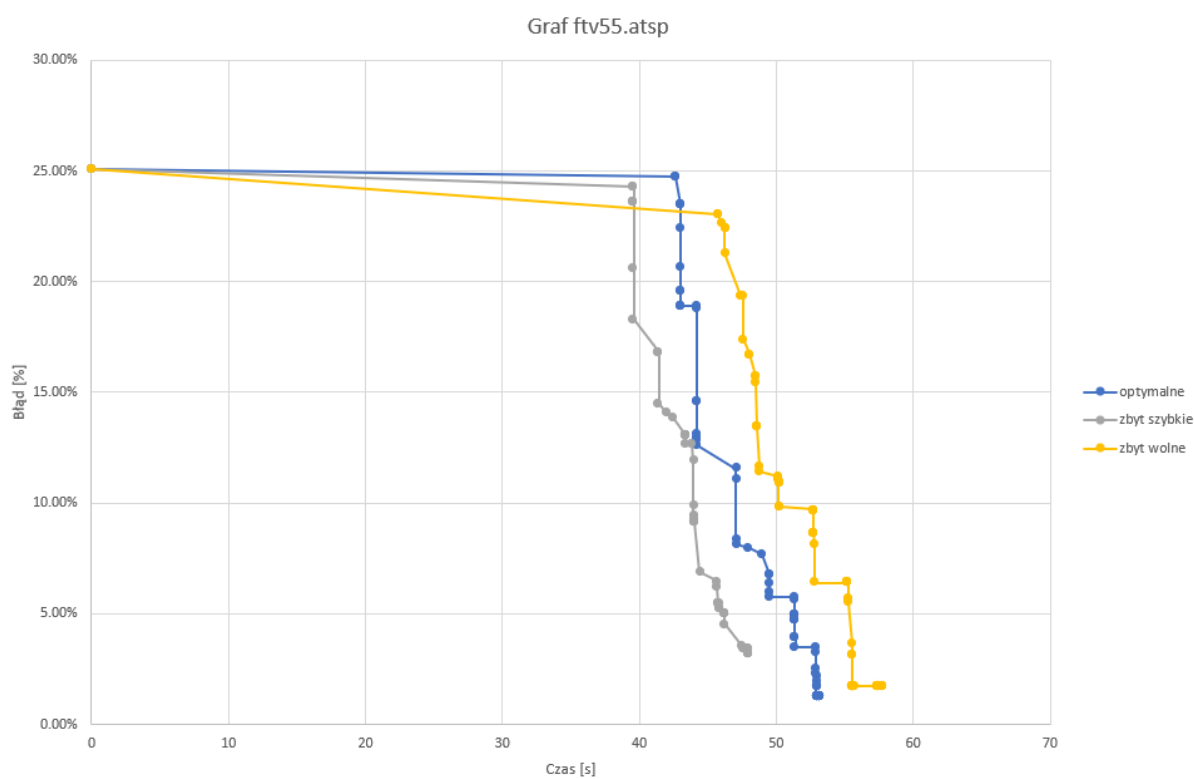
- $\alpha = 0.9999855$ (optymalny schemat schładzania)

nr	Czas [s]	Błąd względny [%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	51.4076	3.17%	1659	0.001689	0.156657
2	53.0612	3.36%	1662	0.001795	0.158156
3	53.2322	1.31%	1629	0.006357	0.197699
4	55.8986	3.30%	1661	0.001596	0.15528
5	48.5402	2.67%	1651	0.002175	0.163108
6	46.9154	4.42%	1679	0.001818	0.158473
7	53.1855	1.31%	1629	0.001568	0.158473
8	120	3.36%	1662	0.002125	0.162497
9	49.4936	2.36%	1646	0.001718	0.157077
10	50.1244	4.91%	1687	0.001677	0.156473

- $\alpha = 0.999987$ (zbyt wolny schemat schładzania)

nr	Czas [s]	Błąd względny [%]	Ścieżka	$\frac{-1}{e^{T_k}}$	T_k
1	52.8919	4.04%	1673	0.0548929	0.344546
2	52.9166	4.29%	1677	0.0548598	0.344474
3	52.6631	4.23%	1676	0.0550111	0.344801
4	51.9081	2.80%	1653	0.0765599	0.389153
5	48.1726	2.74%	1652	0.0549095	0.344582
6	120	1.74%	1636	0.0892704	0.413893
7	56.6757	3.54%	1665	0.0924946	0.420061
8	58.1367	3.36%	1662	0.0808932	0.397673
9	120	2.36%	1646	0.0537043	0.341967
10	56.2412	4.91%	1687	0.0816782	0.399207

Wykres błędu w funkcji czasu:



- Graf ftv170.atsp
 - $a = 0.9999$ (zbyt szybki schemat schładzania)

nr	Czas [s]	Błąd względny [%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	128.386	22.11%	3364	0.002288	0.164469
2	240.002	21.20%	3339	0.004146	0.182296
3	134.241	16.26%	3203	0.028576	0.28128
4	240.002	15.83%	3191	0.064759	0.365353
5	240.001	16.88%	3220	0.054422	0.343525
6	240.001	15.06%	3170	0.287218	0.801595
7	240.002	16.84%	3219	0.075911	0.387869
8	240.001	16.23%	3202	0.001401	0.152189
9	132.751	18.04%	3252	0.06131	0.35819
10	123.054	17.42%	3235	0.003204	0.174117

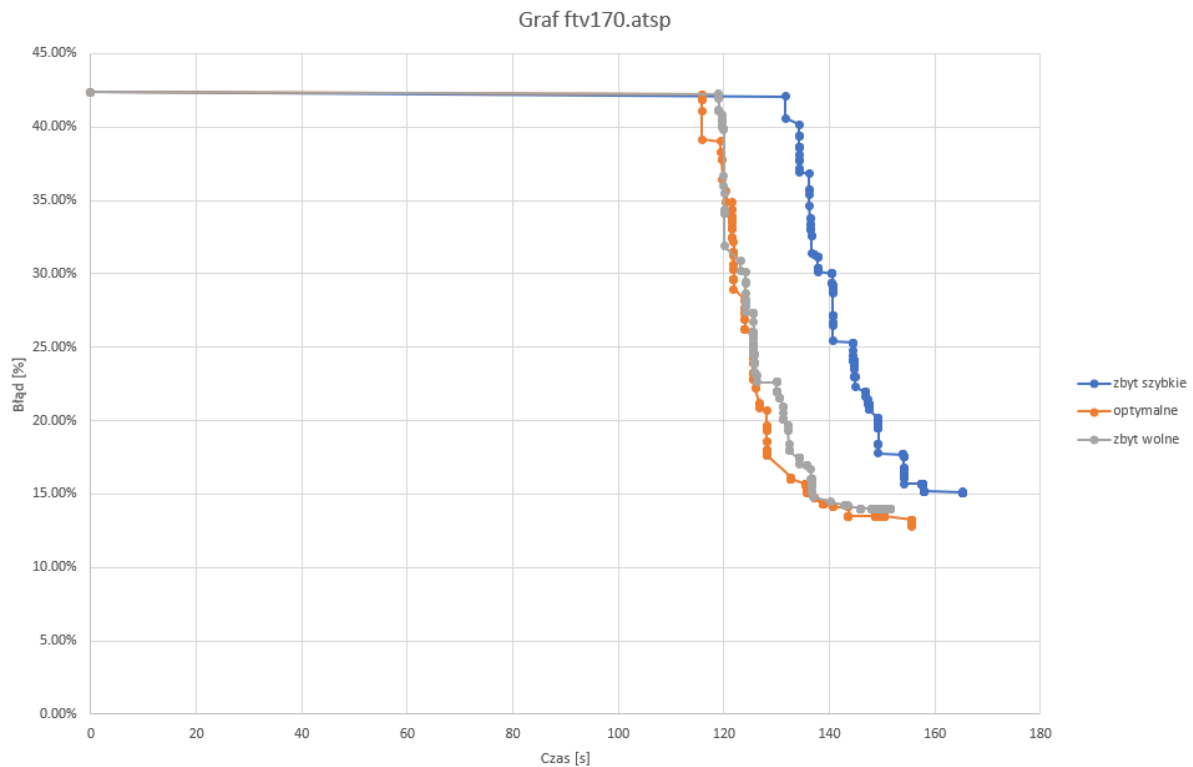
- $a = 0.999905$ (optymalny schemat schładzania)

nr	Czas [s]	Błąd względny [%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	135.697	14.05%	3142	0.0360385	0.300918
2	156.249	17.86%	3247	0.023913	0.267858
3	157.369	22.29%	3369	0.024512	0.269645
4	240.002	16.62%	3213	0.027929	0.279478
5	131.816	18.87%	3275	0.020572	0.257478
6	240	13.79%	3135	0.021964	0.261894
7	240.001	14.34%	3150	0.019942	0.255431
8	137.047	19.24%	3285	0.025166	0.271573
9	128.775	16.55%	3211	0.027636	0.278656
10	135.123	20.36%	3316	0.020847	0.25836

- $a = 0.99991$ (zbyt wolny schemat schładzania)

nr	Czas [s]	Błąd względny [%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	240.001	12.23%	3092	0.118404	0.46868
2	150.434	15.28%	3176	0.218955	0.658375
3	158.854	14.81%	3163	0.19946	0.620293
4	240.001	19.20%	3284	0.115513	0.463311
5	149.628	18.08%	3253	0.117814	0.467584
6	137.615	20.58%	3322	0.137666	0.504306
7	144.553	16.62%	3213	0.122049	0.475435
8	136.816	16.84%	3219	0.105696	0.445001
9	148.6	21.52%	3348	0.115131	0.462603
10	151.563	14.01%	3141	0.136612	0.502358

Wykres błędu w funkcji czasu:



- Graf rgb358.atstp
 - $a = 0.9995$ (zbyt szybki schemat schładzania)

nr	Czas[s]	Błąd względny[%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	222.47	2.15%	1188	1.59E-10	0.044328
2	360.001	1.55%	1181	2.38E-09	0.050357
3	232.287	1.38%	1179	2.92E-11	0.041227
4	244.708	1.89%	1185	2.26E-10	0.04502
5	360.012	1.63%	1182	3.21E-10	0.045747
6	360.007	2.58%	1193	5.26E-11	0.04225
7	225.759	2.15%	1188	1.16E-11	0.039709
8	360.022	1.98%	1186	2.97E-12	0.037678
9	225.077	1.29%	1178	1.30E-08	0.055073
10	360.023	1.46%	1180	5.37E-12	0.038535

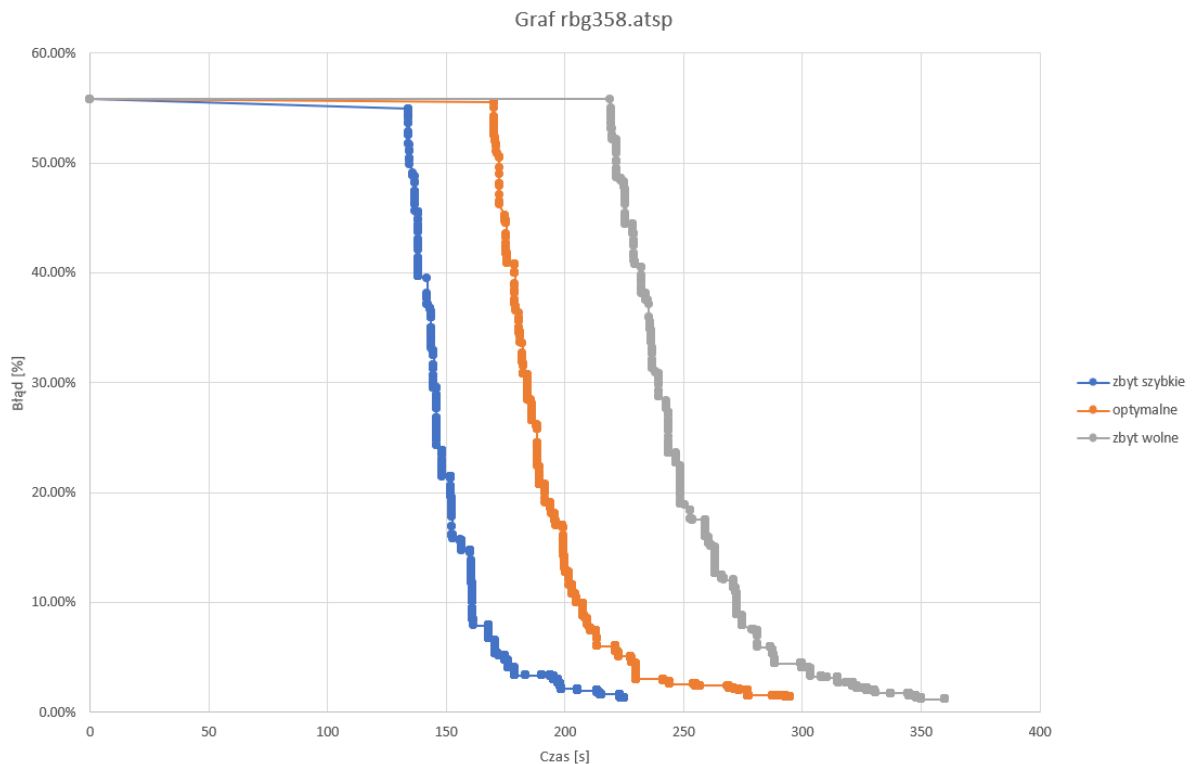
- $a = 0.9996$ (optymalny schemat schładzania)

nr	Czas[s]	Błąd względny[%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	360.025	2.41%	1191	0.001465	0.153237
2	359.926	1.98%	1186	0.014634	0.236719
3	311	2.15%	1188	0.001697	0.156772
4	360.004	2.06%	1187	0.002752	0.169628
5	360.022	2.06%	1187	0.001559	0.154716
6	360.023	1.46%	1180	0.001763	0.157716
7	300.164	1.38%	1179	0.001832	0.158665
8	360.014	2.15%	1188	0.001191	0.148529
9	360.026	1.38%	1179	0.001859	0.159046
10	294.777	1.38%	1179	0.003678	0.178398

- $a = 0.9997$ (zbyt wolny schemat schładzania)

nr	Czas[s]	Błąd względny[%]	Ścieżka	$e^{\frac{-1}{T_k}}$	T_k
1	359.125	1.20%	1177	0.190084	0.602304
2	359.257	2.24%	1189	0.188288	0.59888
3	350.853	2.75%	1195	0.199452	0.620277
4	355.381	1.46%	1180	0.184156	0.591026
5	360.015	1.20%	1177	0.212239	0.645143
6	345.418	2.32%	1190	0.189043	0.600319
7	353.889	1.38%	1179	0.195607	0.612877
8	351.779	1.72%	1183	0.184343	0.59138
9	359.421	1.46%	1180	0.1898	0.601762
10	350.384	1.63%	1182	0.18876	0.599779

Wykres błędu w funkcji czasu:



4. Najlepsze znalezione ścieżki

- ftv55.atsp

37 11 19 20 40 18 39 38 10 51 14 12 15 16 17 25 24 42 22 41 21 50 23 54 27 49 29 26 52 0 33
2 13 35 4 6 5 48 31 47 28 53 43 44 45 30 46 55 34 1 3 7 32 8 36 9 37

- ftv170.atsp

144 143 148 147 137 136 128 130 135 138 139 140 141 7 14 15 159 16 17 18 19 20 21 32
158 36 157 33 31 34 35 37 168 72 78 82 79 80 81 0 1 77 170 38 39 40 156 155 41 42 45 44 46
47 48 51 52 53 43 55 54 58 59 60 63 64 56 57 62 61 68 167 70 69 67 66 65 87 85 86 93 166
108 83 84 71 50 49 73 2 3 10 76 74 75 11 12 13 29 30 28 27 26 23 22 24 25 150 149 161 160
151 152 142 6 8 9 4 5 169 111 112 132 133 134 131 113 115 116 117 118 119 120 121 122
162 123 101 100 102 103 109 107 106 105 165 163 99 98 95 94 92 88 153 154 89 90 91 96
97 104 110 114 164 127 126 125 124 129 146 145 144

- rbg358.atsp

294 225 11 284 188 158 149 279 338 53 132 328 314 256 160 84 32 31 174 56 50 238 191
250 215 308 28 61 80 0 239 21 252 335 105 286 208 115 170 264 341 62 52 114 102 137 88
58 243 41 73 280 159 121 120 144 123 231 104 97 22 357 251 130 111 331 219 196 92 275
30 298 198 127 98 347 83 10 282 281 150 195 190 297 99 233 19 211 241 311 116 36 29 112
278 334 155 78 346 95 136 85 147 175 172 109 325 350 207 240 192 182 300 229 309 24
234 199 209 283 118 146 153 65 273 122 303 200 228 327 71 277 302 274 35 42 26 343 287
4 44 262 332 210 176 254 318 40 167 7 246 15 6 81 242 258 49 5 148 232 86 47 271 310 17
194 131 189 342 13 3 319 261 330 257 76 1 157 248 45 25 38 90 69 126 293 217 201 66 14
87 288 77 18 296 216 59 125 349 141 93 2 57 140 110 197 162 133 164 218 321 356 299 221

267 75 212 333 166 138 113 269 266 43 12 119 244 322 108 259 329 223 129 247 305 272
255 184 183 320 270 260 55 178 265 180 165 312 101 227 315 64 89 276 295 193 220 46
206 204 103 177 340 301 324 181 20 249 345 289 72 307 306 351 304 37 326 168 237 355
317 263 27 96 117 135 39 128 202 285 70 33 337 226 134 82 79 68 353 313 51 67 245 142
352 253 203 74 290 236 48 354 213 222 316 224 169 348 292 339 230 8 23 185 54 100 139
106 323 336 268 154 124 63 171 16 34 9 179 173 187 205 91 161 152 214 186 107 94 156 60
344 291 163 145 151 143 235 294

5. Wnioski

Ogólnie rzecz biorąc, im dłuższa epoka, tym dłużej przeszukujemy w określonej temperaturze i rzadziej schładzamy, co prowadzi z kolei do większego prawdopodobieństwa akceptacji gorszego rozwiązania. Im krótsza natomiast, częściej chłodzimy i istnieje mniejsze prawdopodobieństwo akceptacji gorszego rozwiązania. Jeśli chodzi o temperaturę początkową, im wyższa, tym częściej będą akceptowane gorsze rozwiązania, chyba że długość epoki zostanie skrócona, co będzie oznaczało częstsze chłodzenie.

Z wykresów natomiast wynika, że im większy współczynnik zmiany temperatury α , tym rozwiązanie zostaje znajdowane później w czasie.

6. Źródła

- 1) http://algorytmy.ency.pl/tutorial/symulowane_wyzarzanie
- 2) https://en.wikipedia.org/wiki/Simulated_annealing
- 3) https://www.youtube.com/watch?v=gX-X85dCib0&t=634s&ab_channel=MaciejKomosi%C5%84ski%5Bpl%5D
- 4) https://www.researchgate.net/profile/Stanislaw-Kowalik/publication/305489442_SYMULOWANE_WYZARZANIE_W_ZASTOSOWANIU_DO_WYZNACZANIA_EKSTREMUM_GLOBALNEGO_FUNKCJI_O_WIELU_EKSTREMACH_LOKALNYCH_DALEKO_ODDALONYCH_OD_SIEBIE_LUB_BARDZO_ZAGESZCZONYCH/links/579103b108ae0831552f9468/SYMULOWANE-WYZARZANIE-W-ZASTOSOWANIU-DO-WYZNACZANIA-EKSTREMUM-GLOBALNEGO-FUNKCJI-O-WIELU-EKSTREMACH-LOKALNYCH-DALEKO-ODDALONYCH-OD-SIEBIE-LUB-BARDZO-ZAGESZCZONYCH.pdf
- 5) https://miscj.aut.ac.ir/article_174_b3feae27fb8a7a4a4b5d1e33169b5a6c.pdf