

Projektowanie Efektywnych Algorytmów Etap 2

Temat:

Rozwiązanie TSP za pomocą metod tabu search i symulowanego wyżarzania

Prowadzący
dr inż. Tomasz Kapłon

Autor
Paweł Parczyk

Wstęp

Początki problemu Komiwożażera (TSP - Travelling salesman problem) sięgają XIX wieku, kiedy to William Rowan Hamilton opisał problem istnienia cyklu n -elementowego w grafie o n wierzchołkach. Za pierwszego autora, który sformalizował matematycznie problem komiwożażera, uznaje się Karla Mengera, który w 1930 zwrócił szczególną uwagę na trudność w obliczeniu optymalnego rozwiązania. Z uwagi na prostotę opisu problemu i trudność optymalizacji algorytmów służących do jego rozwiązania, problem stał się bardzo popularny.

Celem projektu było stworzenie dwóch algorytmów rozwiązujących problem TSP dla instancji, których metody dokładne, zaimplementowane na poprzednim etapie projektu nie były w stanie odnaleźć. Wybranymi metodami były wyszukiwanie z zakazami (tabu search) i symulowane wyżarzanie.

Program został napisany w języku c++ w wersji 11 z wykorzystaniem technik programowania obiektowego. Kod został skompilowany przez kompilator g++, a następnie uruchomiony na komputerze zarządzanym przez system Linux Ubuntu 16.04, sama maszyna wyposażona była w 8GB pamięci ram i procesor intel core i5.

Algorytmy zostały napisane w wersjach iteracyjnych w celu zminimalizowania czasu wykonania.

Cechy metod niedokładnych

Podczas realizacji pierwszego etapu zostały zaimplementowane metody, których zadaniem było znalezienie najlepszego rozwiązania (dla rozpatrywanego problemu najlepszy to taki dla którego koszt przejścia jest najniższy). Z uwagi cechy problemu TSP (liczba potencjalnych rozwiązań wynosi $n!$) takie algorytmy mają ograniczenia.

Problem z uzyskaniem wyniku optymalnego pojawia się gdy czas przetwarzania staje się dłuższy niż maksymalny dopuszczony czas, lub maszyna przetwarzająca problem ma mniej pamięci niż algorytm potrzebuje.

Problem skończonych zasobów czasowych i sprzętowych rozwiązują algorytmy niedokładne. Ich zadaniem jest przeszukiwanie przestrzeni stanów przez określony czas i znalezienie jak najlepszego rezultatu w tym czasie. Daje to możliwość przetwarzania znacznie większych instancji problemów, kosztem dokładności rozwiązań. Pomimo zmiany strategii znalezienia optymalnego rozwiązania w dalszym ciągu jest możliwe.

Tabu search

Metoda przeszukiwania z zakazami polega na dokonaniu przeglądu części wszystkich możliwych permutacji problemu i wybraniu takiego rozwiązania, które wśród przeglądanych jest najlepsze.

Zastosowany algorytm:

Algorytm rozpoczyna od wylosowania wierzchołków *perm* wierzchołków grafu. Następnie powtarza następujące kroki aż do osiągnięcia **warunku zakończenia**: Dla każdej pary indeksów w permutacji *perm* i, j jeżeli $i \neq j$ i ruch i, j nie jest zapisany na liście tabu **wykonaj ruch i, j** . Jeżeli nowa droga jest korzystniejsza od aktualnie najlepszej to zapisz najlepszą. Po przeanalizowaniu wszystkich dozwolonych zamian (permutacji *perm*) dopisz zastosowany ruch i, j do listy tabu. A także zamień *perm* na najlepszą znaną ścieżkę o ile jest lepsza od *perm*. Następnie **aktualizuj listę tabu**. Jeżeli nie doszło do zmiany *perm* to zwiększ licznik błędów krytycznych. Jeśli liczba błędów krytycznych przekroczy wartość progową wylosuj nową permutację *perm* i opróżnij listę tabu. Następnie powrót do warunku zakończenia. Jeśli warunek zakończenia jest spełniony zwróć najlepszą znaną ścieżkę.

Poniżej zaznaczono elementy zmienne dla różnych implementacji algorytmu. Z uwagi na nieokreśloną jakość rozwiązań istnieje dowolność implementacji tych modułów algorytmu

Warunek zakończenia:

W zastosowanym algorytmie warunkiem zakończenia było wykonanie zadanej ilości iteracji pętli sprawdzającej najlepsze możliwe rozwiązanie powstałe z ruchu i, j na permutacji *perm*. Takie rozwiązanie pozwala w sposób jednoznaczny sterować czasem wykonywania się algorytmu.

Wykonaj ruch i, j :

Wykonanie ruchu polegającego na zamianie wierzchołka *i*-tego z *j*-tym w permutacji *perm*

Aktualizuj listę tabu:

Mechanizm aktualizacji cechuje kilka parametrów. Czas przebywania ruchu na liście tabu (kadencja) długość listy tabu (ilość ruchów które mogą być zapamiętane). W zastosowanym algorytmie kadencja jest równa długości listy tabu.

Optymalizacja parametrów Tabu search

Stworzony algorytm wymagał zdefiniowania dwóch parametrów **liczby iteracji** i **długości listy tabu**. W celu znalezienia parametrów optymalnych zostały przeprowadzone testy.

Optymalizacja liczby iteracji

Pierwszym parametrem poddanym optymalizacji była liczba iteracji. Przeprowadzono 20 testów dla ustalonych parametrów a następnie uśredniono wyniki.

| Przetwarzanie grafu tabu search | | | | | |
|---------------------------------|-----------|-------|----------|----------|--|
| kadencja | | | | 15 | |
| liczba wierzchołków | | | | 17 | |
| liczba iteracji | | | | 50000 | |
| Najniższy koszt | | | | 39 | |
| l.p. | czas [ms] | droga | czas [s] | Błąd [%] | |
| 1 | 592,09 | 42 | 0,59 | 7,14 | |
| 2 | 571,05 | 49 | 0,57 | 20,41 | |
| 3 | 571,99 | 46 | 0,57 | 15,22 | |
| 4 | 578,30 | 51 | 0,58 | 23,53 | |
| 5 | 571,55 | 46 | 0,57 | 15,22 | |
| 6 | 573,42 | 42 | 0,57 | 7,14 | |
| 7 | 572,36 | 42 | 0,57 | 7,14 | |
| 8 | 570,98 | 44 | 0,57 | 11,36 | |
| 9 | 573,01 | 41 | 0,57 | 4,88 | |
| 10 | 585,94 | 46 | 0,59 | 15,22 | |
| 11 | 623,25 | 45 | 0,62 | 13,33 | |
| 12 | 600,12 | 49 | 0,60 | 20,41 | |
| 13 | 581,68 | 54 | 0,58 | 27,78 | |
| 14 | 577,98 | 50 | 0,58 | 22,00 | |
| 15 | 574,34 | 40 | 0,57 | 2,50 | |
| 16 | 575,68 | 42 | 0,58 | 7,14 | |
| 17 | 611,44 | 50 | 0,61 | 22,00 | |
| 18 | 578,98 | 42 | 0,58 | 7,14 | |
| 19 | 575,20 | 42 | 0,58 | 7,14 | |
| 20 | 573,87 | 44 | 0,57 | 11,36 | |
| Razem | 581,66 | 45,35 | 0,58 | 13,40 | |

Tab 1: Przykładowa tabela wyników przeszukiwania grafu 17 wierzchołkowego przez zadaną liczbę iteracji

Dokładnie w ten sam sposób przetestowano następujące ilości iteracji: 50tyś, 100tyś, 200tyś, 500tyś i 1 milion.

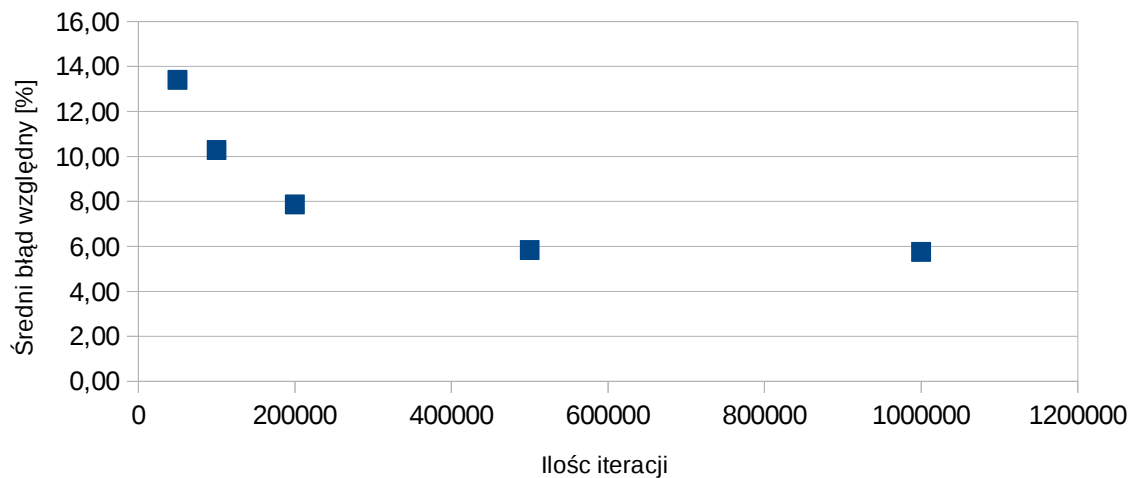
Poniżej została zaprezentowana tabela podsumowująca otrzymane uśrednione wyniki

| Liczba iteracji | 50000 | 100000 | 200000 | 500000 | 1000000 |
|-----------------|-------|--------|--------|--------|---------|
| Średni czas[s] | 0,58 | 1,16 | 2,30 | 5,78 | 11,74 |
| Średni błąd [%] | 13,40 | 10,28 | 7,87 | 5,84 | 5,75 |

Tab 2: Tabela przedstawiająca czasy i średnie błędy uzyskane w testach

Zależność jakości wyników tabu search od liczby iteracji

(czas jest zależny od liczby iteracji)

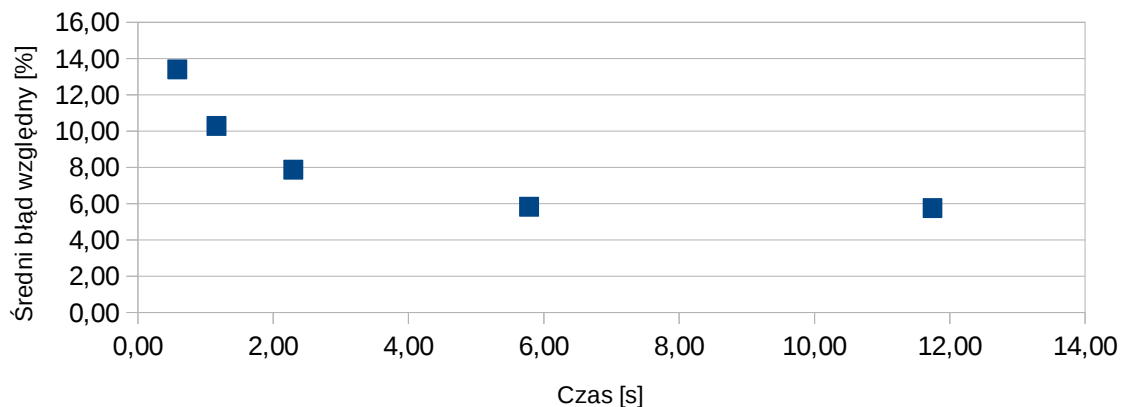


Wyk 1: Wykres zależności jakości od liczby iteracji

Na tym etapie warto zauważyć że czas przetwarzania jest liniowo zależny od liczby iteracji do przetworzenia.

Zależność jakości wyników tabu search od czasu przetwarzania

(czas jest zależny od liczby iteracji)



Wyk 2: Wykres zależności jakości wyników od czasu przetwarzania grafu 17 wierzchołkowego

Na podstawie wykresów Wyk 1 i Wyk 2 można wyciągnąć wniosek, że im dłuższy czas przetwarzania tym wyniki wyszukiwania są bliższe rozwiązaniu optymalnemu. Jednak relacja ta nie jest liniowa. Podwojenie czasu z 6 sekund do 12 przyniosło niewielkie zysk dokładności względem zmiany czasu z 1 sekundy do dwóch. W związku z obserwacjami do dalszych testów została wybrana liczba 500tyś iteracji.

Optymalizacja długości listy tabu

Następnym parametrem którego wartość optymalna należało określić była długość listy tabu. Początkowo zostało postawione pytanie „Jakiej długości powinna być lista tabu w stosunku do liczby możliwych ruchów w permutacji?” zgodnie z lematem o uściskach dłoni liczba możliwych ruchów wyraża się wzorem $\frac{n \cdot (n-1)}{2}$ co dla wybranego do testów grafu daje 136 możliwych ruchów.

Zostały przeprowadzone testy dla długości listy od 5 do 120 ruchów. Każdy test polegał na obliczeniu średniej z 20 prób. Poniżej zaprezentowano tabele z pomiarami dla jednej z długości.

| Przetwarzanie grafu tabu search | | | | | |
|--|-----------|-------|----------|----------|--|
| kadencja | | | | 20 | |
| liczba wierzchołków | | | | 17 | |
| liczba iteracji | | | | 500000 | |
| Długość tabu | | | | 20 | |
| Stosunek dł. tabu do liczby możliwych ruchów [%] | | | | 14,71 | |
| Najniższy koszt | | | | 39 | |
| l.p. | czas [ms] | droga | czas [s] | Błąd [%] | |
| 1 | 5769,63 | 41 | 5,77 | 4,88 | |
| 2 | 5848,27 | 44 | 5,85 | 11,36 | |
| 3 | 5759,17 | 40 | 5,76 | 2,50 | |
| 4 | 5871,17 | 42 | 5,87 | 7,14 | |
| 5 | 5913,59 | 42 | 5,91 | 7,14 | |
| 6 | 5833,78 | 44 | 5,83 | 11,36 | |
| 7 | 5742,79 | 42 | 5,74 | 7,14 | |
| 8 | 5787,01 | 40 | 5,79 | 2,50 | |
| 9 | 5758,27 | 42 | 5,76 | 7,14 | |
| 10 | 5720,46 | 39 | 5,72 | 0,00 | |
| 11 | 5721,03 | 42 | 5,72 | 7,14 | |
| 12 | 5730,45 | 43 | 5,73 | 9,30 | |
| 13 | 5730,79 | 39 | 5,73 | 0,00 | |
| 14 | 5725,54 | 39 | 5,73 | 0,00 | |
| 15 | 5878,99 | 44 | 5,88 | 11,36 | |
| 16 | 5962,03 | 44 | 5,96 | 11,36 | |
| 17 | 5773,34 | 40 | 5,77 | 2,50 | |
| 18 | 5715,21 | 40 | 5,72 | 2,50 | |
| 19 | 5714,93 | 39 | 5,71 | 0,00 | |
| 20 | 5718,51 | 44 | 5,72 | 11,36 | |
| Razem | 5783,75 | 41,5 | 5,78 | 5,84 | |

Tab 3: Przykładowa tabela wyników przeszukiwania grafu 17 wierzchołkowego z zadana długością listy tabu

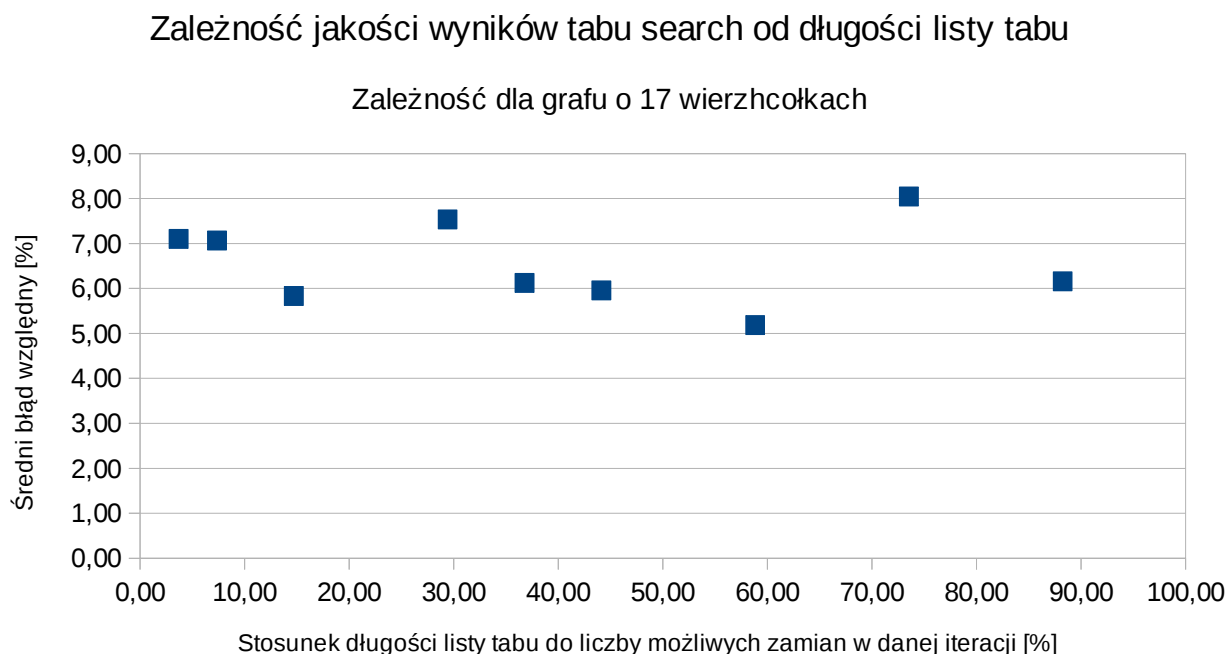
Powyższa tabela prezentuje uzyskane rezultaty dla długości listy tabu równej 20 ruchów co daje 14,71% dostępnych ruchów.

Takie same tabele zostały wygenerowane dla list o długościach 5, 10, 20, 40, 50, 60, 80, 100, 120. Otrzymane wyniki zaprezentowano w tabeli poniżej.

| Długość listy tabu | 5 | 10 | 20 | 40 | 50 |
|------------------------|------|------|-------|-------|-------|
| Długość listy tabu [%] | 3,68 | 7,35 | 14,71 | 29,41 | 36,76 |
| Średni błąd [%] | 7,10 | 7,07 | 5,84 | 7,54 | 6,12 |

| Długość listy tabu | 60 | 80 | 100 | 120 |
|------------------------|-------|-------|-------|-------|
| Długość listy tabu [%] | 44,12 | 58,82 | 73,53 | 88,24 |
| Średni błąd [%] | 5,96 | 5,19 | 8,04 | 6,16 |

Tab 4: Tabela przedstawiająca czasy i średnie błędy dla testowanych wartości długości listy tabu



Wyk 4: Wykres zależności jakości wyników od długości listy tabu dla grafu 17 wierzchołkowego

Uzyskane wyniki niezgodnie z oczekiwaniami nie ułożyły się w żadną określoną funkcję, więc test został przeprowadzony również na grafie większym przy zachowaniu tych samych założeń.

Do testów wybrano graf o wielkości 29 wierzchołków. Wyniki zostały zaprezentowane poniżej.

| Przetwarzanie grafu tabu search | | | | | |
|--|-----------------|----------------|--------------|--------------|--------|
| kadencja | | | | | 20 |
| liczba wierzchołków | | | | | 29 |
| liczba iteracji | | | | | 500000 |
| Długość tabu | | | | | 60 |
| Stosunek dł. tabu do liczby możliwych ruchów [%] | | | | | 14,71 |
| Najniższy koszt | | | | | 2020 |
| l,p, | czas [ms] | droga | czas [s] | Błąd [%] | |
| 1 | 28096,46 | 2527 | 28,10 | 20,06 | |
| 2 | 27684,87 | 2286 | 27,68 | 11,64 | |
| 3 | 27921,47 | 2480 | 27,92 | 18,55 | |
| 4 | 27723,65 | 2433 | 27,72 | 16,97 | |
| 5 | 27959,59 | 2509 | 27,96 | 19,49 | |
| 6 | 27771,42 | 2846 | 27,77 | 29,02 | |
| 7 | 27724,93 | 3117 | 27,72 | 35,19 | |
| 8 | 27973,51 | 2419 | 27,97 | 16,49 | |
| 9 | 28118,61 | 2465 | 28,12 | 18,05 | |
| 10 | 27874,01 | 2341 | 27,87 | 13,71 | |
| 11 | 27683,60 | 2649 | 27,68 | 23,74 | |
| 12 | 28122,85 | 2572 | 28,12 | 21,46 | |
| 13 | 27840,72 | 2450 | 27,84 | 17,55 | |
| 14 | 28021,89 | 3001 | 28,02 | 32,69 | |
| 15 | 27868,28 | 2871 | 27,87 | 29,64 | |
| 16 | 27846,99 | 2532 | 27,85 | 20,22 | |
| 17 | 28021,08 | 2528 | 28,02 | 20,09 | |
| 18 | 27791,78 | 2472 | 27,79 | 18,28 | |
| 19 | 28141,71 | 2500 | 28,14 | 19,20 | |
| 20 | 27677,98 | 2939 | 27,68 | 31,27 | |
| Razem | 27893,27 | 2596,85 | 27,89 | 21,67 | |

Tab 5: Przykładowa tabela wyników przeszukiwania grafu 29 wierzchołkowego z zadana długością listy tabu

Testy przeprowadzono dla ilości wierzchołków stanowiących taki sam udział w ilości ruchów jak w przypadku testów na grafie 17 wierzchołkowym.

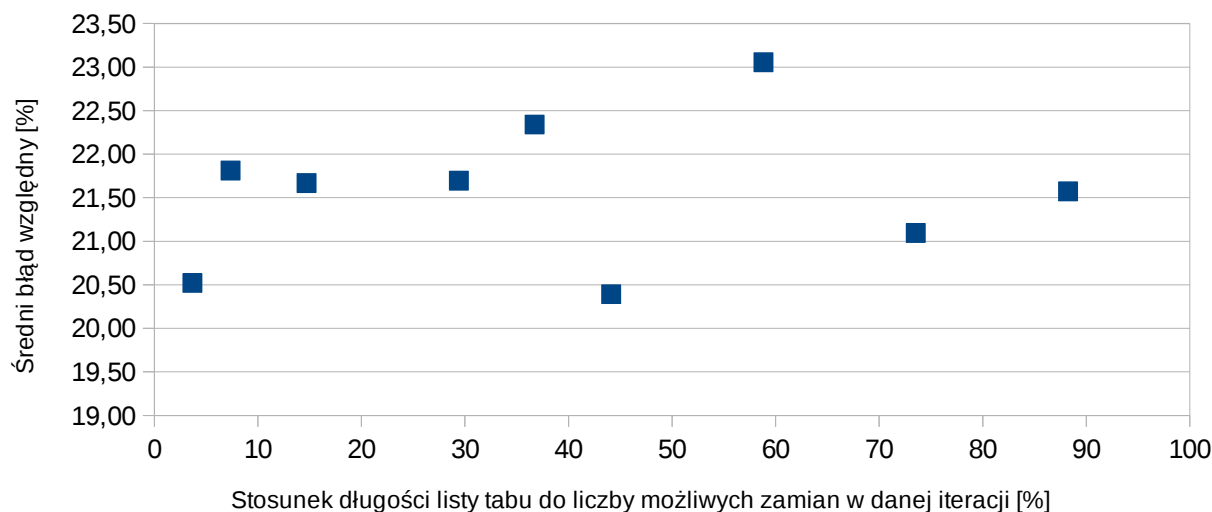
| Długość listy tabu | 15 | 30 | 60 | 120 | 150 |
|------------------------|-------|-------|-------|-------|-------|
| Długość listy tabu [%] | 3,68 | 7,35 | 14,71 | 29,41 | 36,72 |
| Średni błąd [%] | 20,52 | 21,81 | 21,67 | 21,69 | 22,34 |

| Długość listy tabu | 180 | 239 | 299 | 359 |
|------------------------|-------|-------|-------|-------|
| Długość listy tabu [%] | 44,12 | 58,82 | 73,53 | 88,24 |
| Średni błąd [%] | 20,39 | 23,06 | 21,10 | 21,57 |

Tab 6: Tabela przedstawiająca czasy i średnie błędy dla testowanych wartości długości listy tabu

Zależność jakości wyników tabu search od długości listy tabu

Zależność dla grafu o 29 wierzchołkach



Wyk 5: Wykres zależności jakości wyników od długości listy tabu dla grafu 29 wierzchołkowego

Na podstawie uzyskanych wyników można wyciągnąć wnioski, iż dla zaimplementowanego algorytmu funkcja jakości wyników od długości listy tabu nie wyraża się żadnym wzorem. Na wykresach występuje kilka minimów lokalnych. Wszystkie w podobnych lokalizacjach.

Wnioski:

Dla zaimplementowanego algorytmu długość listy tabu nie ma znaczenia. Dlatego do dalszych prób można przyjąć stałą wartość tablicy tabu na około 30 ruchów (uzależnianie długości listy tabu od instancji problemu nie przyniosło pożądanych rezultatów). Natomiast parametr liczby iteracji w sposób jednoznaczny jest odpowiedzialny za poprawę jakości wyników.

Symulowane wyżarzanie

Metoda symulowanego wyżarzania polega na losowaniu, w każdej iteracji nowego rozwiązania i jeśli jest ono lepsze od aktualnie najlepszego przemnożone przez pewien współczynnik zależny od zmieniającej się temperatury podmianie. Początkowo gdy temperatura jest wysoko algorytm „chętnie” przyjmuje rozwiązania gorsze jednak wraz z stygnięciem tendencja ta zanika i algorytm stabilizuje wynik. Nazwa algorytmu nawiązuje do metody studzenia metali w metalurgii, które nadaje stopom wytrzymałość i elastyczność.

Zastosowany algorytm:

Na początku zostaje wylosowana permutacja $perm$ i ustawiona temperatura t równa **maksymalnej temperaturze** następnie w każdej kolejnej iteracji dopóki ilość wykonanych iteracji jest mniejsza niż **liczba zadanych iteracji** losowane są dwa indeksy i, j następnie na permutacji wykonywany jest **ruch** (zamiana wartości pod indeksami i i j). Później zostaje policzona różnica

kosztu nowej drogi i kosztu minimalnego δ . Jeżeli δ jest mniejsza niż 0 to nowa droga jest lepsza od *perm* więc *perm* zostaje zamienione na nową drogę. Jeżeli nie to zostaje wylosowana liczba $x \in (0,1)$ i policzona wartość $e = \exp(\frac{-\delta}{t})$. Jeżeli $e < x$ to za *perm* zostanie podstawiona nowa droga. Następnie aktualna temperatura t zostaje zmieniona zgodnie ze **współczynnikiem chłodzenia**. Dodatkowo został wprowadzony licznik iteracji podczas których nie doszło do żadnych zmian. Ma on na celu zakończenie przetwarzania w sytuacji gdy już nic się nie zmienia.

Poniżej zaznaczono parametry pracy algorytmu:

Maksymalna temperatura

Temperatura z jaką algorytm rozpoczyna pracę.

Liczba zadanych iteracji

Liczba która definiuje jak długo algorytm ma się wykonywać.

Współczynnikiem chłodzenia

Wartość zgodnie z którą aktualna temperatura t będzie się zmieniać w kolejnych iteracjach algorytmu.

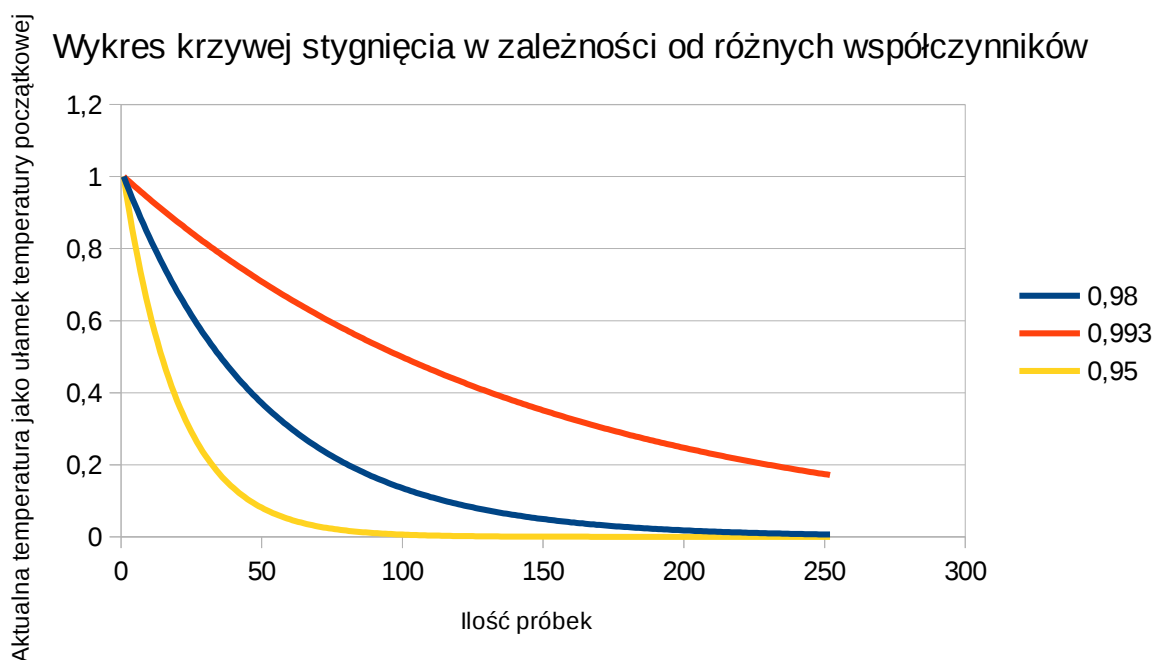
Podobnie do wyszukiwania z zakazami w symulowanym wyżarzaniu również można wprowadzać zmiany w celu optymalizacji algorytmu. W zależności od znajomości problemu zmianą mogą podlegać: funkcja zmiany temperatury, sposób generowania indeksów i i j , i funkcję ruchu.

Optymalizacja parametrów symulowanego wyżarzania

Analiza wpływu parametrów na zachowanie się algorytmu

Współczynnik chłodzenia

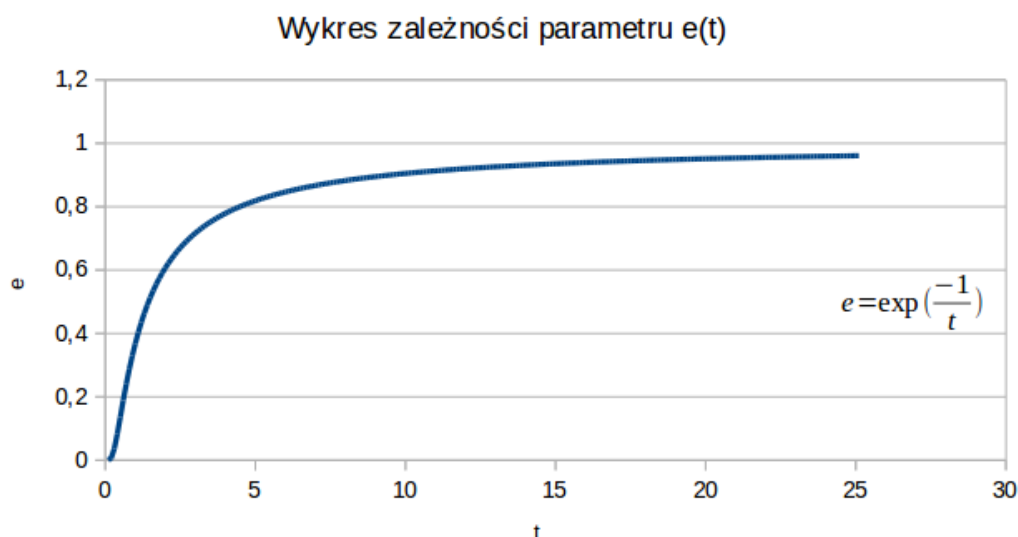
Jest to parametr odpowiedzialny za określenie jak szybko zacznie się stabilizować wynik i przestaną być rozpatrywane rozwiązania z gorszym kosztem. Poniżej zostały zaprezentowane wykresy prezentujące szybkość straty temperatur w zależności od zadanego współczynnika. Manipulując tym parametrem można zwiększać częstotliwość zmian permutacji *perm*, ponieważ im wyższa temperatura przetwarzania tym więcej gorszych wyników zostanie zakwalifikowanych przez warunek $e < x$.



Wyk 6: Wykres zależności temperatury od ilości próbek przy zadanym współczynniku

Bieżąca temperatura

Zgodnie ze wzorem $e = \exp\left(\frac{-\delta}{t}\right)$ bieżąca temperatura decyduje o wartości zmiennej e wykorzystywanej w warunku przepuszczającym do dalszego przetwarzania permutacje o koszcie większym niż aktualnie przetwarzana. Praca na większej temperaturze zwiększa prawdopodobieństwo, a na niższej zmniejsza. Dzięki temu algorytm nie blokuje się w minimach lokalnych.



Wyk 7: Wykres zależności $e(t)$

Zgodnie z zaprezentowanym wykresem im większa temperatura tym prawdopodobieństwo zakwalifikowania rozwiązania gorszego większe.

Liczba iteracji:

liczba iteracji wpływa na czas działania algorytmu ale jednocześnie daje możliwość zrobienia przeglądu większej liczby permutacji a w konsekwencji zwiększa szanse na wylosowanie rozwiązania optymalnego.

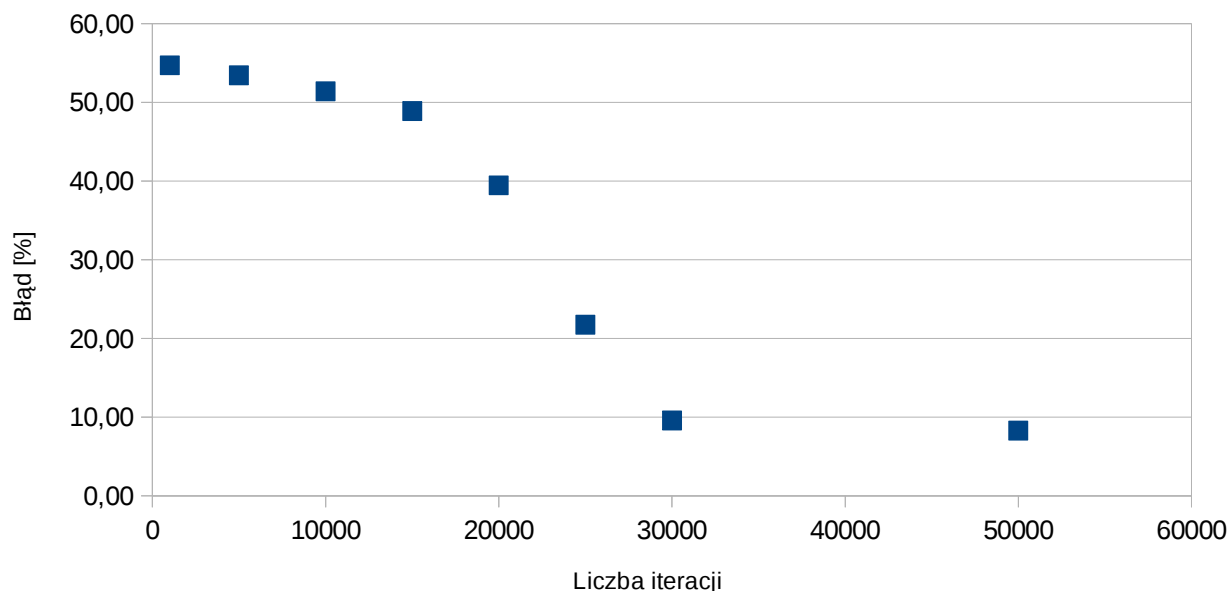
Optymalizacja parametrów algorytmu

Początkowo parametr liczby iteracji wydał się być niezależny od reszty parametrów więc, została przeprowadzona symulacja właśnie dla niego. Poniżej zaprezentowano wyniki z prób. Wyniki zostały od razu uśrednione.

| Tabela zależności błędu od liczby iteracji | | | |
|--|-----------|---------|----------|
| Temperatura | | 5000 | |
| Współczynnik chłodzenia | | 0,9998 | |
| Liczba iteracji | Czas [ms] | Droga | Błąd [%] |
| 1000 | 1,01 | 4521,70 | 54,70 |
| 5000 | 1,47 | 4387,95 | 53,45 |
| 10000 | 1,94 | 4200,10 | 51,40 |
| 15000 | 2,91 | 3999,60 | 48,90 |
| 20000 | 3,80 | 3364,70 | 39,45 |
| 25000 | 4,71 | 2607,35 | 21,75 |
| 30000 | 5,58 | 2252,15 | 9,60 |
| 50000 | 9,19 | 2218,60 | 8,30 |

Tab 7: Tabela zależności błędu od liczby iteracji

Wykres jakości przeszukiwania od liczby iteracji



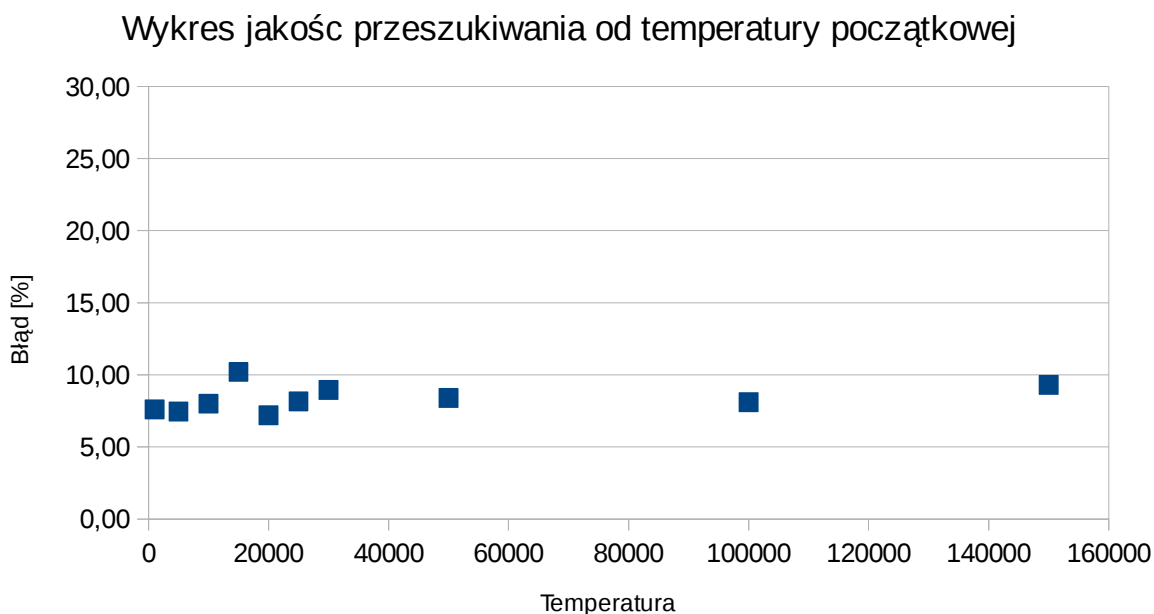
Wyk 8: Wykres zależności błędu od liczby iteracji

Wyniki potwierdzają wcześniejszą hipotezę, iż im większa liczba iteracji (dłuższy czas przetwarzania) tym wyniki przetwarzania będą lepsze.

Następnie przy ustalonych współczynniku i liczbie iteracji wyznaczono zależność błędu od temperatury przetwarzania. Wyniki przedstawiono poniżej.

| Tabela zależności błędu od temperatury | | | | |
|--|-----------|---------|----------|--|
| Liczba iteracji | | 100000 | | |
| Współczynnik chłodzenia | | 0,9998 | | |
| Temperatura | Czas [ms] | Droga | Błąd [%] | |
| 1000 | 17,79 | 2203,80 | 7,60 | |
| 5000 | 17,65 | 2196,15 | 7,45 | |
| 10000 | 17,50 | 2212,15 | 8,00 | |
| 15000 | 17,51 | 2265,50 | 10,20 | |
| 20000 | 18,13 | 2191,80 | 7,20 | |
| 25000 | 18,14 | 2216,65 | 8,15 | |
| 30000 | 17,83 | 2235,95 | 8,95 | |
| 50000 | 17,54 | 2221,40 | 8,40 | |
| 100000 | 17,60 | 2213,80 | 8,10 | |
| 150000 | 17,80 | 2244,65 | 9,30 | |

Tab 8: Tabela zależności błędu od temperatury

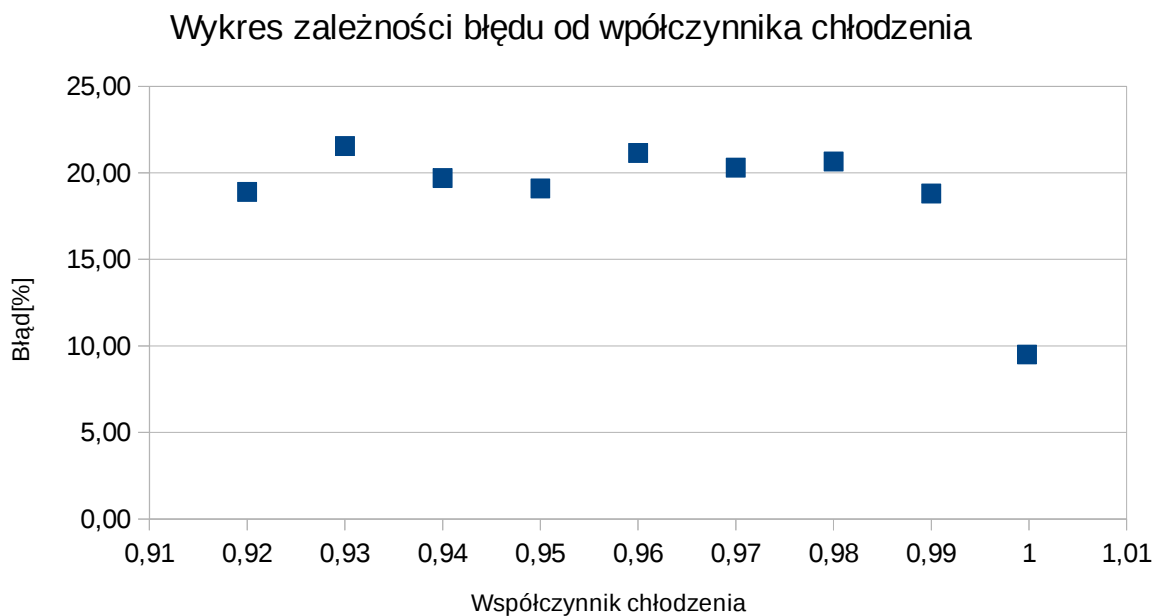


Wyk 9: Wykres jakości rozwiązań od temperatury początkowej

Na podstawie otrzymanych rezultatów nie można stwierdzić, że błąd jest funkcją bezpośrednią temperatury. Na wykresie widać, że niezależnie od temperatury błąd obliczeń cały czas mieści się w granicach 7 – 11 %.

Zadaniem temperatury jest zagwarantowanie, że algorytm nie zablokuje się w minimum lokalnym podczas przetwarzania i przez to zwiększa szansę na znalezienie rezultatów bliskich wartości optymalnej. Na podstawie otrzymanego przedziału błędów można stwierdzić że algorytm podczas pracy z dowolną z podanych temperatur skutecznie opuszczał minima lokalne w poszukiwaniu minimum globalnego.

Działanie współczynnika w przeciwieństwie do dwóch pozostałych parametrów nie jest takie bezpośrednie. Współczynnik oddziałuje na temperaturę obniżając ją. Im szybciej będzie ją obniżał tym wcześniej algorytm zacznie się stabilizować przez co rośnie szansa na zakończenie pracy w lokalnym minimum.



Wyk 10: Wykres zależności błędu od współczynnika chłodzenia

Zgodnie z oczekiwaniami na wykresie da się zaobserwować, do którego momentu algorytm nie potrafi sobie poradzić z lokalnymi minimami. Dla współczynników > 0.99 widać znaczny spadek błędów.

Wnioski:

Na podstawie wykonanych prób optymalizacyjnych można określić optymalne parametry dla algorytmu. Są nimi współczynnik powyżej 0.99, ilość iteracji nie jest ograniczona z góry, im więcej tym algorytm będzie miał więcej czasu, więc wzrośnie prawdopodobieństwo znalezienia najlepszego rozwiązania. Natomiast temperatura początkowa jest znacznie trudniejsza do określenia. Bez dokładniejszych testów – zwłaszcza testów połączonych z testami współczynnika nie da się jednoznacznie określić kiedy jaką wartość powinien przyjąć ten parametr. Zgodnie z przeprowadzonymi testami wartość temperatury może być dowolna ale większa niż 20000 (ominięcie maksimum).