

# Projektowanie Efektywnych Algorytmów

## Projekt 2

W ramach projektu dokonałem implementacji oraz analizę efektywności algorytmu **Symulowanego Wyżarzania (Simulated Annealing)** dla asymetrycznego problemu komiwojażera (ATSP).

### 1. Wstęp teoretyczny

Algorytm Symulowanego Wyżarzania został po raz pierwszy opisany w 1953r. przez Metropolisia. Swoją nazwę i sposób działania zawdzięcza analogi do zjawisk fizycznych jakim jest proces ochładzania i stygnięcia metalu. Algorytm polega na ciągłym poprawianiu i ulepszaniu bieżącego rozwiązania. Potrafi on wychodzić z minimum lokalnego, w celu optymalizacji w kierunku minimum globalnego. Dodatkowo zaimplementowałem zapamiętywanie dotychczasowej ścieżki o najmniejszym koszcie co sprawia, że wyniki są bliższe optymalnemu rozwiązaniu.

### 2. Założenia projektu

W swoim pomiarze czasu, jako temperaturę początkową przyjąłem  $T_0 = 1000$ , Długość epoki  $L = 1000$ , współczynnik redukowania temperatury  $\alpha = 0,95$ . Chłodzenie temperatury odbywa się na podstawie mnożenia temperatury przez współczynnik  $\alpha$ . Jest to tak zwane chłodzenie geometryczne. Do pomiaru czasu przyjąłem generowanie 10 losowych instancji (wynik nie wpływa znacząco na średnią) dla każdej wielkości  $N = \{10, 15, 20, 25, 30, 35, 40\}$

### 3. Sposób działania algorytmu

W programie użyłem wskaźników do dynamicznego stworzenia tablic. Metoda shuffle losuje pierwszą losową permutację. Metodą neighbourPermutation wyznaczam losowo sąsiada bieżącej permutacji sposobem swap. Następnie sprawdzam różnicę pomiędzy bieżącym a sąsiednim rozwiązaniem. Kiedy ta różnica jest mniejsza przypisuję do bieżącego rozwiązania, odpowiednie rozwiązanie sąsiednie. Zawsze w przejściu pętli porównuję najmniejsze dotychczasowe znalezione rozwiązanie z rozwiązaniem bieżącym. Robię tak dlatego by rozwiązanie końcowe było bliżej rozwiązaniu optymalnemu. Sposobem wyjścia z minimum lokalnego stosuje odpowiednie porównanie przy każdym przejściu pętli:  $x < \exp\left(\frac{-\delta}{t}\right)$

Gdzie:

x- losowa liczba zmiennoprzecinkowa z przedziału (0;1)

$\delta$ - różnica rozwiązania bieżącego z sąsiednim

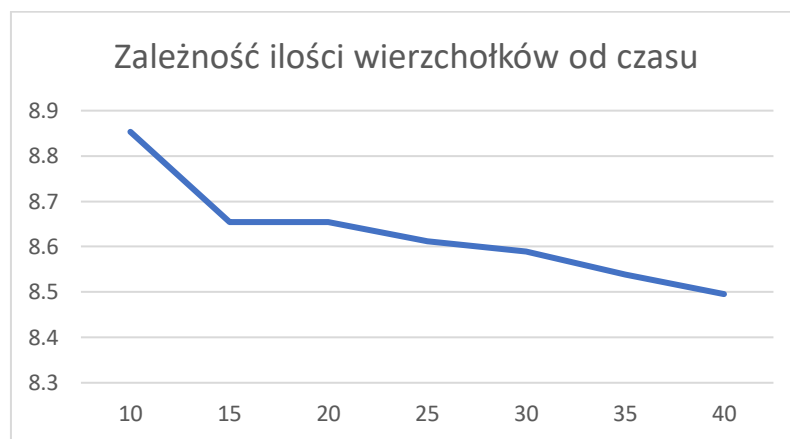
t- temperatura

Następnie po wyjściu z pętli wewnętrznej temperatura jest chłodzona geometrycznie

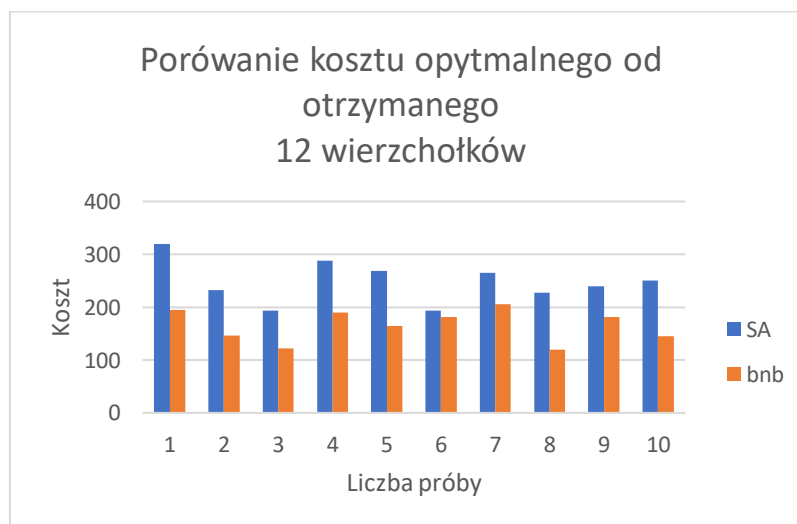
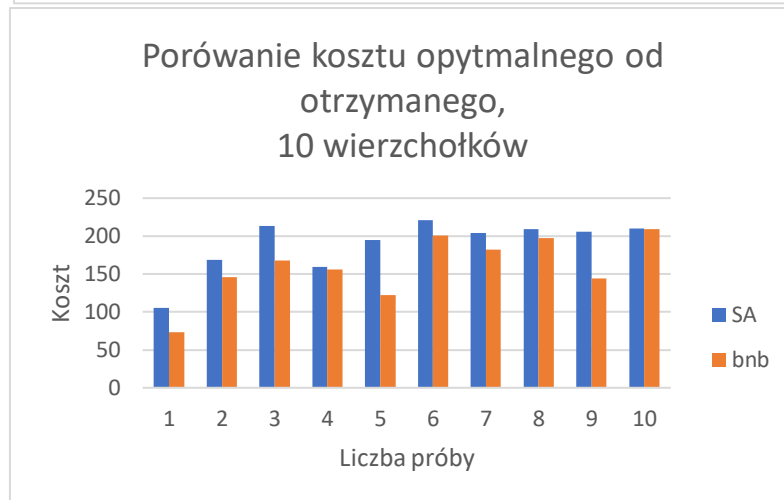
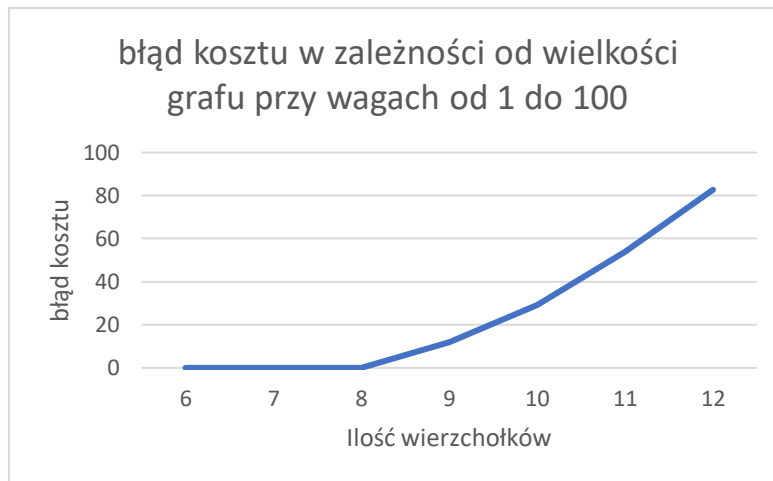
#### 4. Wyniki pomiarów

Pomiar czasu w zależności od ilości wierzchołków

N	czas[s]
10	8,853289
15	8,654103
20	8,654957
25	8,611573
30	8,589829
35	8,537863
40	8,495498



Jak widząc na powyższym wykresie, ilość wierzchołków nie wpływa zbyt mocno na czas działania algorytmu. Dzieje się tak ze względu na implementację, ilość wywołań pętli jest stała dla każdej wielkości. Sprawia to, że dla większej ilości wierzchołków znaleziony koszt zaczyna odbiegać coraz bardziej od optymalnego kosztu co pokazuje poniższy graf.



## 5. Wnioski

Algorytm Symulowanego Wyżarzania radzi sobie z taką ilością wierzchołków z jakimi nie radzą sobie algorytmy b&b i Brute Force. Czas oczekiwania tych dwóch jest znacznie większy niż symulowanego wyżarzania dla większej ilości wierzchołków. Jednak dzieje się to kosztem wytworzenia rozwiązania zbliżonego a nie rozwiązania optymalnego.