

Projektowanie Efektywnych Algorytmów-Projekt 2

Temat: Algorytm oparty na metodzie poszukiwania lokalnego

Opracował: Miłosz Siofer

Numer Indeksu:256794

Prowadzący: Mgr inż. Łukasz Kamiński

1. Wstęp teoretyczny

W tym projekcie pragnę przedstawić rozwiązanie do projektu nr 2, dotyczące poszukiwania lokalnego za pomocą dwóch algorytmów:

- Symulowane wyżarzanie (ang. Simulated Annealing)– jedna z technik projektowania algorytmów heurystycznych (metaheurystyka). Cechą charakterystyczną tej metody jest występowanie parametru sterującego zwanego temperaturą, który maleje w trakcie wykonywania algorytmu. Im wyższą wartość ma ten parametr, tym bardziej chaotyczne mogą być zmiany. Jest to metoda iteracyjna: najpierw losowane jest pewne rozwiązanie, a następnie jest ono w kolejnych krokach modyfikowane. Jeśli w danym kroku uzyskamy rozwiązanie lepsze, wybieramy je zawsze. Istotną cechą symulowanego wyżarzania jest jednak to, że z pewnym prawdopodobieństwem może być również zaakceptowane rozwiązanie gorsze (ma to na celu umożliwienie wyjście z maksimum lokalnego).
- Przeszukiwanie Tabu (ang. Tabu Search) – to algorytm wyszukiwania w sąsiedztwie, który wykorzystuje listę tabu. Lista tabu reprezentuje zestaw potencjalnych rozwiązań, których wyszukiwanie jest niedozwolone w przypadku wielu kroków (czas używania karty). Proces podejmowania decyzji na krok jest podobny do zachłannych algorytmów, ale z listą niedozwolonych przenosin (zazwyczaj ostatnio odwiedzone). Przy każdym czyszczeniu wykonywane jest przeniesienie, które powoduje poprawę najlepszej funkcji celu i nie znajduje się na liście tabu.

2. Plan eksperymentu

Założenia projektowe:

- Struktury przechowujące dane alokowane są dynamicznie, zależnie od rozmiaru problemu;
- Program posiada możliwość wczytywania danych z pliku, w celu weryfikacji poprawności;
- Algorytm został zaprojektowany zgodnie z paradygmatami programowania obiektowego;
- Program umożliwia włączenie/wyłączenie dywersyfikacji w algorytmie Tabu Search;

W programie znajdują się następujące klasy:

- Menu –interfejs użytkownika,
- SA –algorytm symulowanego wyżarzania,
- Tabu –algorytm przeszukiwania tabu search;

3. Plan testów

Aby wykonać pomiary skorzystałem pliki z macierzami br17,ftv70,ftv170 znajdujące się na następującej stronie:<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/atsp/index.html> .

Mierzenie czasu odbywało się przy pomocy funkcji QueryPerformanceCounter() oraz QueryPerformanceFrequency() z biblioteki <Windows.h>. Aby uzyskać czas należało zliczoną liczbę impulsów licznika podzielić przez częstotliwość. Zastosowane rozwiązanie pozwoliło mi na odpowiednie badanie czasu wykonania algorytmu. Poniżej w punktach 4 i 5 zamieściłem skutki moich badań w postaci tabel oraz wykresów:

4. Tabele wyników testów

Dla br17.astp

Nazwa klasy	Czas(w sekundach)
Symulowanie wyżarzania	0,0005421s
Tabu Search	0,0014487s

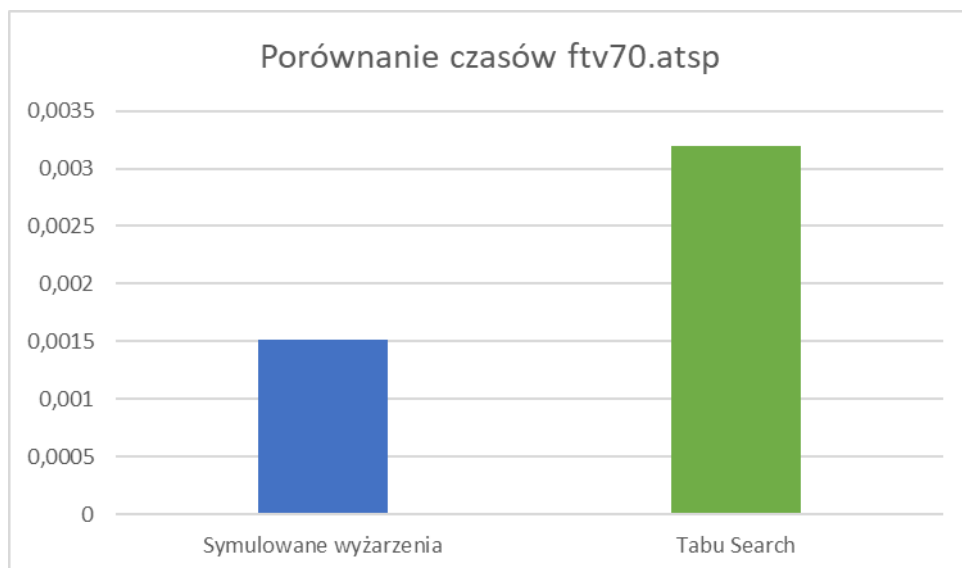
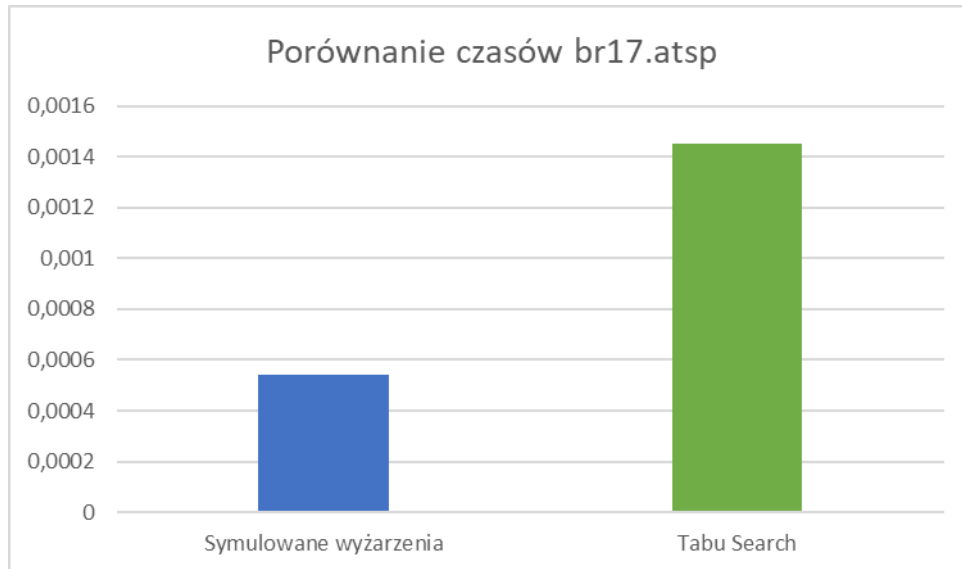
Dla ftv70.astp

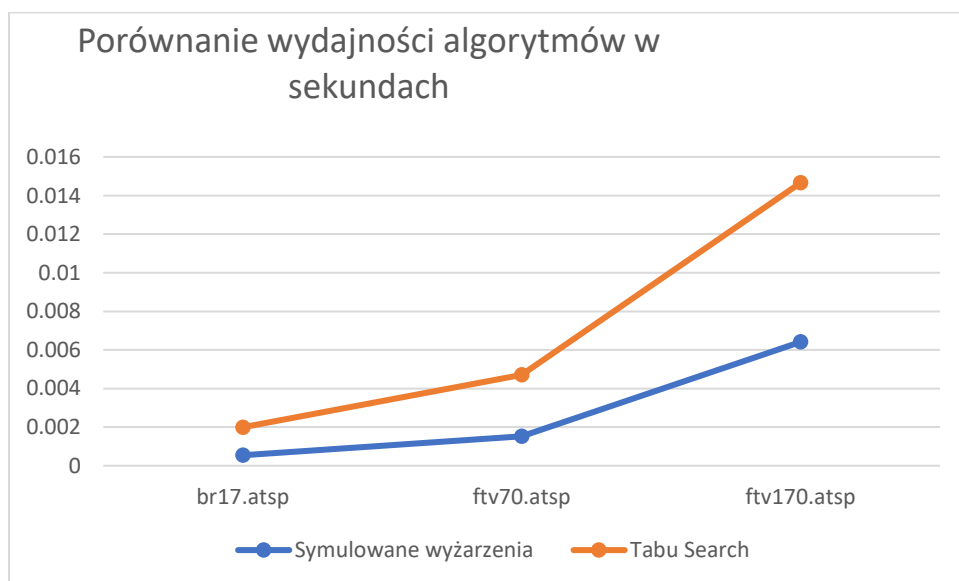
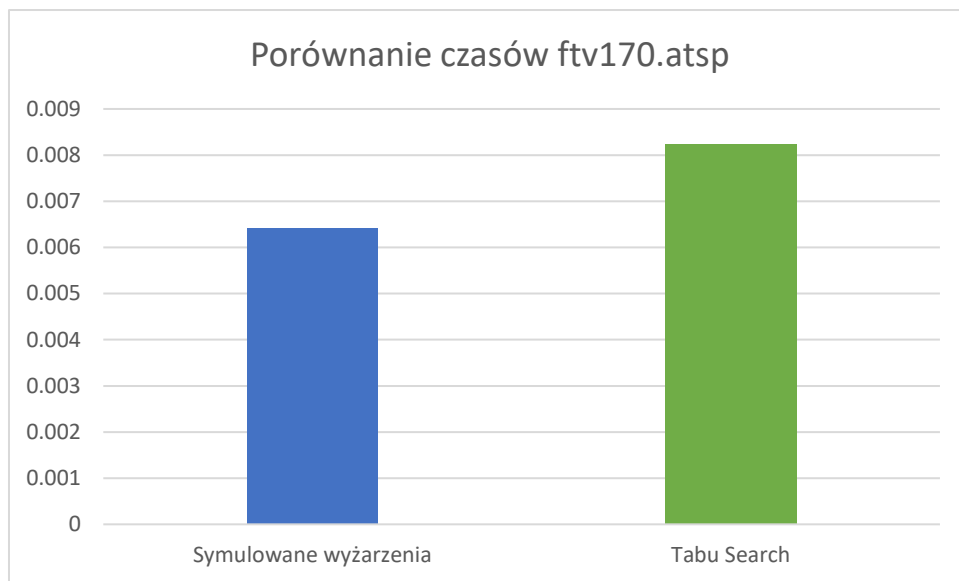
Nazwa klasy	Czas(w sekundach)
Symulowanie wyżarzania	0,0015149s
Tabu Search	0,00319554s

Dla ftv170.astp

Nazwa klasy	Czas(w sekundach)
Symulowanie wyżarzania	0,006411s
Tabu Search	0,0082471s

4.1 Wykresy wyników testów





5.Wnioski

Algorytmy poszukiwania lokalnego nie są dokładne, ponieważ nie zawsze znajdują najlepsze rozwiązanie. W zamian znajdują rozwiązania bliskie optymalnemu w krótkim czasie (w porównaniu do algorytmów dokładnych). Implementacja algorytmu to jedynie połowa sukcesu. W zależności od rozmiaru grafu. Gdy korzystałem wyżej wymienione algorytmy opierałem się na ich szybkości. Algorytmy te są na tyle szybkie, że najlepszym rozwiązaniem jest uruchamiać program wiele razy dla tego samego problemu i wyciągać optymalne rozwiązanie.

