

Politechnika Wrocławska
Wydział Elektroniki

Projektowanie Efektywnych Algorytmów

Projekt 2

Autor: Wojciech Wójcik 235621

Termin: wt. 18:55
Prowadzący: dr inż. Jarosław Rudy

18 grudnia 2018

Spis treści:

1. Założenia projektowe	2
2. Wstęp teoretyczny	2
2.1. Tabu Search	2
2.2. Symulowane wyżarzanie	2
3. Pomiary	3
3.1. Pomiary dla instancji z projektu pierwszego.	3
3.2. Pomiary dla instancji ftv47	4
4. Wnioski	6

1. Założenia projektowe

Zadaniem projektowym była implementacja dla problemu komiwojażera algorytmów Tabu Search i Symulowanego Wyżarzania (ang. Simulated Annealing) oraz porównanie efektywności obu algorytmów, a także wyznaczenie błędu względnego (porównując z najlepszym znanym rozwiązaniem).

2. Wstęp teoretyczny

2.1. Tabu Search

Tabu Search może być określona jako metoda dynamicznej zmiany sąsiedztwa danego rozwiązania, co oznacza że $N(x)$ nie jest stałe (jednorazowo określone) dla każdego x , lecz może zmieniać się w zależności od informacji na temat rozwiązywanego problemu, zgromadzonych na wcześniejszych etapach poszukiwania. Jest metaheurystyką do rozwiązywania problemów optymalizacyjnych, opartą na iteracyjnym przeszukiwaniu przestrzeni rozwiązań, wykorzystującą (zmienne) sąsiedztwo pewnych rozwiązań, zapamiętującą niektóre ruchy (transformacje rozwiązań) i częstość ich występowania, w celu unikania minimów lokalnych i poszukiwania rozwiązań globalnie optymalnych w rozsądnym czasie.

Lista Tabu reprezentowana jest za pomocą tablicy, do której wpisuje się kadencje dla zamienionych wierzchołków w permutacji miast. Zrealizowana została prosta funkcja aspiracji, której działanie reguluje parametr. Odpowiada on za sprawdzanie, czy dane przejście, mimo iż znajdujące się na liście Tabu, znacząco poprawia rozwiązanie. Parametr ten może być dowolnie regulowany, lecz nie powinien być większy od jedynki, gdyż wtedy możemy zmienić obecne rozwiązanie na gorsze. Prosto i intuicyjnie zrealizowana została funkcja CriticalEvent, która w przypadku braku znalezienia lepszego rozwiązania w otoczeniu, losuje nową permutację początkową. Po ewentualnym przekroczeniu tej liczby, program kończy pracę, gdyż nie może znaleźć lepszego rozwiązania niż obecne. Jeżeli zmiana permutacji miast przyniesie skutek, licznik wywołań CriticalEvent jest zerowany. Warunek zakończenia określony jest poprzez określoną ilość iteracji dla danej instancji.

2.2. Symulowane wyżarzanie

Algorytm Symulowanego Wyżarzania jest szczególnym przypadkiem algorytmu genetycznego. Nawiązuje do zjawiska fizycznego - zastygania cieczy tworzącej uporządkowaną, krystaliczną strukturę. W wysokiej temperaturze cząsteczki cieczy poruszają się swobodnie, ale wraz ze spadkiem temperatury możliwości ruchowe oraz prędkość poruszania się cząsteczek spada. Operacja powolnego schładzania ma na celu doprowadzenie metalu do równowagi termodynamicznej w stosunku do stanu wyjściowego oraz osiągnięcie pożądanych cech. Cały proces jest sterowany przez parametr zwany temperaturą.

Dla każdej iteracji wyznaczana jest nowa temperatura, która jest wyliczana wzorem $T = T * s$, gdzie s to sztywno zadana przez nas stała. Jest to sposób geometryczny wyznaczania temperatury. Iteracje wykonują się dopóki wyliczona temperatura jest większa od sztywno zadanej w kodzie temperatury końcowej (w programie użyta wartość temperatury końcowej $T_k = 0.01$). Taka niska temperatura końcowa sprawia, że niemal niemożliwa jest zmiana rozwiązania na gorsze. Wielkość stałej s wpływa na szybkość schładzania jak i na długość wykonywania algorytmu.

3. Pomiar

3.1. Pomiar dla instancji z projektu pierwszego.

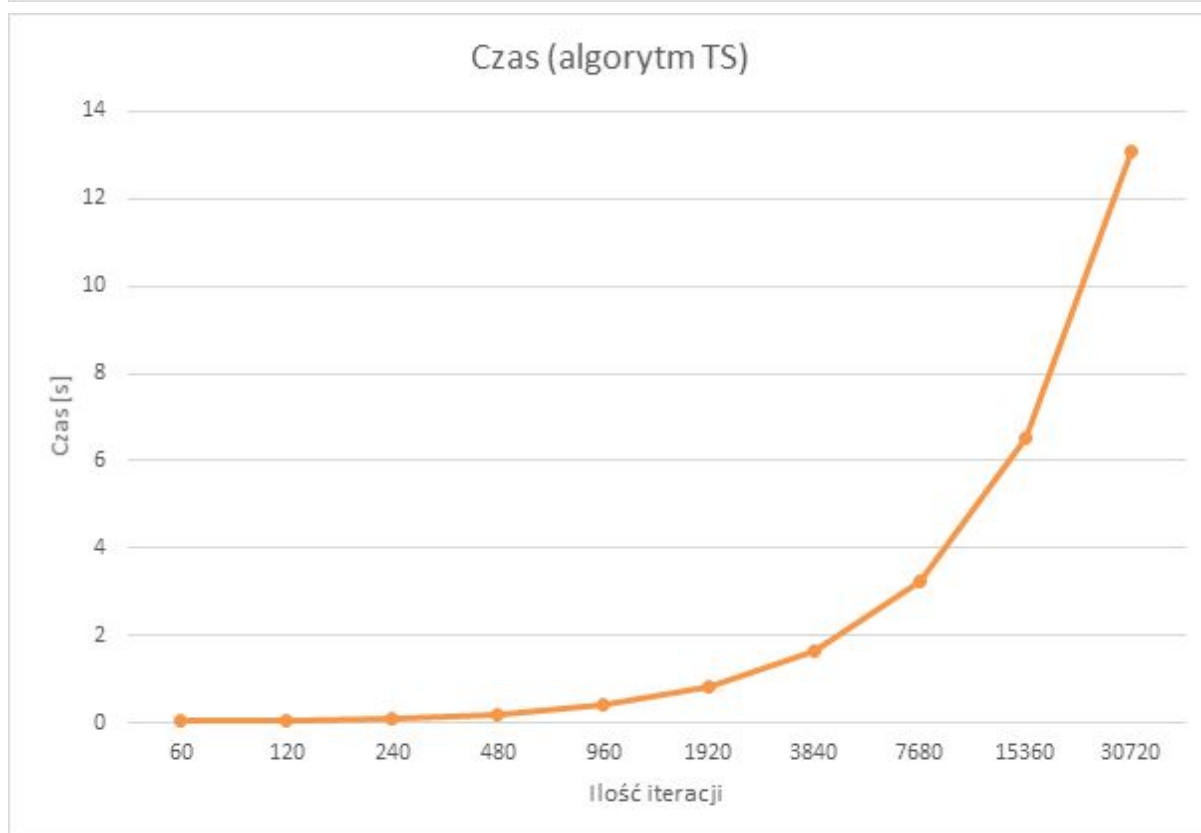
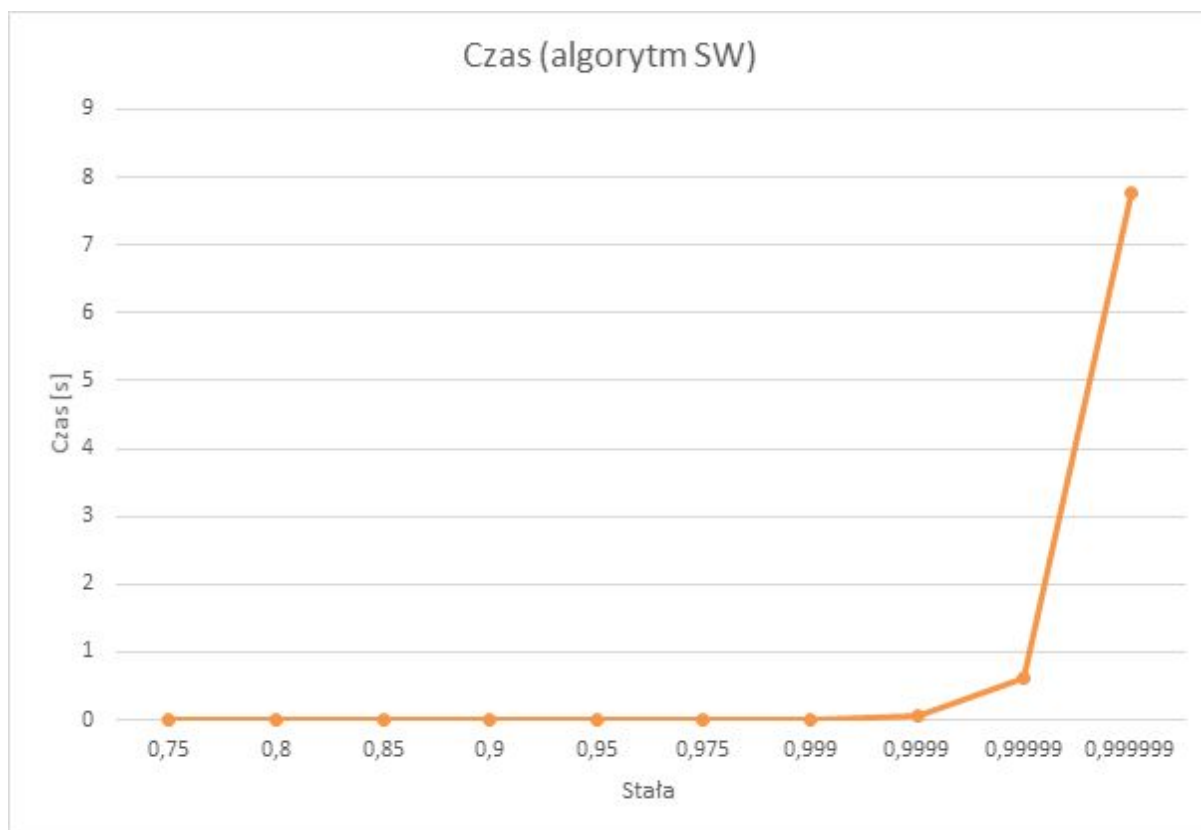
Tabela 1: Pomiar SW dla instancji z pierwszego projektu

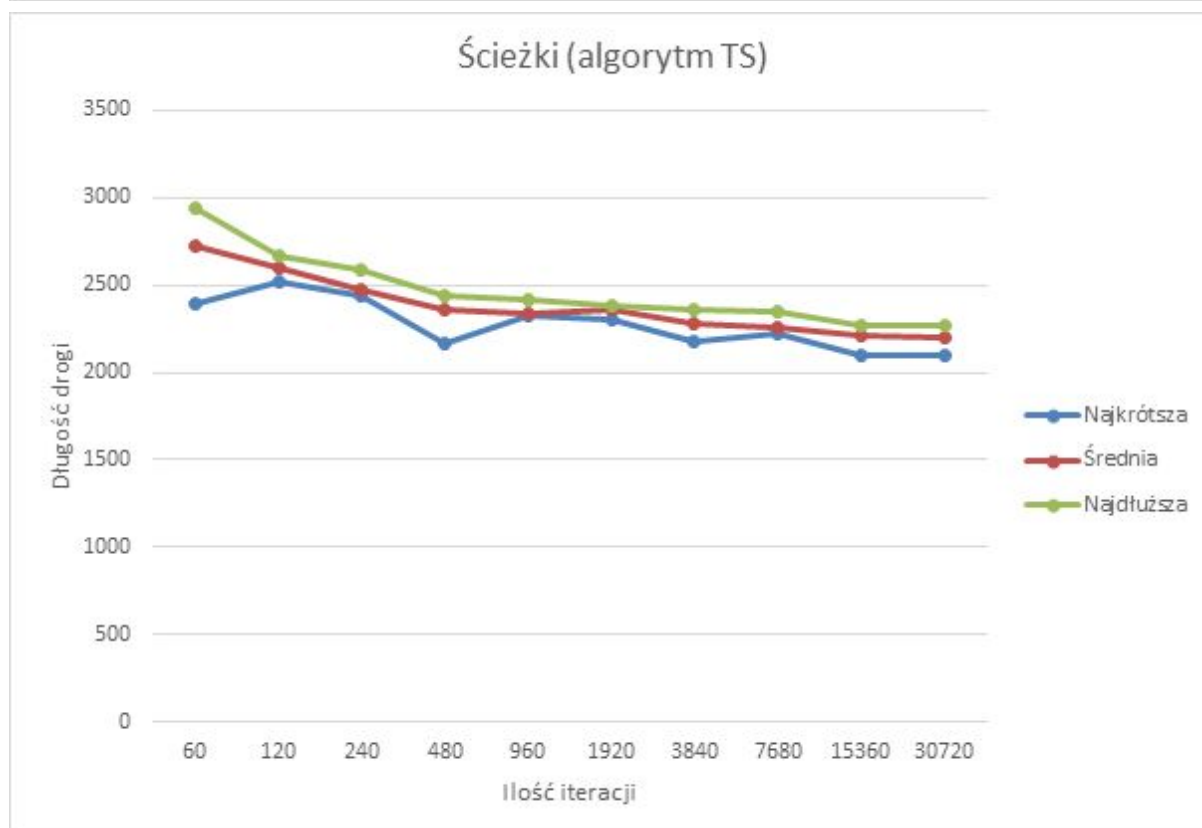
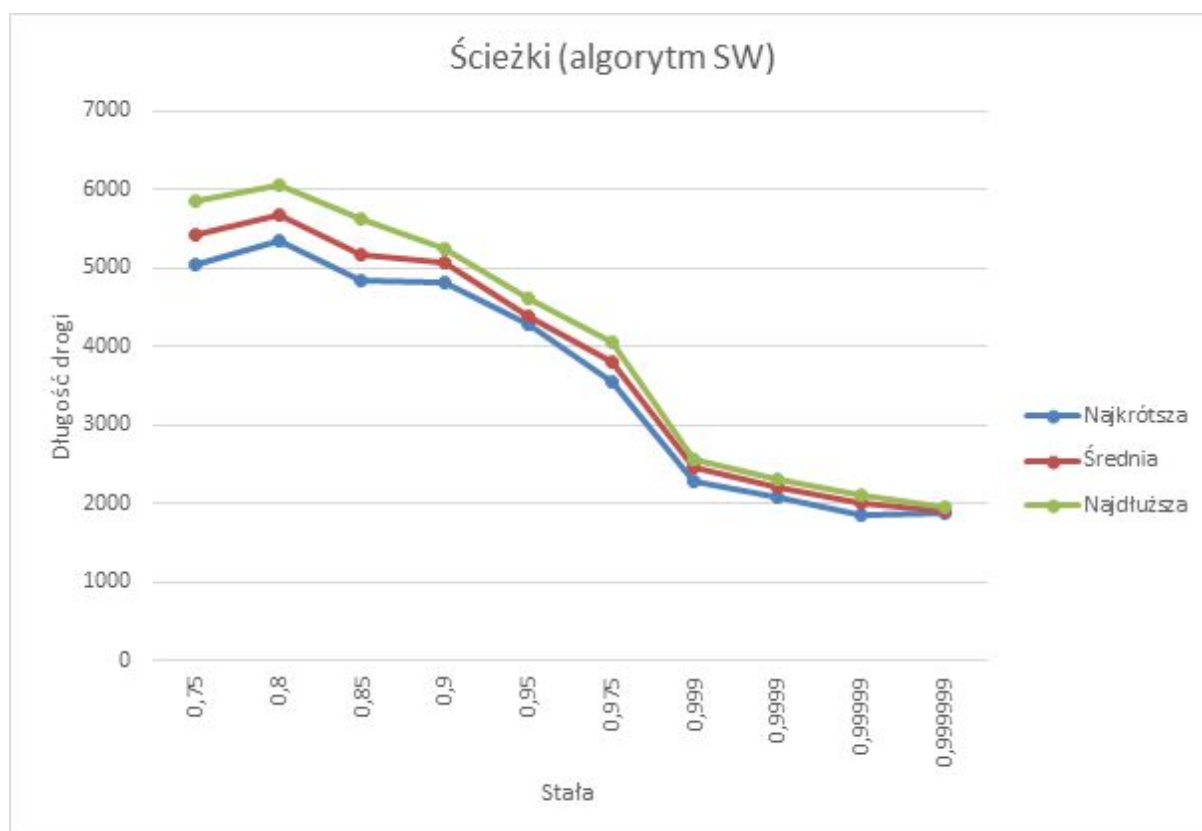
Rozmiar	Uzyskane średnie rozwiązanie	Czas [ms]	Błąd względny [%]
tsp_10	215	190	~1%
tsp_15	307	242	5,49%
tsp_17	41	261	5%

Tabela 2: Pomiar TS dla instancji z pierwszego projektu

Rozmiar	Uzyskane średnie rozwiązanie	Czas [ms]	Błąd względny [%]
tsp_10	212	35	0%
tsp_15	312	262	7,29%
tsp_17	44	329	12%

3.2. Pomiary dla instancji ftv47





4. Wnioski

Algorytmy nie gwarantują uzyskania najlepszej drogi. Ich wadą jest mała losowość - mimo tych samych parametrów jest możliwe uzyskanie różnych wyników - zależy to między innymi od początkowej permutacji drogi. W algorytmie Tabu Search wahania wyników wydają się być większe niż w algorytmie Symulowanego Wyżarzania. Może być to związane z doбором argumentów do argumentów - nie da ich się łatwo porównać.

Dużym plusem tych algorytmów metaheurystycznych jest możliwe uzyskanie przybliżonej odpowiedzi na zadany problem w akceptowalnym czasie. Porównując je do rozwiązań problemu komiwojażera z projektu pierwszego tego przedmiotu można łatwo zauważyć, że błąd względny wynosi maksymalnie 10% (przy instancji 17 miast, czas wykonywania algorytmów poniżej 1s).

W obu algorytmach występuje wzrost czasu wykonania algorytmu proporcjonalny do ilości miast. Jest to związane choćby z najprostszą operacją obliczania długości drogi, która podczas jednej iteracji jest wykonywana wielokrotnie.

Oba algorytmy po znalezieniu lokalnego minimum (które może być minimum globalnym), starają się szukać innej, bardziej optymalnej drogi. Mechanizm ten ma za zadanie zapobiec utknięciu w lokalnym minimum.

Dużą zaletą algorytmów jest mała złożoność pamięciowa.