

Projektowanie Efektywnych Algorytmów

Projekt

26/01/2023

259193 Kacper Wróblewski

Dodatek

<i>Spis treści</i>	<i>strona</i>
Opis badań	2
Porównanie złożoności czasowej	3
Porównanie złożoności pamięciowej	4
Wnioski	5

1. Opis badań

Metody zastosowane do rozwiązywania TSP zostały zestawione na wykresie zależności wielkości instancji od czasu wykonywania programu w celu porównania ich złożoności czasowej (wykres 1.) oraz na wykresie zależności wielkości instancji od zajętości pamięci procesu badanej instancji problemu (wykres 2.).

Wyniki zostały opracowane w programie MS Excel.

Zebrane dane pochodzą z badań przeprowadzonych podczas wykonywania zadań numer: 1., 2., 4. oraz 7.

Oznaczenia metod na wykresie:

- Brute-Force – przegląd zupełny (zadanie 1.),
- Held-Karp – programowanie dynamiczne (zadanie 2.),
- SA – symulowane wyżarzanie (ang. simulated annealing, zadanie 4.),
- ACO – algorytm mrówkowy (ang. ant colony optimization, zadanie 7.)

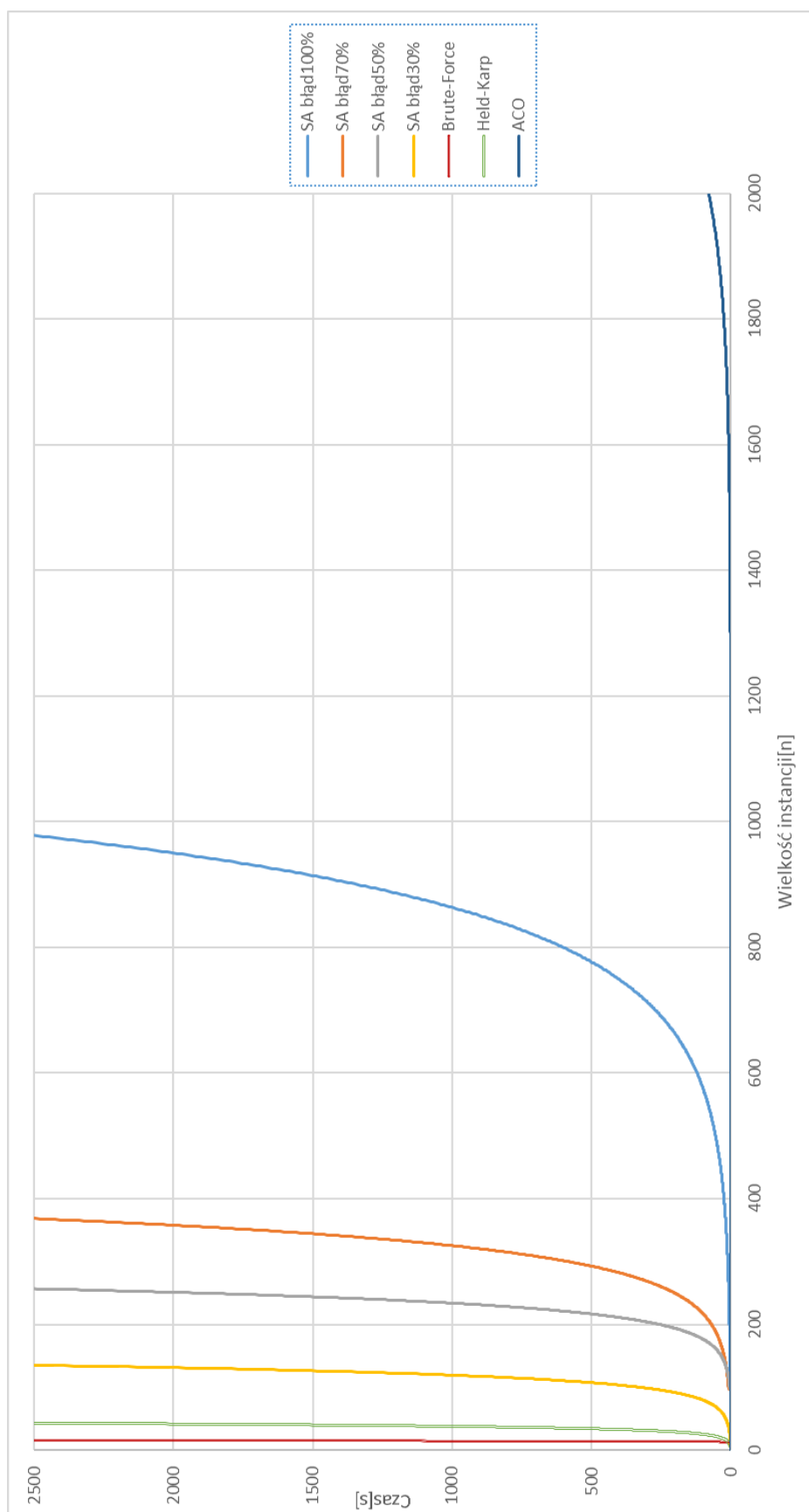
Algorytm symulowanego wyżarzania został przebadany kilka razy dla różnych wartości jego parametrów, aby ustalić złożoność czasową w zależności od wartości stopy błędu. Był on badany dla ustawienia współczynników dających maksymalną wartość błędu kolejno: 100%, 70%, 50% oraz mniej niż 30%.

Algorytm mrówkowy był badany raz, z zachowaniem założeń opisanych w zadaniu.

Pozostałe metody, jako że nie są algorytmami heurystycznymi, a dającymi dokładne wyniki, zostały przebadane raz.

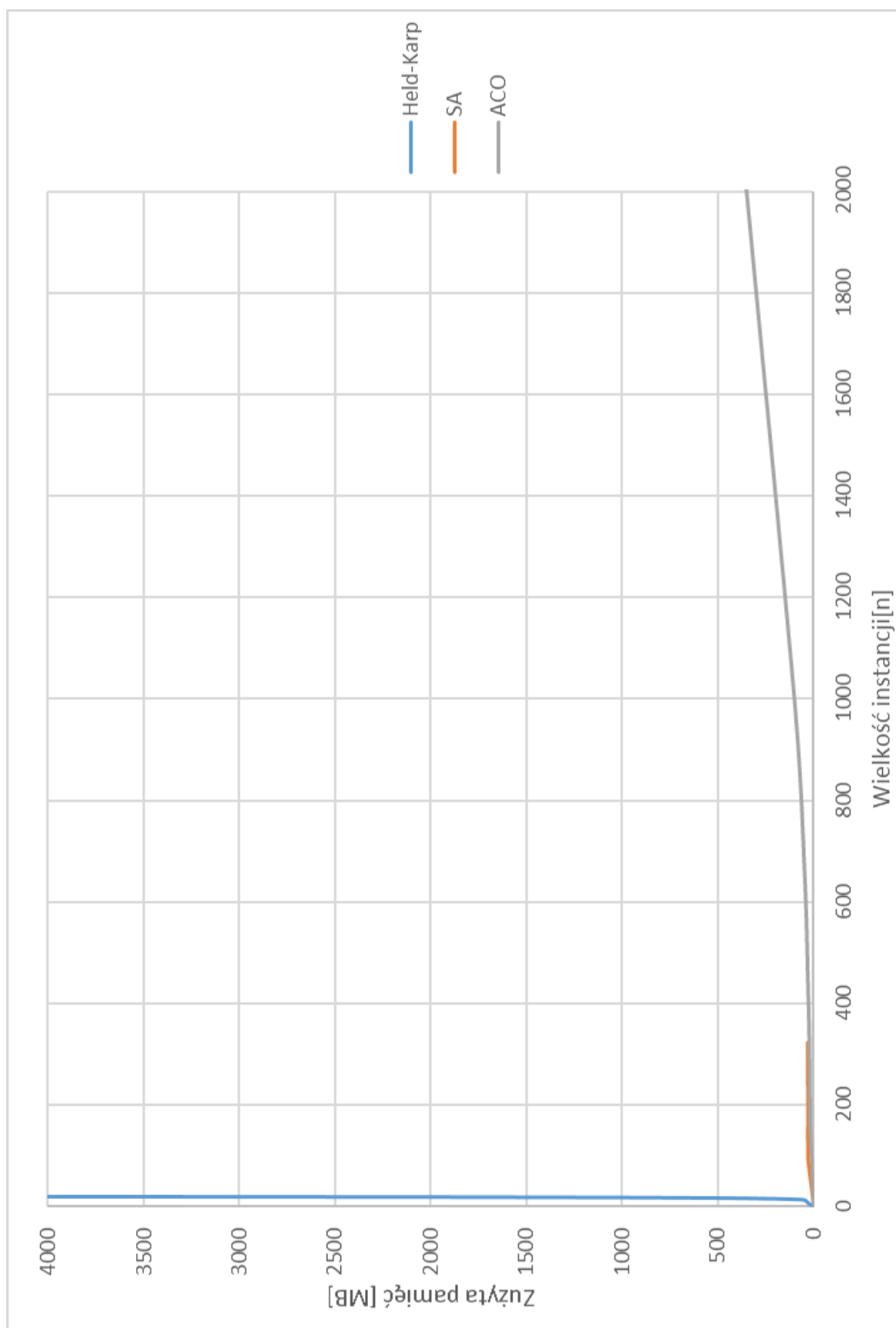
Badanie pamięciowości algorytmów programowania dynamicznego, symulowanego wyżarzania oraz ACO wykonano za pomocą narzędzia menedżer zadań Windows, który mierzył ilość pamięci potrzebnej na działanie procesu wykonującego algorytm z jedną instancją problemu.

2. Porównanie złożoności czasowej:



Wykres 1. Zestawienie wszystkich badanych metod w funkcji wielkości instancji od czasu

3. Porównanie złożoności pamięciowej:



Wykres 2. Zestawienie złożoności pamięciowej

4. Wnioski

Najgorszą złożoność ze wszystkich zbadanych algorytmów ma metoda przeglądu zupełnego, ponieważ jej krzywa rośnie najszybciej ze wszystkich (wykres 1.), potwierdzając jego teoretyczną złożoność $O(n!)$. Nieco lepszą zależność wykazuje algorytm Held-Karpa o złożoności $O(2^n \cdot n^2)$. Dzieje się to jednak kosztem zużycia pamięci przez algorytm. Posiada on najszybciej rosnącą krzywą wykładniczą pamięciowości ze wszystkich zbadanych metod (wykres 2.).

Algorytm mrówkowy oraz symulowane wyżarzanie posiadają podobną złożoność pamięciową, rosnącą wolno, co sugeruje, że nie należy się martwić o pamięć przy stosowaniu tych metod. Należy jednak mieć na uwadze, że wraz ze zmianą parametrów określających ilość iteracji tych metod koszt pamięci będzie rósł.

Algorytmy heurystyczne, jakimi są algorytmy SA oraz ACO, nie dają gwarancji znalezienia optymalnego rozwiązania, jednak działają w czasie znacznie krótszym, np. dla instancji rozmiaru 14, algorytm mrówkowy jest w stanie znaleźć rozwiązanie w czasie 20 tys. razy krótszym w porównaniu do przeglądu zupełnego.

Te algorytmy posiadają parametry wpływające na czas działania oraz jakość rozwiązania, co widać na wykresie 1. Symulowane wyżarzanie wraz ze wzrostem dokładności działa dłużej. Podobne wnioski można wyciągnąć dla algorytmu mrówkowego.

Algorytm mrówkowy osiągający złożoność $O(CC \cdot n^3)$ (CC to ilość iteracji podana przez użytkownika) daje zadowalające wyniki dla największych metod w najkrótszym czasie. Jest on zatem najlepszą metodą dla najbardziej złożonych instancji. Natomiast algorytmy przeglądu zupełnego oraz programowania dynamicznego, mimo tego, że nie radzą sobie przy instancjach większych niż 14, są dobrą metodą dającą dokładne wyniki w zadowalającym czasie dla małych problemów.