**Data wykonania** **sprawozdania:** 24.03.2019

**Sprawozdanie**

**Temat:** Algorytm NEH

**Grupa:**

1. Fryderyk Fogel
2. Jakub Dąbrowski

Algorytm NEH głównie zajmuje się problemem przepływowym, gdzie sumuje czasy wykonywania poszczególnych zadań na wszystkich maszyn, a następnie sortuje je malejąco. Działanie bez żadnych modyfikacji wykorzystuje dwa pierwsze zadania maszyn o najdłuższym czasie pracy i liczy wartość Makespan dla wszystkich permutacji. Najmniejsza wartość daje najlepszą kolejność wykonania zadań. Następnie dokładając na koniec kolejne zadania i sprawdzając każde możliwe ustawienie, sprawdza kolejne wartości Makespan. Wszystko jest wykonywane w pętli aż do sprowadzenia wszystkich zadań i obliczenia najlepszej konfiguracji.

W zestawieniu NEH’a do innych metod czyli przeglądu zupełnego oraz algorytmu Johnsona funkcjonuje on najlepiej i najskuteczniej. Przegląd zupełny jest najwolniejszy – przeszukuje wszystkie możliwe permutacje zadań, natomiast algorytm Johnsonsa jest wykorzystywany tylko i wyłączenie dla dwóch lub trzech maszyn. W instancji ta000 gdzie występowały trzy maszyny Johnson poradził sobie bardzo podobnie jak NEH uzyskując taki sam wynik.

Odnośnie testów algorytmu NEH czy działa poprawnie, został on podany porównaniu z programem NehDemo.exe. Wszystkie wyniki, czyli cmax dla najlepszej permutacji oraz najlepsze ustawienie zadań potwierdzają dobre działanie naszego programu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instancja | ta000 | ta001 | ta060 | ta089 | ta120 |
| Czas wykonania funkcji | 0.4s | 0.3s | 4.1s | 31.5s | 1285.4s |
| Cmax | 32 | 1286 | 4079 | 6677 | 26 984 |

Można zauważyć, że dla początkowych instancji NEH bez modyfikacji radzi sobie dobrze, kłopot staje się, gdy mamy dużo większą liczbę zadań i maszyn. Algorytm ten nadaje się jedynie dla problemów, których ilość możliwych permutacji jest mała. Ulepszając NEH’a do zastosowania akceleracji zwiększamy złożoność obliczeniową – zmniejszamy czas wykonywania funkcji o około s.