

SPD - Lab nr 1 - Sprawozdanie

Marcin Wiśniewski (226524) Piotr Kida (226237)

22 marca 2018

Spis treści

1	Cel	2
2	Problem przepływowy	3
2.1	Algorytm Johnsona	3
2.2	Przegląd zupełny	3
3	Problem RPQ	4
3.1	SortR	4
3.2	Przegląd zupełny	4
4	Wnioski	5
4.1	Problem przepływowy (Flowshop)	5
4.2	Problem RPQ	5
4.3	Konkluzja	5

1 Cel

Celem pierwszych zajęć było stworzenie algorytmów rozwiązujących:

- I. Problem przepływowy,
- II. Problem RPQ.

Następnym zadaniem było przeanalizowanie poszczególnych parametrów funkcjonowania utworzonych programów oraz wzajemne porównanie uzyskanych wyników.

2 Problem przepływowy

Celem tej sekcji ćwiczenia było stworzenie algorytmów rozwiązujących problem Flowshop dla zestawu 2-maszynowego oraz 3-maszynowego. Algorytmy wykorzystane w naszej implementacji to:

- I. Algorytm Johnsona,
- II. Przegląd zupełny (Bruteforce).

2.1 Algorytm Johnsona

Utworzono dwie wersje Algorytmu Johnsona, wariant dla dwóch oraz dla trzech maszyn, wyniki pracy obu algorytmów przedstawione zostały w poniższej tabeli:

	2-maszynowy	3-maszynowy
Czas:	4,22ms	1,61ms
Cmax:	27	28

2.2 Przegląd zupełny

Zadaniem tego algorytmu było obliczenie wszystkich możliwych rozwiązań oraz dobór rozwiązania optymalnego, wyniki działania algorytmu realizującego przegląd zupełny przedstawione zostały w poniższej tabeli:

	2-maszynowy	3-maszynowy
Czas:	887,04ms	939,31ms
Cmax:	27	28

3 Problem RPQ

Celem tej sekcji ćwiczenia było stworzenie dwóch algorytmów rozwiązujących ten sam problem inną metodyką, metody te to:

- I. SortR,
- II. Przegląd zupełny (Bruteforce).

3.1 SortR

Zadaniem algorytmu SortR było posortowanie wszystkich zadań według rozmiaru zadanej zmiennej R, w kolejności rosnącej. Wyniki działania przygotowanego algorytmu przedstawione zostały w poniższej tabeli:

	SortR
Czas:	1,93ms
Cmax:	34

3.2 Przegląd zupełny

Zadaniem tego algorytmu było obliczenie wszystkich permutacji oraz zwrócenie wartości optymalnej, działanie algorytmu jest zaprezentowane w poniższej tabeli:

	Przegląd zupełny
Czas:	9603,51ms
Cmax:	34

4 Wnioski

4.1 Problem przepływowy (Flowshop)

Zgodnie z oczekiwaniami implementacja z zastosowaniem Algorytmu Johnsona wykonała swoją pracę w znacząco krótszym czasie niż implementacja typu Bruteforce, co spowodowane jest znacznie mniejszą ilością obliczeń wykonywanych przez algorytm. Wniosek ten jest zgodny zarówno dla setupu 2. jak i 3-maszynowego. Zaskoczeniem były czasy wykonywania programów z implementacją Algorytmu Johnsona, czas wykonywania obliczeń dla zestawu 3-maszynowego jest prawie 3. krotnie krótszy od czasu wykonywania problemu teoretycznie mniej skomplikowanego. Sytuacja ta może wynikać z nierównej optymalizacji utworzonych rozwiązań. Czasy te, patrząc przez pryzmat ich rozmiarów (milisekundy), nie różnią się od siebie na tyle znacząco by sytuacja ta potrzebowała dalszego wytłumaczenia. Algorytmy przeglądu zupełnego, poprzez fakt obliczania każdego możliwego rozwiązania problemu a następnie odnajdywania rozwiązania optymalnego, pracują w czasie niewspółmiernie większym od zastosowanego w pierwszym rozwiązaniu Algorytmu Johnsona, można więc bez większych zastanowień odpowiedzieć, że rozwiązanie wykorzystujące wcześniej wymieniony algorytm jest rozwiązaniem optymalnym w zbiorze założonych możliwości.

4.2 Problem RPQ

Wnioski do tej części zadania są praktycznie tożsame z wnioskami części poprzedzającej. W przypadku problemu RPQ, poza wykorzystaniem metody Bruteforce, korzystamy również z algorytmu wykorzystującego sortowanie po zmiennej R, w pracy tej nazywane algorytmem "SortR". Algorytm sortujący pracuje kilkakrotnie krócej od algorytmu przeglądu zupełnego, zwracając oczywiście taki sam wynik. Powody dla takiego stanu rzeczy wyjaśnione zostały w pierwszej sekcji tego rozdziału.

4.3 Konkluzja

Na podstawie przerobionych przykładów zauważamy dokładnie jak ważną kwestią w obróbce i optymalizacji danych jest fakt wykorzystywania przemyślanych algorytmów, które minimalizują ilość obliczeń potrzebnych do uzyskania optymalnego rozwiązania postawionego problemu.