

Akceleracja algorytmu *NEH*

Mariusz Makuchowski

25 listopada 2022

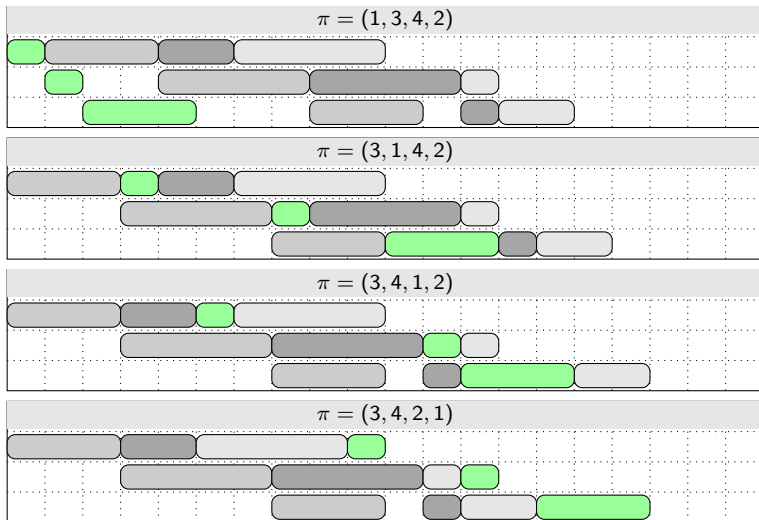
Algorytm NEH

Algorytm buduje rozwiązanie poprzez dokładanie jeszcze nieuszeregowanych zadań do bieżącej kolejności.

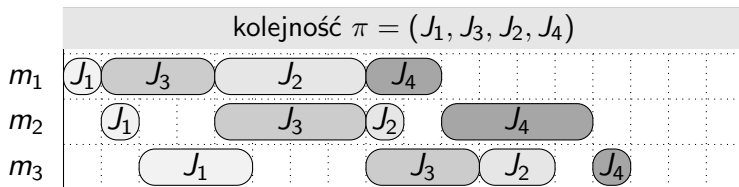
Przykład: $J = \{1, 2, 3, 4\}$

- krok 1: $\pi = (3)$
- krok 2: $\pi = (3, 2)$
- krok 3: $\pi = (3, 4, 2)$
- krok 4: $\pi = (1, 3, 4, 2)$

NEH: przykład krok 4



Obliczenie $C_{max}(\pi)$



	J_1	J_3	J_2	J_4
m_1	1/1	3/4	4/8	2/10
m_2	1/2	4/8	1/9	4/14
m_3	3/5	3/11	2/13	1/15

Złożoność obliczeniowa: naiwana implementacja

- n kroków
 - średnio $n/2$ próbnymi kolejnościami
 - dla każdej kolejności obliczamy C_{max}

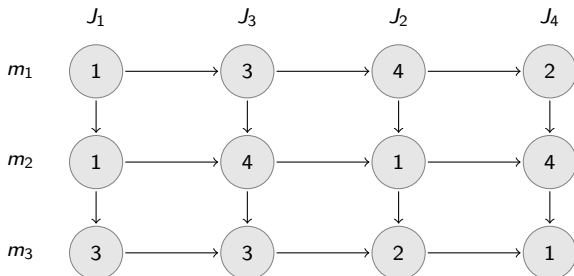
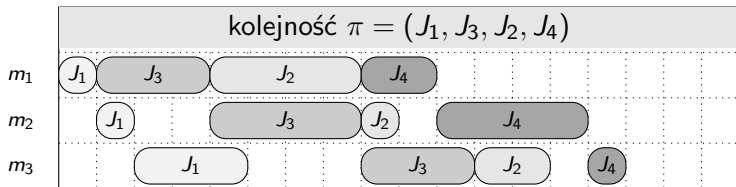
Ponieważ wyliczenie $C_{max}(\pi)$ zajmuje $O(nm)$ złożoność naiwnej implementacji NEH wynosi $O(n^3m)$

W danym kroku kolejności są "podobne".

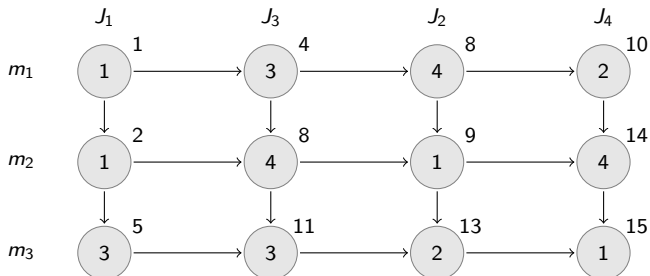
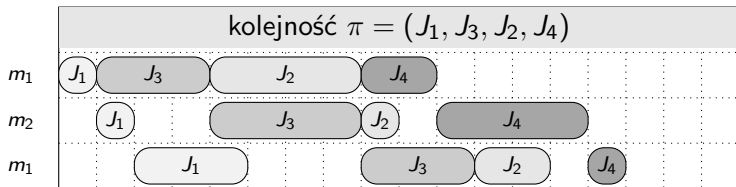
- (5,1,3,2,4)
- (1,5,3,2,4)
- (1,3,5,2,4)
- (1,3,2,5,4)
- (1,3,2,4,5)

Można to podobieństwo wykorzystać.

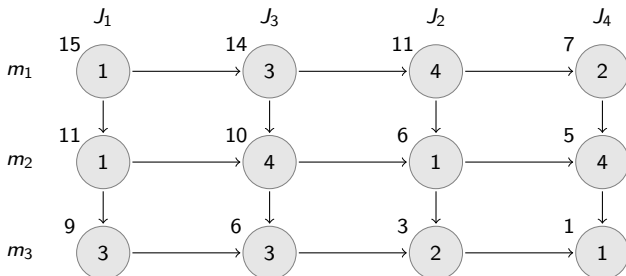
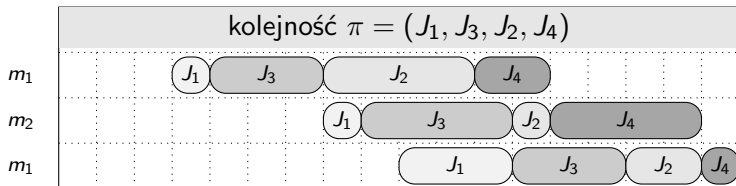
Model grafowy



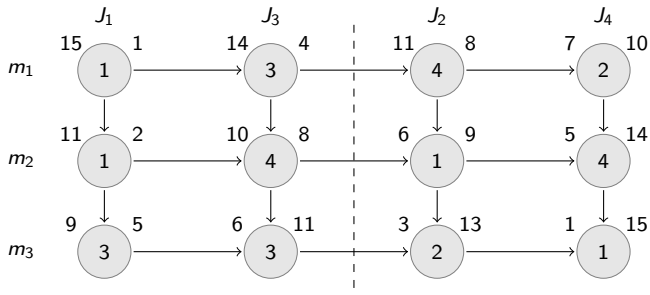
Model grafowy: najdłuższa dochodząca ścieżka



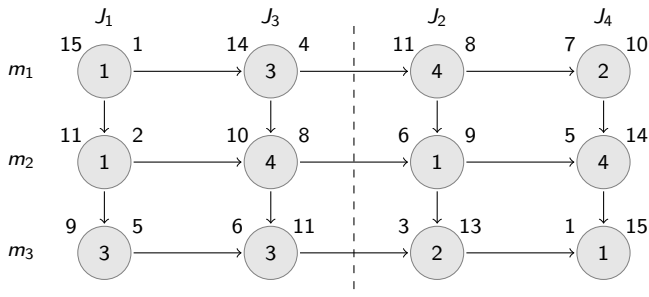
Model grafowy: najdłuższa wychodząca ścieżka



Model grafowy: cięcie grafu

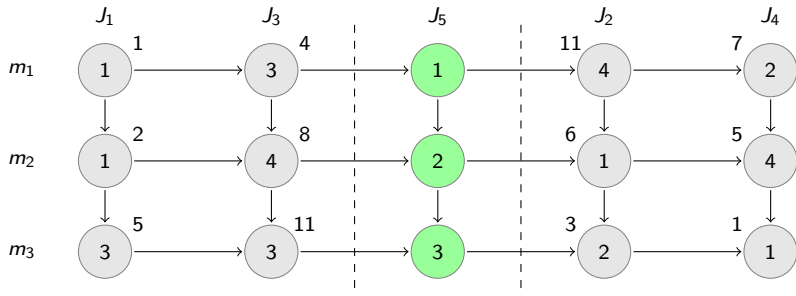


Model grafowy: cięcie grafu

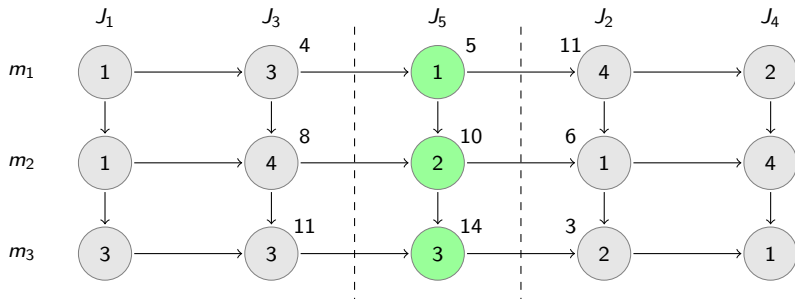


$$C_{max} = \max \left\{ \begin{array}{l} 4 + 11 \\ 8 + 6 \\ 11 + 3 \end{array} \right. = 15$$

Model grafowy: dodanie zadania



Model grafowy: dodanie zadania



$$C_{J_5, m_1} = 4 + 1 = 5$$

$$C_{J_5, m_2} = \max(5, 8) + 2 = 10$$

$$C_{J_5, m_3} = \max(10, 11) + 3 = 14$$

$$C_{\max} = \max \begin{cases} 5+11 \\ 10+6 \\ 14+3 \end{cases} = 17$$

Złożoność algorytmu: Quick NEH

- n kroków
 - dla każdego kroku obliczamy dwie tablice; $O(nm)$
 - średnio $n/2$ próbnymi kolejnościami
 - dla każdej kolejności obliczamy C_{max} ; $O(m)$

Całkowita złożoność algorytmu quick NEH to $O(n^2m)$

Działanie programu na CPU i5 2500MHz

Naiwny NEH

```
data.110 11869 0.076
data.111 26670 1.129
data.112 27232 1.185
data.113 26848 1.140
data.114 27055 1.132
data.115 26727 1.154
data.116 26992 1.173
data.117 26797 1.180
data.118 27138 1.200
data.119 26631 1.157
data.120 26984 1.132
TotalTime: 12.961s
```

Quick NEH

```
data.110 11869 0.002
data.111 26670 0.010
data.112 27232 0.011
data.113 26848 0.011
data.114 27055 0.009
data.115 26727 0.010
data.116 26992 0.010
data.117 26797 0.012
data.118 27138 0.010
data.119 26631 0.009
data.120 26984 0.009
TotalTime: 0.157s
```