# Szeregowanie zadań

## Wykład nr 6

dr Hanna Furmańczyk

# System otwarty - open shop

## $O3||C_{max}$

Problem $O3||C_{max}$ jest NP-trudny.

### $O3||C_{\max}$

Problem $O3||C_{\max}$ jest NP-trudny.

### Dowód

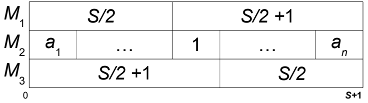
Redukcja $PP \to O3||C_{\max}$: bierzemy $n$ zadań o czasach $(0, a_i, 0)$ $i = 1, \ldots, n$ oraz oraz trzy zadania z czasami $(S/2, 1, S/2)$, $(S/2 + 1, 0, 0)$, $(0, 0, S/2 + 1)$. Pytamy o istnienie uszeregowania z $C_{\max} \leq S + 1$.

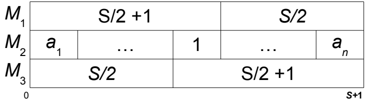
## $O3||C_{\max}$

Problem $O3||C_{\max}$ jest NP-trudny.

## Dowód

Redukcja $PP \rightarrow O3||C_{\max}$: bierzemy $n$ zadań o czasach $(0, a_i, 0)$ $i = 1, \ldots, n$ oraz oraz trzy zadania z czasami $(S/2, 1, S/2)$, $(S/2+1, 0, 0)$, $(0, 0, S/2+1)$. Pytamy o istnienie uszeregowania z $C_{\max} \leq S+1$.



# Minimalizacja $\sum C_j$

**Twierdzenie**

Problem $O2||\sum C_j$ jest NP–trudny.

## Algorytm Gonzalez-Sahni, $O(n)$

# $O2||C_{\max}$

## Algorytm Gonzalez-Sahni, $O(n)$

- Podziel zadania na zbiory $N_1 = \{Z_j : p_{1j} < p_{2j}\}$, $N_2 = \{Z_j : p_{1j} \geq p_{2j}\}$.

## Algorytm Gonzalez-Sahni, $O(n)$

- Podziel zadania na zbiory $N_1 = \{Z_j : p_{1j} < p_{2j}\}$, $N_2 = \{Z_j : p_{1j} \geq p_{2j}\}$.
- Wybierz 2 zadania $Z_r, Z_l$ takie, że: $p_{1r} \geq \max_{Z_j \in N_2} p_{2j}$; $p_{2l} \geq \max_{Z_j \in N_1} p_{1j}$.

# $O2||C_{\max}$

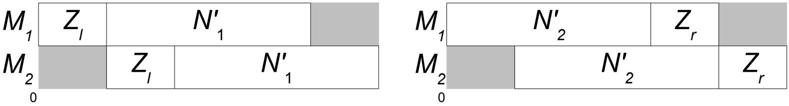
## Algorytm Gonzalez-Sahni, $O(n)$

- Podziel zadania na zbiory $N_1 = \{Z_j : p_{1j} < p_{2j}\}$, $N_2 = \{Z_j : p_{1j} \geq p_{2j}\}$.
- Wybierz 2 zadania $Z_r, Z_l$ takie, że: $p_{1r} \geq \max_{Z_j \in N_2} p_{2j}$; $p_{2l} \geq \max_{Z_j \in N_1} p_{1j}$.
- $p_1 := \sum_i p_{1i}$; $p_2 := \sum_i p_{2i}$; $N_1' := N_1 \backslash \{Z_r, Z_l\}$; $N_2' := N_2 \backslash \{Z_r, Z_l\}$; Dla $N_1' \cup \{Z_l\}$ i $N_2' \cup \{Z_r\}$ utwórz harmonogramy (permutacyjne i no–idle) z zadaniem z $\{Z_r, Z_l\}$ umieszczonym „z brzegu" .

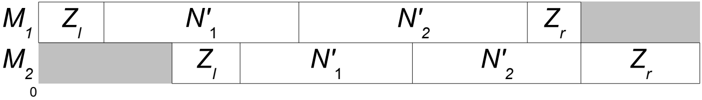
# $O2||C_{\max}$

## Algorytm Gonzalez-Sahni, $O(n)$

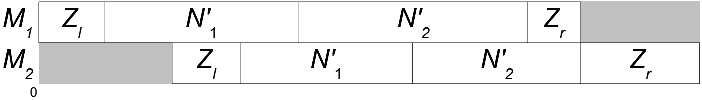
- Podziel zadania na zbiory $N_1 = \{Z_j : p_{1j} < p_{2j}\}$, $N_2 = \{Z_j : p_{1j} \geq p_{2j}\}$.
- Wybierz 2 zadania $Z_r, Z_l$ takie, że: $p_{1r} \geq \max_{Z_j \in N_2} p_{2j}$; $p_{2l} \geq \max_{Z_j \in N_1} p_{1j}$.
- $p_1 := \sum_i p_{1i}$; $p_2 := \sum_i p_{2i}$; $N_1' := N_1 \backslash \{Z_r, Z_l\}$; $N_2' := N_2 \backslash \{Z_r, Z_l\}$; Dla $N_1' \cup \{Z_l\}$ i $N_2' \cup \{Z_r\}$ utwórz harmonogramy (permutacyjne i no–idle) z zadaniem z $\{Z_r, Z_l\}$ umieszczonym „z brzegu" .



- Jeżeli $p_1 - p_{1l} \geq p_2 - p_{2r}$ ($p_1 - p_{1l} < p_2 - p_{2r}$), to "dosuń" operacje z $N'_1 \cup \{Z_l\}$ na $M_2$ w prawo (z $N'_2 \cup \{Z_r\}$ na $M_1$ w lewo)

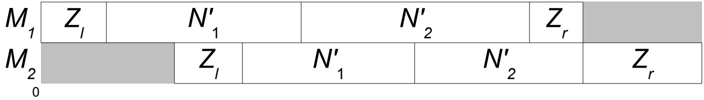


- Jeżeli $p_1 - p_{1l} \geq p_2 - p_{2r}$ ($p_1 - p_{1l} < p_2 - p_{2r}$), to "dosuń" operacje z $N_1' \cup \{Z_l\}$ na $M_2$ w prawo (z $N_2' \cup \{Z_r\}$ na $M_1$ w lewo)

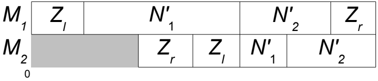


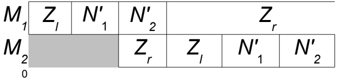

- Zmień kolejność wykonywania operacji na procesorze $M_2$ tak, aby operacja $O_{2r}$ była wykonywana jako pierwsza - możemy otrzymać jedną z dwóch sytuacji ($p_{1r} \leq p_1 - p_{2r}$; $p_{1r} > p_1 - p_{2r}$)

- Jeżeli $p_1 - p_{1l} \geq p_2 - p_{2r}$ ($p_1 - p_{1l} < p_2 - p_{2r}$), to "dosuń" operacje z $N_1' \cup \{Z_l\}$ na $M_2$ w prawo (z $N_2' \cup \{Z_r\}$ na $M_1$ w lewo)




- Zmień kolejność wykonywania operacji na procesorze $M_2$ tak, aby operacja $O_{2r}$ była wykonywana jako pierwsza - możemy otrzymać jedną z dwóch sytuacji ($p_{1r} \leq p_1 - p_{2r}$; $p_{1r} > p_1 - p_{2r}$)



**Przykład.** Algorytm Gonzalez–Sahni, $m$=2, $n$=5.

| | $Z_1$ | $Z_2$ | $Z_3$ | $Z_4$ | $Z_5$ |
|---|---|---|---|---|---|
| $M_1$ | 4 | $\boxed{1}$ | 4 | $\boxed{5}$ | 3 |
| $M_2$ | $\boxed{2}$ | 3 | $\boxed{4}$ | 6 | $\boxed{2}$ |
| | $N_2$ | $N_1$ | $N_2$ | $N_1$ | $N_2$ |

$N_1$: $Z_2, Z_4$

$N_2$: $Z_1, Z_3, Z_5$

$Z_r := Z_1$

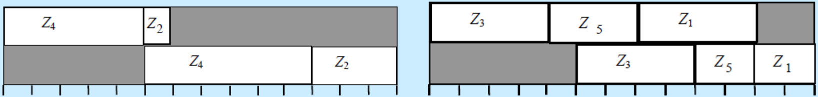
$Z_l := Z_4$

$p_{1r} \geq \max_{Z_j \in N2} p_{2j} = 4$

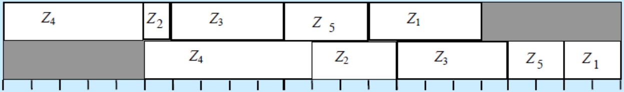
$p_{2l} \geq \max_{Z_j \in N1} p_{1j} = 5$

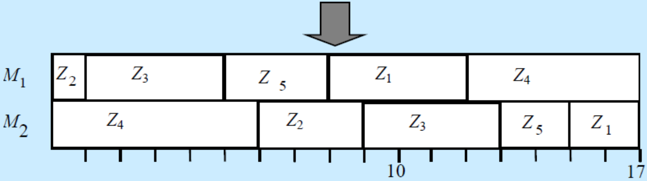
$N_1'$: $Z_2$

$N_2'$: $Z_3, Z_5$




Scalanie:



## Algorytm wielomianowy

1. Budujemy graf dwudzielny $G(V_1, V_2, E) = G(Z, M, E)$: wierzchołki jednej partycji to zadania, a drugiej to procesory; każdej niepustej operacji $O_{ij}$ odpowiada krawędź $\{Z_j, M_i\}$.

**Algorytm wielomianowy**

1. Budujemy graf dwudzielny $G(V_1, V_2, E) = G(Z, M, E)$: wierzchołki jednej partycji to zadania, a drugiej to procesory; każdej niepustej operacji $O_{ij}$ odpowiada krawędź $\{Z_j, M_i\}$.
2. Kolorujemy krawędzie $\Delta(G)$ kolorami - kolory to jednostki czasu przydzielone operacjom (poprawny harmonogram $\equiv$ poprawne pokolorowanie)

## Algorytm wielomianowy

1. Budujemy graf dwudzielny $G(V_1, V_2, E) = G(Z, M, E)$: wierzchołki jednej partycji to zadania, a drugiej to procesory; każdej niepustej operacji $O_{ij}$ odpowiada krawędź $\{Z_j, M_i\}$.
2. Kolorujemy krawędzie $\Delta(G)$ kolorami - kolory to jednostki czasu przydzielone operacjom (poprawny harmonogram $\equiv$ poprawne pokolorowanie)
3. $C^*_{max} = \Delta(G) = \max\{\max_i \sum_{j=1}^{n} p_{ij}, \max_j \sum_{i=1}^{m} p_{ij}\}$

**Algorytm Gonzalez Sahni - pseudowielomianowy, wzgl. l. krawędzi**

Slajdy dodatkowe.

## $J2||C_{\max}$; $O(n \log n)$

Algorytm oparty na alg. Johnsona.
$J_i$ - zbiór zadań składających się tylko z jednej operacji do wykonania na $M_i$
$J_{hi}$ - zbiór zadań składających się z dwóch operacji, z których pierwsza ma być wykonana na $M_h$, a druga na $M_i$ $(hi = 12, 21)$

# System ogólny - ang. *job shop*

## $J2||C_{\max}$; $O(n \log n)$

Algorytm oparty na alg. Johnsona.
$J_i$ - zbiór zadań składających się tylko z jednej operacji do wykonania na $M_i$
$J_{hi}$ - zbiór zadań składających się z dwóch operacji, z których pierwsza ma być wykonana na $M_h$, a druga na $M_i$ ($hi = 12, 21$)

- Uporządkuj zbiory $J_{hi}$ zgodnie z alg. Johnsona, a zbiory $J_i$ - dowolnie.

# System ogólny - ang. *job shop*

## $J2||C_{\max}$; $O(n \log n)$

Algorytm oparty na alg. Johnsona.
$J_i$ - zbiór zadań składających się tylko z jednej operacji do wykonania na $M_i$
$J_{hi}$ - zbiór zadań składających się z dwóch operacji, z których pierwsza ma być wykonana na $M_h$, a druga na $M_i$ ($hi = 12, 21$)

- Uporządkuj zbiory $J_{hi}$ zgodnie z alg. Johnsona, a zbiory $J_i$ - dowolnie.
- Przydziel do procesora $M_1$ zadania w kolejności $J_{12}, J_1, J_{21}$, a do procesora $M_2$ w kolejności $J_{21}, J_2, J_{12}$.