Szeregowanie zadań

- Operacje – pogrupowane w Zadania
- Procesory
- Dodatkowe zasoby

Ograniczenia

- żadna operacja nie może być jednocześnie wykonywana na więcej niż jednym procesorze
- żaden procesor nie może jednocześnie wykonywać więcej niż jedną operację
- inne – np. kolejnościowe, czasowe itd.

Klasyczny problem szeregowania

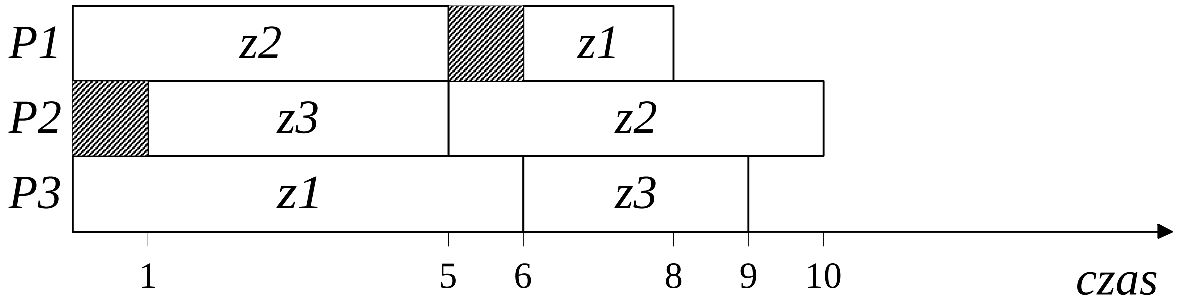
Należy ustalić kolejność wykonywania zadań na poszczególnych procesorach (oraz ewentualnie przydział dodatkowych zasobów), tak aby zachowując ograniczenia optymalizować zadane kryterium uszeregowania

Typowe parametry zadania (operacji)

- p_{lj} – czas wykonywania na procesorze l
- r_j – termin gotowości do wykonania
- d_j – żądany termin ukończenia
- w_j – priorytet (waga)
- ograniczenia kolejnościowe
- możliwość przerywania (podzielność)
- wymagania zasobowe

Uszeregowanie (harmonogram)

Prezentacja graficzna – wykres Gantta



Parametry oceny uszeregowania

- C_j – termin (moment) ukończenia
- F_j – czas przepływu $F_j = C_j - r_j$
- L_j – nieterminowość $L_j = C_j - d_j$
- T_j – spóźnienie $T_j = \max\{C_j - d_j, 0\}$
- U_j – czy zadanie (operacja) spóźnione

Typowe kryteria uszeregowania
(minimalizowane)

- $C_{\max}$ – długość uszeregowania $C_{\max} = \max\{C_j\}$
- $L_{\max}$ – maksymalna nieterminowość $L_{\max} = \max\{L_j\}$
- $\overline{F}$ – średni czas przepływu ($1/n\ \Sigma F_j$)
- $\overline{F}_w$ – średni ważony czas przepływu ($\Sigma w_j F_j / \Sigma w_j$)
- $\Sigma\ T_j$ – sumaryczne (średnie) spóźnienie
- $\Sigma\ w_j T_j$ – sumaryczne (średnie) ważone spóźnienie
- $\Sigma\ U_j$ – liczba zadań spóźnionych
- $\Sigma\ w_j U_j$ – sumaryczne kary za spóźnienie

Klasyfikacja problemów szeregowania

- szeregowanie na jednym procesorze
- szeregowanie na procesorach równoległych
  - jednakowych (P)
  - jednorodnych (Q)
  - różnych (R)
- szeregowanie w systemie przepływowym
  - permutacyjnym (PF)
  - ogólnym (F)
- szeregowanie w systemie gniazdowym (J)
- szeregowanie w systemie otwartym (O)

Notacja

α | β | γ

- α – rodzaj systemu i liczba procesorów,
  - ∗ rodzaj systemu np., *P*, *Q*, *R*, *F*, *PF*, *J*, *O*
  - ∗ drugie pole puste gdy liczba procesorów nie jest ustalona
- β – charakterystyka zadań
  - ∗ gdy pole puste – zadania niepodzielne, niezależne, z zerowymi (jednakowymi) terminami gotowości, bez dodatkowych wymagań zasobowych
  - ∗ *pmtn* – zadania podzielne
  - ∗ *prec* – występują ograniczenia kolejnościowe (zadania zależne)
  - ∗ r_j – różne terminy gotowości zadań
  - ∗ *res* – występują wymagania na dodatkowe zasoby
- γ – rodzaj kryterium szeregowania, np.

$C_{\max}$, $L_{\max}$, ΣC_j, $\Sigma w_j C_j$, ΣT_j, $\Sigma w_j T_j$, ΣU_j, $\Sigma w_j U_j$

Szeregowanie na jednym procesorze

- $1 \mid r_j \mid C_{\max}$ – szeregowanie według niemalejących r_j
- $1 \mid \mid L_{\max}$ – reguła EDD (*Earliest Due Date*)
- $1 \mid r_j \mid L_{\max}$ – problem NP-trudny
- $1 \mid \mid \Sigma\, C_j$ – reguła SPT (*Shortest Processing Time*)
- $1 \mid \mid \Sigma\, w_jC_j$ – szeregowanie według niemalejących współczynników p_j / w_j
- $1 \mid \mid \Sigma\, T_j$ – problem NP-trudny
- $1 \mid \mid \Sigma\, U_j$ – algorytm Moore'a (Hodgsona)
    1. uszereguj zadania według niemalejących d_j;
    2. znajdź pierwsze zadanie j, które jest opóźnione. Jeżeli nie ma zadań opóźnionych to STOP;
    3. spośród uszeregowanych zadań 1,...,j usuń zadanie z najdłuższym czasem wykonywania p_j;

  Usunięte zadania umieść na końcu uszeregowania.
- $1 \mid \mid \Sigma\, w_jU_j$ – problem NP-trudny
- problem z przezbrojeniami – NP-trudny

Szeregowanie na procesorach równoległych

- $P2 \mid\mid C_{max}$ – problem NP-trudny
- $P \mid pmtn \mid C_{max}$ – przy m procesorach

  $C^{*}_{max} = \max\{ \max\{p_j\}, 1/m \sum p_j \}$
- $R \mid pmtn \mid C_{max}$ – metoda dwufazowa
- $P \mid\mid \Sigma C_j$ – uogólniona reguła SPT
- $R \mid\mid \Sigma C_j$ – model sieciowy
- $P2 \mid\mid \Sigma w_j C_j$ – problem NP-trudny

Szeregowanie czasooptymalne zadań podzielnych na procesorach równoległych

Problem $R \mid pmtn \mid C_{\max}$

Metoda dwufazowa

- faza pierwsza – wyznaczenie minimalnej długości uszeregowania i przydział zadań do procesorów

t_{lj} – zmienna decyzyjna oznaczająca sumaryczny czas realizacji zadania j na procesorze l

$$\min C_{\max}$$

przy ograniczeniach

$$\sum_j t_{lj} \le C_{\max} \qquad \forall\ l$$

$$\sum_l t_{lj} \le C_{\max} \qquad \forall\ j$$

$$\sum_l \frac{1}{p_{lj}} t_{lj} = 1 \qquad \forall\ j$$

$$t_{lj} \ge 0 \qquad \forall\ l, j$$

- faza druga – określenie uszeregowania

problem $O \mid pmtn \mid C_{\max}$

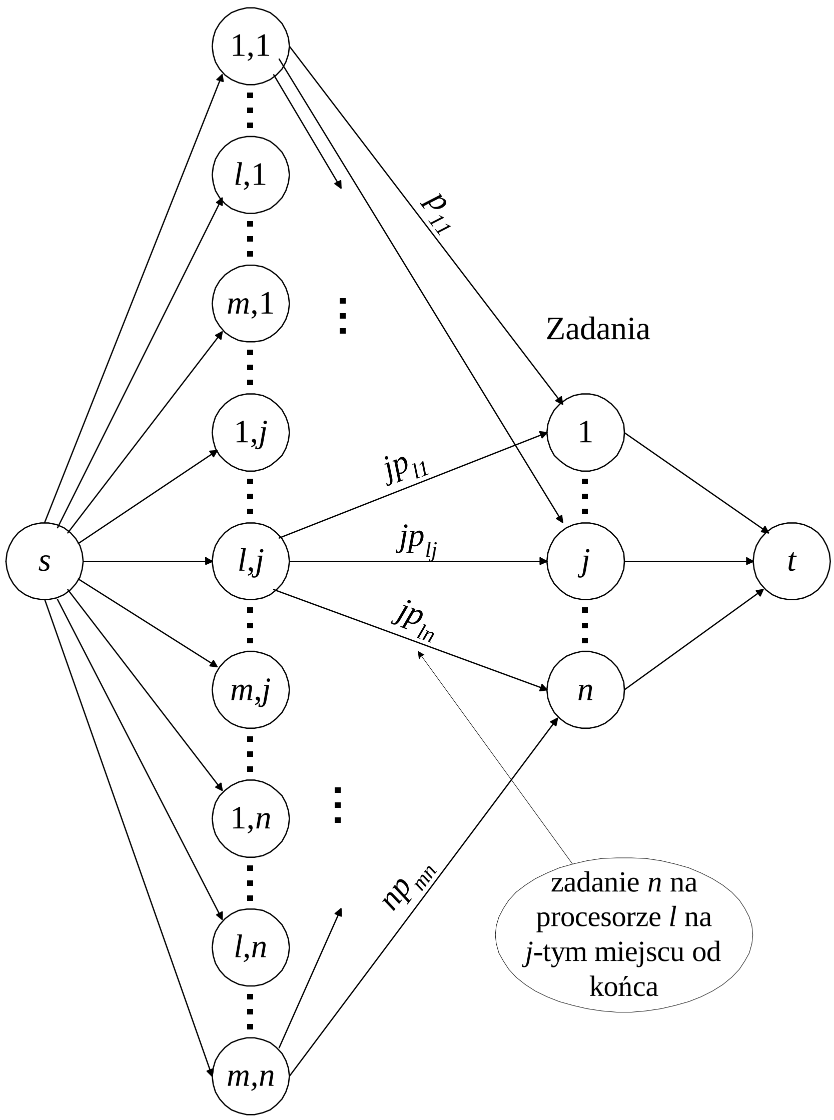
model sieciowy (w przypadku całkowitych czasów wykonywania operacji – model kolorowania krawędzi multigrafu dwudzielnego)

Szeregowanie zadań na procesorach równoległych z kryterium ΣC_j

Problem $R \mid \mid \Sigma C_j$

Model sieciowy

n zadań, m procesorów równoległych



Przepustowości wszystkich łuków [0, 1]

Wartość przepływu od s do t równa n

Problem przepływowy (flow shop)

(minimalizacja $C_{\max}$)

- permutacyjny – wszystkie procesory wykonują zadania w tej samej kolejności
- ogólny – każdy z procesorów może wykonywać zadania w dowolnej kolejności

Istnieje rozwiązanie optymalne ogólnego problemu przepływowego, w którym

- kolejność wykonywania zadań na pierwszych dwóch procesorach jest taka sama
- kolejność wykonywania zadań na ostatnich dwóch procesorach jest taka sama

Wniosek

Przy liczbie procesorów $m \le 3$ rozwiązań optymalnych ogólnego problemu przepływowego wystarczy poszukiwać wśród uszeregowań permutacyjnych.

- $F2 \,|\,|\, C_{\max}$ – algorytm Johnsona
- $F3 \,|\,|\, C_{\max}$ – problem NP-trudny

Problem przepływowy, 2 procesory

$F2||C_{\max}$

Algorytm Johnsona

1. podziel zadania na dwa zbiory
   - $S1 \leftarrow \{ j : p_{1j} \leq p_{2j} \}$
   - $S2 \leftarrow \{ j : p_{1j} > p_{2j} \}$
2. wykonaj najpierw zadania ze zbioru *S1* w kolejności niemalejących wartości p_{1j}
3. następnie wykonaj zadania ze zbioru *S2* w kolejności nierosnących wartości p_{2j}

Kolejność wykonywania zadań na obu procesorach taka sama

Szeregowanie w systemie otwartym

- $O2 \,|\,|\, C_{\max}$ – algorytm Gonzaleza i Sahni'ego
- $O3 \,|\,|\, C_{\max}$ – problem NP-trudny
- $O \,|\, pmtn \,|\, C_{\max}$

$$C^{*}_{\max} = \max\{\max_{l}\{\sum_{j} p_{lj}\}, \max_{j}\{\sum_{l} p_{lj}\}\}$$

model sieciowy (w przypadku całkowitych czasów wykonywania operacji – model kolorowania krawędzi multigrafu dwudzielnego)

Dynamiczne reguły szeregowania

- **FIFO** (*First-In First-Out*), **FCFS** (*First-Come First-Served*) – według kolejności pojawiania się zadań

- **SPT** (*Shortest Processing Time*) – najpierw wykonywane najkrótsze zadania

Reguła korzystna w przypadku:

- minimalizacji średniego czasu przepływu
- minimalizacji średniej liczby zadań przebywających w systemie
- minimalizacji średniego spóźnienia
- minimalizacji liczby zadań spóźnionych

Niedogodności:

- blokuje zadania o długich czasach wykonywania

- **LPT** (*Longest Processing Time*) – najpierw wykonywane najdłuższe zadania

Reguła korzystna w przypadku:

- szeregowania czasooptymalnego na procesorach równoległych, np. dla problemu $Pm \mid\mid C_{\max}$

$$C_{\max}(LPT) \leq \left(\frac{4}{3} - \frac{1}{3m}\right) C_{\max}^{*}$$

Dynamiczne reguły szeregowania
dla zadań z żądanymi terminami ukończenia

- **EDD** (*Earliest Due Date*) – najpierw zadania z najwcześniejszym żądanym terminem ukończenia

- **MOD** – szeregowanie według zmodyfikowanego terminu zakończenia $MOD = \max\{ d_j, t + P_j(t) \}$, gdzie

  * t – aktualna chwila

  * $P_j(t)$ – suma czasów realizacji pozostałych do wykonania operacji zadania

- **CR** (*Critical Ratio*) – szeregowanie według wskaźnika

$$CR = \frac{d_j - t}{P_j(t)}$$

- **STO** (*Slack Time per Operation*) – szeregowanie według wskaźnika

$$STO = \frac{d_j - t - P_j(t)}{NOP}$$ gdzie

NOP – liczba operacji zadania pozostałych do wykonania