Szeregowanie zadań

Przedmiot fakultatywny 15h wykładu + 15h ćwiczeń

dr Hanna Furmańczyk

25 lutego 2020

Zasady zaliczenia

1. ćwiczenia (ocena):
   - kolokwium,
   - zadania dodatkowe (implementacje algorytmów),
   - praca na ćwiczeniach.
2. Wykład (zal):
   - zaliczone ćwiczenia,
   - zadanie z wykładu.

Motywacja

Szeregowanie zadań:

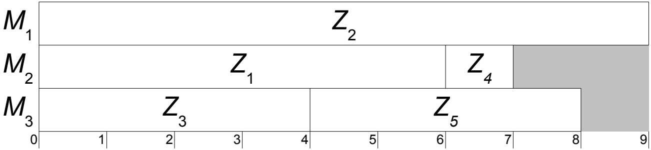
- część wielozadaniowego systemu operacyjnego, odpowiedzialna za ustalanie kolejności dostępu zadań do procesora [jak rozdzielić czas procesora i dostęp do innych zasobów pomiędzy zadania, które w praktyce zwykle o te zasoby konkurują]
- serwery baz danych,
- organizacja obliczeń rozproszonych,
- linie produkcyjne,
- plany zajeć szkolnych, konferencji, itp.
- planowanie projektu,
- organizacja pracy.

Historia

- linia produkcyjna Henry'ego Forda (pierwsze lata XX w.),
- algorytm Jacksona - 1955 (również dla produkcji przemysłowej),
- ...

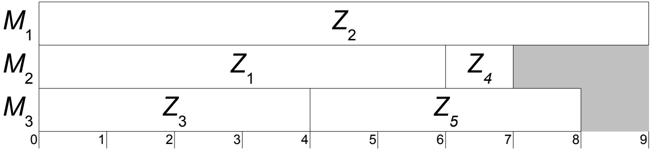
Przykłady

1. Pięć zadań o czasach wykonania $[p_1, \ldots, p_5] = [6, 9, 4, 1, 4]$ należy uszeregować na trzech identycznych maszynach tak, by zakończyły się one możliwie jak najszybciej.



Przykłady

1. Pięć zadań o czasach wykonania $[p_1, \ldots, p_5] = [6, 9, 4, 1, 4]$ należy uszeregować na trzech identycznych maszynach tak, by zakończyły się one możliwie jak najszybciej.



Czy ten harmonogram jest poprawny?

- żadne zadanie nie może być jednocześnie wykonywane przez różne maszyny,

Zasady poprawności harmonogramu - wstęp

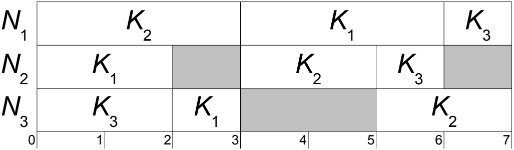
- żadne zadanie nie może być jednocześnie wykonywane przez różne maszyny,
- żaden procesor nie pracuje równocześnie nad różnymi zadaniami,

Zasady poprawności harmonogramu - wstęp

- żadne zadanie nie może być jednocześnie wykonywane przez różne maszyny,
- żaden procesor nie pracuje równocześnie nad różnymi zadaniami,
- ciąg dalszy nastąpi.

2 Jednodniowy plan zajęć (K_i - klasy, N_j - nauczyciele)

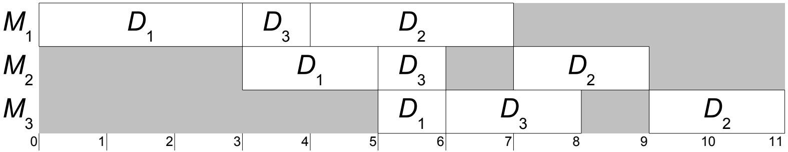
	N_1	N_2	N_3
K_1	3	2	1
K_2	3	2	2
K_3	1	1	2



Procesory dedykowane - system otwarty (kolejność operacji dowolna).

3 Taśma produkcyjna (ważna kolejność operacji)

	M_1	M_2	M_3
D_1	3	2	1
D_2	3	2	2
D_3	1	1	2



Procesory dedykowane - system przepływowy (kolejność operacji musi być zgodna z numeracją maszyn).

Dziedzina ta zajmuje się *szeregowaniem* (układaniem harmonogramów) *zadań* (programów, czynności, prac) na *maszynach* (procesorach, obrabiarkach, stanowiskach obsługi).

Dziedzina ta zajmuje się *szeregowaniem* (układaniem harmonogramów) *zadań* (programów, czynności, prac) na *maszynach* (procesorach, obrabiarkach, stanowiskach obsługi). Szukamy harmonogramu wykonania dla danego zbioru zadań w określonych warunkach, tak by zminimalizować przyjęte *kryterium oceny (koszt) uszeregowania.*

Dziedzina ta zajmuje się *szeregowaniem* (układaniem harmonogramów) *zadań* (programów, czynności, prac) na *maszynach* (procesorach, obrabiarkach, stanowiskach obsługi). Szukamy harmonogramu wykonania dla danego zbioru zadań w określonych warunkach, tak by zminimalizować przyjęte *kryterium oceny (koszt) uszeregowania.*

Model deterministyczny: parametry systemu i zadań są od początku znane.

Sposoby obsługi zadań

1. Procesory równoległe (każdy procesor może obsłużyć każde zadanie):

Sposoby obsługi zadań

1. Procesory równoległe (każdy procesor może obsłużyć każde zadanie):
   - procesory identyczne - wszystkie są jednakowo szybkie,
   - procesory jednorodne - mają różne szybkości, ale stosunki czasów wykonania zadań są niezależne od maszyn,
   - procesory dowolne - prędkości zależą od wykonywanych zadań.

2 Procesory dedykowane

2 Procesory dedykowane

- zadania są podzielone na operacje (zadanie Z_j składa się z operacji O_{ij} do wykonania na maszynach M_i, o długościach czasowych p_{ij}); zadanie kończy się wraz z wykonaniem swej najpóźniejszej operacji,

2 Procesory dedykowane

- zadania są podzielone na operacje (zadanie Z_j składa się z operacji O_{ij} do wykonania na maszynach M_i, o długościach czasowych p_{ij}); zadanie kończy się wraz z wykonaniem swej najpóźniejszej operacji,
- dopuszcza się sytuacje, gdy zadanie nie wykorzystuje wszystkich maszyn (operacje puste),

2 Procesory dedykowane

- zadania są podzielone na operacje (zadanie Z_j składa się z operacji O_{ij} do wykonania na maszynach M_i, o długościach czasowych p_{ij}); zadanie kończy się wraz z wykonaniem swej najpóźniejszej operacji,
- dopuszcza się sytuacje, gdy zadanie nie wykorzystuje wszystkich maszyn (operacje puste),
- żadne dwie operacje tego samego zadania nie mogą wykonywać się rownocześnie,

2 Procesory dedykowane

- zadania są podzielone na operacje (zadanie Z_j składa się z operacji O_{ij} do wykonania na maszynach M_i, o długościach czasowych p_{ij}); zadanie kończy się wraz z wykonaniem swej najpóźniejszej operacji,
- dopuszcza się sytuacje, gdy zadanie nie wykorzystuje wszystkich maszyn (operacje puste),
- żadne dwie operacje tego samego zadania nie mogą wykonywać się rownocześnie,
- żaden procesor nie może rownocześnie pracować nad różnymi operacjami.

2 Procesory dedykowane

- zadania są podzielone na operacje (zadanie Z_j składa się z operacji O_{ij} do wykonania na maszynach M_i, o długościach czasowych p_{ij}); zadanie kończy się wraz z wykonaniem swej najpóźniejszej operacji,
- dopuszcza się sytuacje, gdy zadanie nie wykorzystuje wszystkich maszyn (operacje puste),
- żadne dwie operacje tego samego zadania nie mogą wykonywać się rownocześnie,
- żaden procesor nie może rownocześnie pracować nad różnymi operacjami.

Przykład 2 i 3.

Procesory dedykowane cd.

Trzy główne typy systemów obsługi dla maszyn dedykowanych:

- system przepływowy (ang. *flow shop) - operacje każdego zadania są wykonywane przez procesory w tej samej kolejności wyznaczonej przez numery maszyn (przykład 3),*
- *system otwarty (ang. open shop) - kolejność wykonania operacji w obrębie zadań jest dowolna (przykład 2),*
- *system gniazdowy (ang. job shop) - dla każdego zadania mamy dane przyporządkowanie maszyn operacjom oraz wymaganą kolejność.*

Parametry zadań

Dane:

n zadań $Z = \{Z_1, \ldots, Z_n\}$; m maszyn (procesorów) $\{M_1, \ldots, M_m\}$.

Parametry zadań

Dane:

n zadań $Z = \{Z_1, \ldots, Z_n\}$; m maszyn (procesorów) $\{M_1, \ldots, M_m\}$.

- *Czas wykonywania zadania Z_j*
  - Dla procesorów identycznych jest on niezależny od maszyny i wynosi p_j.

Parametry zadań

Dane:

n zadań $Z = \{Z_1, \ldots, Z_n\}$; m maszyn (procesorów) $\{M_1, \ldots, M_m\}$.

- *Czas wykonywania zadania Z_j*
  - Dla procesorów identycznych jest on niezależny od maszyny i wynosi p_j.
  - Procesory jednorodne M_i charakteryzują się współczynnikami szybkości b_i, wtedy czas dla M_i to p_j/b_i.

Parametry zadań

Dane:

n zadań $Z = \{Z_1, \ldots, Z_n\}$; m maszyn (procesorów) $\{M_1, \ldots, M_m\}$.

- *Czas wykonywania zadania Z_j*
  - Dla procesorów identycznych jest on niezależny od maszyny i wynosi p_j.
  - Procesory jednorodne M_i charakteryzują się współczynnikami szybkości b_i, wtedy czas dla M_i to p_j/b_i.
  - Dla maszyn dowolnych mamy czasy p_{ij} zależne od zadań i procesorów.

Parametry zadań cd.

- *Moment przybycia zadania Z_j: r_j* (ang. *release time*). Czas, od którego zadanie może zostać podjęte. Wartość domyślna - zero.

Parametry zadań cd.

- *Termin zakończenia zadania Z_j: d_j.*
  Opcjonalny parametr. Występuje w dwóch wariantach. Może oznaczać czas, od którego nalicza się spóźnienie (ang. *due date*), lub termin, którego przekroczyć nie wolno (ang. *deadline*).

Parametry zadań cd.

- *Waga zadania Z_j: w_j.*
  Opcjonalny parametr, określający ważność zadania przy naliczaniu kosztu harmonogramu. Domyślnie zadania są jednakowej wagi i wtedy $w_j = 1$.

Parametry zadań cd.

- *Moment przybycia zadania Z_j: r_j* (ang. *release time*).
  Czas, od którego zadanie może zostać podjęte. Wartość domyślna - zero.
- *Termin zakończenia zadania Z_j: d_j.*
  Opcjonalny parametr. Występuje w dwóch wariantach. Może oznaczać czas, od którego nalicza się spóźnienie (ang. *due date*), lub termin, którego przekroczyć nie wolno (ang. *deadline*).
- *Waga zadania Z_j: w_j.*
  Opcjonalny parametr, określający ważność zadania przy naliczaniu kosztu harmonogramu. Domyślnie zadania są jednakowej wagi i wtedy $w_j = 1$.

Zadania zależne

Relacja częściowego porządku

W zbiorze zadań Z można wprowadzić ograniczenia kolejnościowe w postaci dowolnej relacji częściowego porządku. Wówczas $Z_i \prec Z_j$ oznacza, że zadanie Z_j może się zacząć wykonywać dopiero po zakończeniu Z_i

Zadania zależne

Relacja częściowego porządku

W zbiorze zadań Z można wprowadzić ograniczenia kolejnościowe w postaci dowolnej relacji częściowego porządku. Wówczas $Z_i \prec Z_j$ oznacza, że zadanie Z_j może się zacząć wykonywać dopiero po zakończeniu Z_i (czemu?

Relacja częściowego porządku

W zbiorze zadań Z można wprowadzić ograniczenia kolejnościowe w postaci dowolnej relacji częściowego porządku. Wówczas $Z_i \prec Z_j$ oznacza, że zadanie Z_j może się zacząć wykonywać dopiero po zakończeniu Z_i (czemu? np. Z_j korzysta z wyników pracy Z_i).

Zadania zależne

Relacja częściowego porządku

W zbiorze zadań Z można wprowadzić ograniczenia kolejnościowe w postaci dowolnej relacji częściowego porządku. Wówczas $Z_i \prec Z_j$ oznacza, że zadanie Z_j może się zacząć wykonywać dopiero po zakończeniu Z_i (czemu? np. Z_j korzysta z wyników pracy Z_i).

Jeśli ograniczenia te nie występują, mówimy o zadaniach *niezależnych* (tak się przyjmuje domyślnie), w przeciwnym razie są one *zależne*.

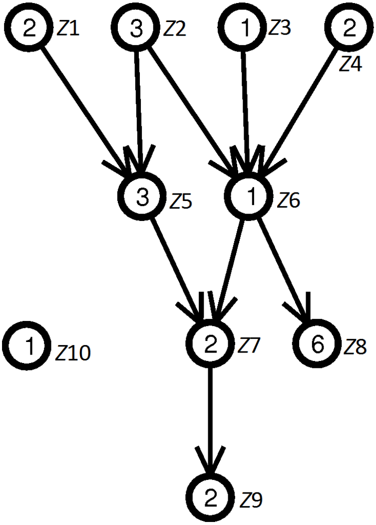
Zadania zależne

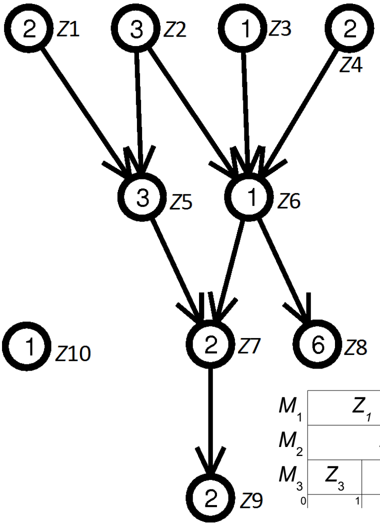
Relacja częściowego porządku

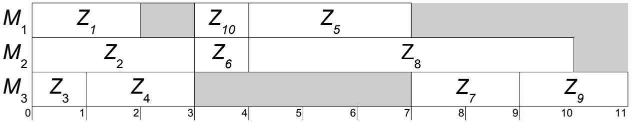
W zbiorze zadań Z można wprowadzić ograniczenia kolejnościowe w postaci dowolnej relacji częściowego porządku. Wówczas $Z_i \prec Z_j$ oznacza, że zadanie Z_j może się zacząć wykonywać dopiero po zakończeniu Z_i (czemu? np. Z_j korzysta z wyników pracy Z_i).

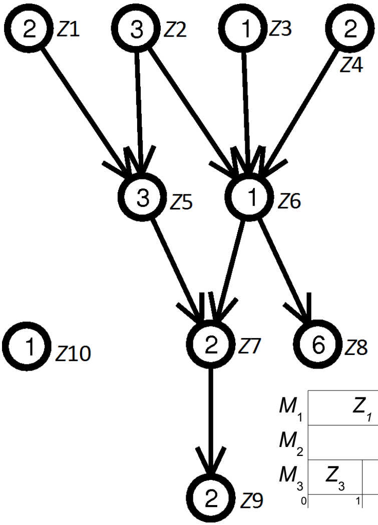
Jeśli ograniczenia te nie występują, mówimy o zadaniach *niezależnych* (tak się przyjmuje domyślnie), w przeciwnym razie są one *zależne*.

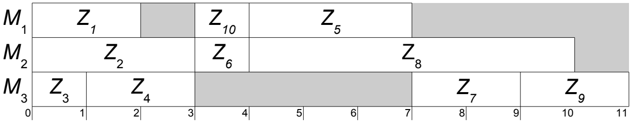
acykliczny digraf (diagram Hassego)


2
Z1
3
Z2
1
Z3
2
Z4
3
Z5
1
Z6
1
Z10
2
Z7
6
Z8
2
Z9

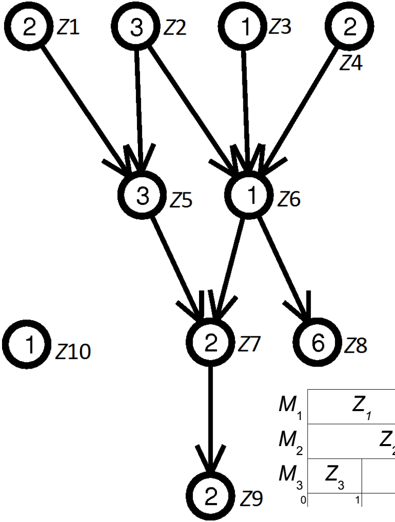

2 Z1
3 Z2
1 Z3
2 Z4
3 Z5
1 Z6
1 Z10
2 Z7
6 Z8
2 Z9

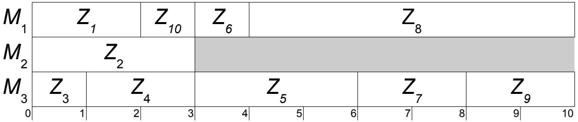

M_1 Z_1 Z_{10} Z_5
M_2 Z_2 Z_6 Z_8
M_3 Z_3 Z_4 Z_7 Z_9
0 1 2 3 4 5 6 7 8 9 10 11

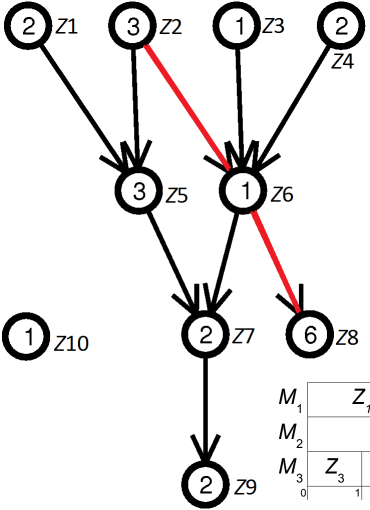


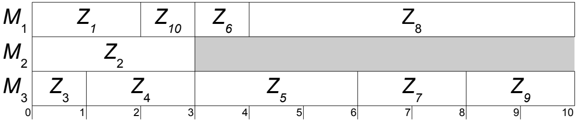


To nie jest uszeregowanie optymalne.


2 Z1
3 Z2
1 Z3
2 Z4
3 Z5
1 Z6
1 Z10
2 Z7
6 Z8
2 Z9


M_1 Z_1 Z_{10} Z_6 Z_8
M_2 Z_2
M_3 Z_3 Z_4 Z_5 Z_7 Z_9
0 1 2 3 4 5 6 7 8 9 10





To jest uszeregowanie optymalne (ścieżka krytyczna).

Parametry zadań cd.

Zadania mogą być:

- *niepodzielne* - przerwy w wykonaniu są niedopuszczalne (domyślnie),

Parametry zadań cd.

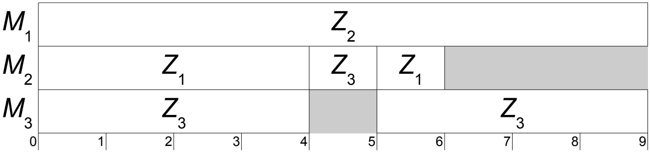
Zadania mogą być:

- *niepodzielne* - przerwy w wykonaniu są niedopuszczalne (domyślnie),
- *podzielne* - wykonanie można przerwać i podjąć ponownie, w przypadku maszyn równoległych nawet na innym procesorze.

Parametry zadań cd.

Zadania mogą być:

- *niepodzielne* - przerwy w wykonaniu są niedopuszczalne (domyślnie),
- *podzielne* - wykonanie można przerwać i podjąć ponownie, w przypadku maszyn równoległych nawet na innym procesorze.



Zasady poprawności harmonogramu (już w całości):

- w każdej chwili procesor może wykonywać co najwyżej jedno zadanie,

Zasady poprawności harmonogramu (już w całości):

- w każdej chwili procesor może wykonywać co najwyżej jedno zadanie,
- w każdej chwili zadanie może być obsługiwane przez co najwyżej jeden procesor,

Zasady poprawności harmonogramu (już w całości):

- w każdej chwili procesor może wykonywać co najwyżej jedno zadanie,
- w każdej chwili zadanie może być obsługiwane przez co najwyżej jeden procesor,
- zadanie Z_j wykonuje się w całości w przedziale czasu $[r_j, \infty)$,

Zasady poprawności harmonogramu (już w całości):

- w każdej chwili procesor może wykonywać co najwyżej jedno zadanie,
- w każdej chwili zadanie może być obsługiwane przez co najwyżej jeden procesor,
- zadanie Z_j wykonuje się w całości w przedziale czasu $[r_j, \infty)$,
- spełnione są ograniczenia kolejnościowe,

Zasady poprawności harmonogramu (już w całości):

- w każdej chwili procesor może wykonywać co najwyżej jedno zadanie,
- w każdej chwili zadanie może być obsługiwane przez co najwyżej jeden procesor,
- zadanie Z_j wykonuje się w całości w przedziale czasu $[r_j, \infty)$,
- spełnione są ograniczenia kolejnościowe,
- w przypadku zadań niepodzielnych każde zadanie wykonuje się nieprzerwanie w pewnym domknięto-otwartym przedziale czasowym, dla zadań podzielnych czasy wykonania tworzą skończoną sumę rozłącznych przedziałów.

Kryteria kosztu harmonogramu

Dla uszeregowanego zadania Z_j możemy określić:

Dla uszeregowanego zadania Z_j możemy określić:

- *moment zakończenia* C_i (ang. *completion time*),

Kryteria kosztu harmonogramu

Dla uszeregowanego zadania Z_j możemy określić:

- *moment zakończenia* C_i (ang. *completion time*),
- *czas przepływu przez system* $\bar{F}_i = C_i - r_i$ (ang. *flow time*),

Dla uszeregowanego zadania Z_j możemy określić:

- *moment zakończenia* C_i (ang. *completion time*),
- *czas przepływu przez system* $\bar{F}_i = C_i - r_i$ (ang. *flow time*),
- *opóźnienie* $L_i = C_i - d_i$ (ang. *lateness*),

Kryteria kosztu harmonogramu

Dla uszeregowanego zadania Z_j możemy określić:

- *moment zakończenia* C_i (ang. *completion time*),
- *czas przepływu przez system* $\bar{F}_i = C_i - r_i$ (ang. *flow time*),
- *opóźnienie* $L_i = C_i - d_i$ (ang. *lateness*),
- *spóźnienie* $T_i = \max\{C_i - d_i, 0\}$ (ang. *tardiness*),

Kryteria kosztu harmonogramu

Dla uszeregowanego zadania Z_j możemy określić:

- *moment zakończenia* C_i (ang. *completion time*),
- *czas przepływu przez system* $\bar{F}_i = C_i - r_i$ (ang. *flow time*),
- *opóźnienie* $L_i = C_i - d_i$ (ang. *lateness*),
- *spóźnienie* $T_i = \max\{C_i - d_i, 0\}$ (ang. *tardiness*),
- *„znacznik spóxnienia"* $U_i = w(C_i > d_i)$, a więc odpowiedź (0/1 czyli Nie/Tak) na pytanie „czy zadanie się spóźniło?".

Kryteria kosztu harmonogramu

Najczęściej stosowane kryteria:

Kryteria kosztu harmonogramu

Najczęściej stosowane kryteria:

- *długość uszeregowania* $C_{\max} = \max\{C_j : j = 1, \dots, n\}$,

Kryteria kosztu harmonogramu

Najczęściej stosowane kryteria:

- *długość uszeregowania* $C_{\max} = \max\{C_j : j = 1, \ldots, n\}$,
- *całkowity (łączny) czas zakończenia zadania* $\sum C_j = \sum_{i=1}^{n} C_i$,

Kryteria kosztu harmonogramu

Najczęściej stosowane kryteria:

- *długość uszeregowania* $C_{\max} = \max\{C_j : j = 1, \ldots, n\}$,
- *całkowity (łączny) czas zakończenia zadania* $\sum C_j = \sum_{i=1}^{n} C_i$,
- *średni czas przepływu* $\bar{F} = (\sum_{i=1}^{n} \bar{F}_i)/n$,

Kryteria kosztu harmonogramu

Najczęściej stosowane kryteria:

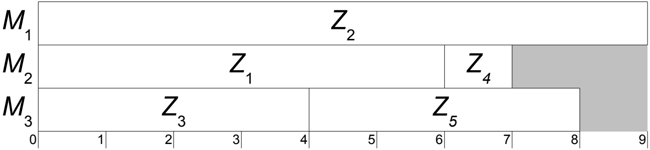
- *długość uszeregowania* $C_{\max} = \max\{C_j : j = 1, \ldots, n\}$,
- *całkowity (łączny) czas zakończenia zadania* $\sum C_j = \sum_{i=1}^{n} C_i$,
- *średni czas przepływu* $\bar{F} = (\sum_{i=1}^{n} \bar{F}_i)/n$,

Przykład: uszeregowanie na trzech maszynach równoległych zadań niezależnych, niepodzielnych.

Najczęściej stosowane kryteria:

- *długość uszeregowania* $C_{\max} = \max\{C_j : j = 1, \ldots, n\}$,
- *całkowity (łączny) czas zakończenia zadania* $\sum C_j = \sum_{i=1}^{n} C_i$,
- *średni czas przepływu* $\bar{F} = (\sum_{i=1}^{n} \bar{F}_i)/n$,

Przykład: uszeregowanie na trzech maszynach równoległych zadań niezależnych, niepodzielnych.

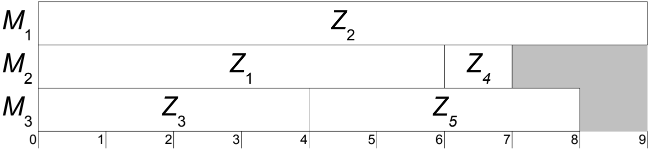


$p_1 = 6, p_2 = 9, p_3 = 4, p_4 = 1, p_5 = 4$

Najczęściej stosowane kryteria:

- *długość uszeregowania* $C_{\max} = \max\{C_j : j = 1, \ldots, n\}$,
- *całkowity (łączny) czas zakończenia zadania* $\sum C_j = \sum_{i=1}^{n} C_i$,
- *średni czas przepływu* $\bar{F} = (\sum_{i=1}^{n} \bar{F}_i)/n$,

Przykład: uszeregowanie na trzech maszynach równoległych zadań niezależnych, niepodzielnych.



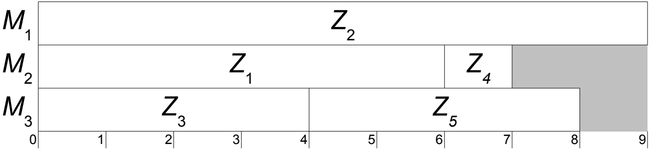
$p_1 = 6, p_2 = 9, p_3 = 4, p_4 = 1, p_5 = 4$

$$C_{\max} = 9$$

Najczęściej stosowane kryteria:

- *długość uszeregowania* $C_{\max} = \max\{C_j : j = 1, \ldots, n\}$,
- *całkowity (łączny) czas zakończenia zadania* $\sum C_j = \sum_{i=1}^{n} C_i$,
- *średni czas przepływu* $\bar{F} = (\sum_{i=1}^{n} \bar{F}_i)/n$,

Przykład: uszeregowanie na trzech maszynach równoległych zadań niezależnych, niepodzielnych.



$p_1 = 6, p_2 = 9, p_3 = 4, p_4 = 1, p_5 = 4$

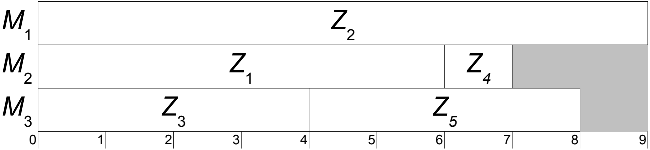
$$C_{\max} = 9$$

$$\sum C_j = 6 + 9 + 4 + 7 + 8 = 34$$

Kryteria cd.

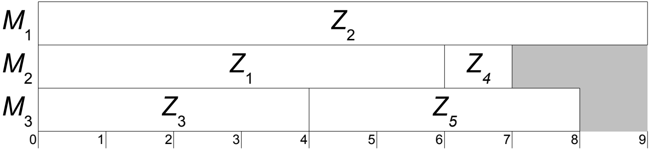
Można wprowadzać wagi (priorytety) zadań:

$w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 1, w_5 = 1$



Kryteria cd.

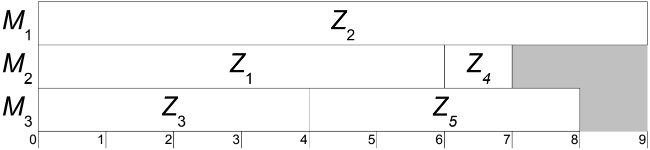
Można wprowadzać wagi (priorytety) zadań:
$w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 1, w_5 = 1$



- *całkowity ważony czas zakończenia* $\sum w_j C_j = \sum_{i=1}^{n} w_i C_i$

Kryteria cd.

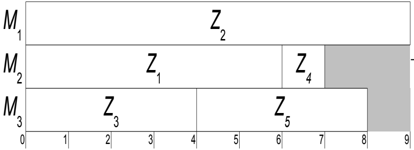
Można wprowadzać wagi (priorytety) zadań:
$w_1 = 1, w_2 = 2, w_3 = 3, w_4 = 1, w_5 = 1$



- *całkowity ważony czas zakończenia* $\sum w_j C_j = \sum_{i=1}^{n} w_i C_i$

$$\sum w_j C_j = 6 + 18 + 12 + 7 + 8 = 51$$

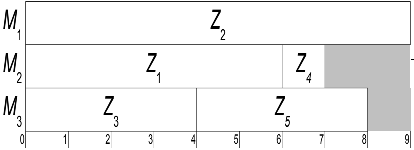
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8

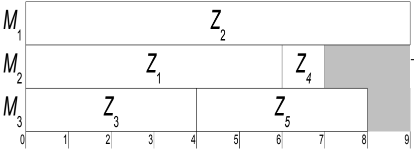
Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$



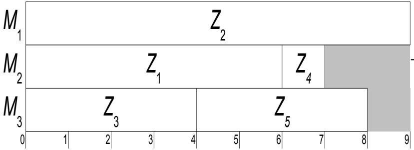
Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:					

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1				

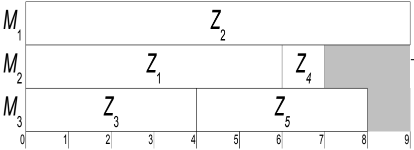
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2			

Kryteria oparte na wymaganych terminach zakończenia

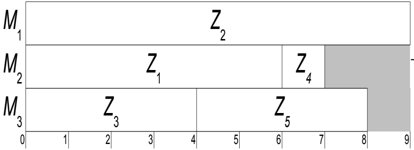
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1		

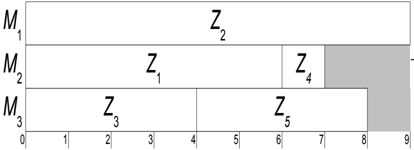
Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	

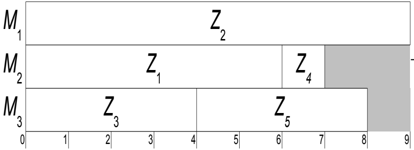
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$

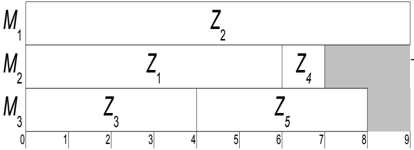


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0

$L_{\max} = 2$

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$

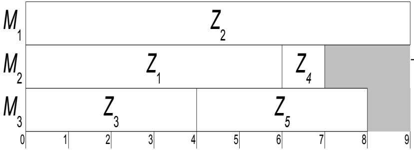


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0

$L_{\max} = 2$

Kryteria oparte na wymaganych terminach zakończenia

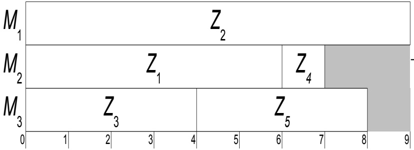
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:					

$L_{\max} = 2$

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$

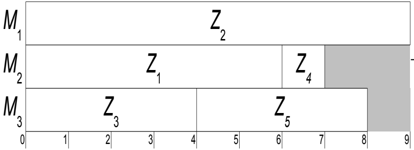


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0				

$L_{\max} = 2$

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$

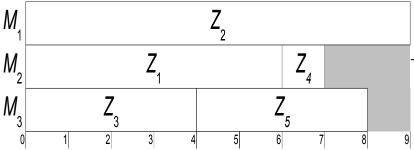


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2			

$L_{\max} = 2$

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$

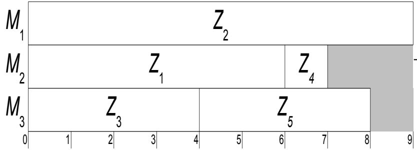


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0		

$L_{\max} = 2$

Kryteria oparte na wymaganych terminach zakończenia

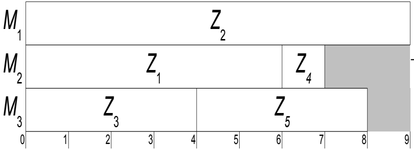
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	

$L_{\max} = 2$

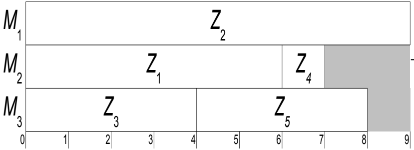
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	0

$L_{\max} = 2$

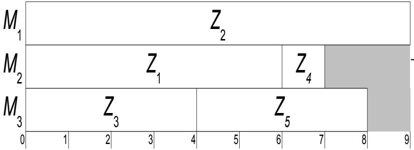
- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	0

$L_{\max} = 2$ $T_{\max} = 2$

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$
- *całkowite spóźnienie* $\sum T_j = \sum_{i=1}^{n} T_i$

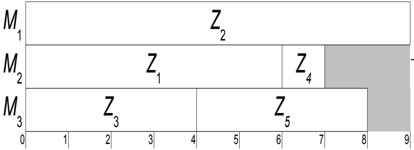


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	0

$L_{\max} = 2$ $\quad$ $T_{\max} = 2$

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \dots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \dots, n\}$
- *całkowite spóźnienie* $\sum T_j = \sum_{i=1}^{n} T_i$

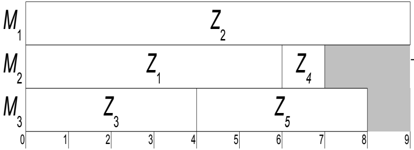


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	0

$L_{\max} = 2$ $\quad T_{\max} = 2$ $\quad \sum T_j = 4$

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$
- *całkowite spóźnienie* $\sum T_j = \sum_{i=1}^{n} T_i$
- *liczba spóźnionych zadań* $\sum U_j = \sum_{i=1}^{n} U_i$

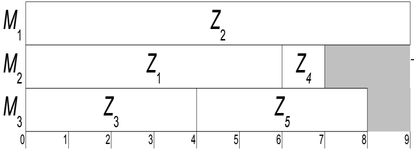


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	0

$L_{\max} = 2$ $\quad T_{\max} = 2$ $\quad \sum T_j = 4$

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$
- *całkowite spóźnienie* $\sum T_j = \sum_{i=1}^{n} T_i$
- *liczba spóźnionych zadań* $\sum U_j = \sum_{i=1}^{n} U_i$

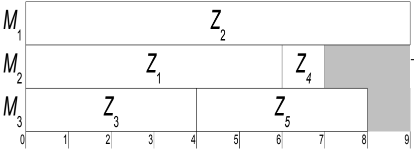


Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	0

$L_{\max} = 2$ $\quad T_{\max} = 2$ $\quad \sum T_j = 4$ $\quad \sum U_j = 2$

Kryteria oparte na wymaganych terminach zakończenia

- *maksymalne opóźnienie* $L_{\max} = \max\{L_j : j = 1, \ldots, n\}$
- *maksymalne spóźnienie* $T_{\max} = \max\{T_j : j = 1, \ldots, n\}$
- *całkowite spóźnienie* $\sum T_j = \sum_{i=1}^{n} T_i$
- *liczba spóźnionych zadań* $\sum U_j = \sum_{i=1}^{n} U_i$
- można wprowadzać wagi zadań, łączyć kryteria, np. łączne ważone spóźnienie $\sum w_j T_j = \sum_{i=1}^{n} w_i T_i$.



Zadanie:	Z_1	Z_2	Z_3	Z_4	Z_5
d_i	7	7	5	5	8
L_i:	-1	2	-1	2	0
T_i:	0	2	0	2	0

$L_{\max} = 2$ $\quad T_{\max} = 2$ $\quad \sum T_j = 4$ $\quad \sum U_j = 2$

Jak to opisać?

Jak to opisać? Notacja trójpolowa

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne
Q - proc. jednorodne
R - proc. dowolne
O - system otwarty (ang. *open shop*)

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

J - system ogólny (ang. *job shop*)

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne
Q - proc. jednorodne
R - proc. dowolne
O - system otwarty (ang. *open shop*)
F - system przepływowy (ang. *flow shop*)
J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne

pmtn - zadania podzielne (ang. *preemption*)

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne

pmtn - zadania podzielne (ang. *preemption*)

prec - zadania zależne

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne
Q - proc. jednorodne
R - proc. dowolne
O - system otwarty (ang. *open shop*)
F - system przepływowy (ang. *flow shop*)
J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne
pmtn - zadania podzielne (ang. *preemption*)
prec - zadania zależne
r_j - różne wartości momentów przybycia

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne

pmtn - zadania podzielne (ang. *preemption*)

prec - zadania zależne

r_j - różne wartości momentów przybycia

$p_j = 1$ lub *UET* - zadania jednostkowe

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne
Q - proc. jednorodne
R - proc. dowolne
O - system otwarty (ang. *open shop*)
F - system przepływowy (ang. *flow shop*)
J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne
pmtn - zadania podzielne (ang. *preemption*)
prec - zadania zależne
r_j - różne wartości momentów przybycia
$p_j = 1$ lub *UET* - zadania jednostkowe
$p_{ij} \in \{0, 1\}$ lub *ZUET* - operacje jednostkowe lub puste (procesory dedykowane)

Jak to opisać? Notacja trójpolowa

$$\alpha|\beta|\gamma$$

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne

pmtn - zadania podzielne (ang. *preemption*)

prec - zadania zależne

r_j - różne wartości momentów przybycia

$p_j = 1$ lub *UET* - zadania jednostkowe

$p_{ij} \in \{0, 1\}$ lub *ZUET* - operacje jednostkowe lub puste (procesory dedykowane)

$C_j \leq d_j$ - istnieją wymagane i nieprzekraczalne terminy zakończenia zadań

Jak to opisać? Notacja trójpolowa

$\alpha|\beta|\gamma$

γ - kryterium optymalizacji

α - środowisko maszynowe

P - procesory identyczne

Q - proc. jednorodne

R - proc. dowolne

O - system otwarty (ang. *open shop*)

F - system przepływowy (ang. *flow shop*)

J - system ogólny (ang. *job shop*)

β - charakterystyka zadań

puste: zadania są niepodzielne, niezależne, z $r_j = 0$, czasy wykonania i ewentualne wymagane terminy zakończenia d_j dowolne

pmtn - zadania podzielne (ang. *preemption*)

prec - zadania zależne

r_j - różne wartości momentów przybycia

$p_j = 1$ lub *UET* - zadania jednostkowe

$p_{ij} \in \{0, 1\}$ lub *ZUET* - operacje jednostkowe lub puste (procesory dedykowane)

$C_j \leq d_j$ - istnieją wymagane i nieprzekraczalne terminy zakończenia zadań

cd. wartości β:

no-idle - procesory musza pracować w sposób ciagły, bez okienek

cd. wartości β:
no-idle - procesory musza pracować w sposób ciagły, bez okienek
no-wait - okienka między operacjami w zadaniach są zabronione (proc. dedykowane)

cd. wartości β:

no-idle - procesory musza pracować w sposób ciagły, bez okienek

no-wait - okienka między operacjami w zadaniach są zabronione (proc. dedykowane)

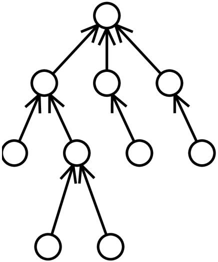
in–tree, out–tree, chains, ... – różne szczególne postaci relacji zależności kolejnościowych (prec).

cd. wartości β:

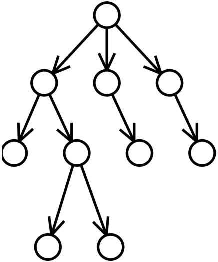
no-idle - procesory musza pracować w sposób ciagły, bez okienek

no-wait - okienka między operacjami w zadaniach są zabronione (proc. dedykowane)

in–tree, out–tree, chains, ... – różne szczególne postaci relacji zależności kolejnościowych (prec).



in-tree



out-tree

Przykłady - notacja trójpolowa

$P3|prec|C_{\max}$

Przykłady - notacja trójpolowa

$P3|prec|C_{max}$

Szeregowanie niepodzielnych zadań zależnych na trzech identycznych maszynach równoległych w celu zminimalizowania długości harmonogramu.

Przykłady - notacja trójpolowa

$P3|prec|C_{max}$

Szeregowanie niepodzielnych zadań zależnych na trzech identycznych maszynach równoległych w celu zminimalizowania długości harmonogramu.

$R|pmtn, prec, r_j|\sum U_j$

Przykłady - notacja trójpolowa

$P3|prec|C_{\max}$

Szeregowanie niepodzielnych zadań zależnych na trzech identycznych maszynach równoległych w celu zminimalizowania długości harmonogramu.

$R|pmtn, prec, r_j|\sum U_j$

Szeregowanie podzielnych zadań zależnych z różnymi czasami przybycia i terminami zakończenia na równoległych dowolnych maszynach (liczba procesorów jest częścią danych) w celu minimalizacji liczby zadań spóźnionych.

Przykłady - notacja trójpolowa

$P3|prec|C_{\max}$

Szeregowanie niepodzielnych zadań zależnych na trzech identycznych maszynach równoległych w celu zminimalizowania długości harmonogramu.

$R|pmtn, prec, r_j|\sum U_j$

Szeregowanie podzielnych zadań zależnych z różnymi czasami przybycia i terminami zakończenia na równoległych dowolnych maszynach (liczba procesorów jest częścią danych) w celu minimalizacji liczby zadań spóźnionych.

$1|r_j, C_j \leq d_j|-$

Przykłady - notacja trójpolowa

$P3|prec|C_{\max}$

Szeregowanie niepodzielnych zadań zależnych na trzech identycznych maszynach równoległych w celu zminimalizowania długości harmonogramu.

$R|pmtn, prec, r_j|\sum U_j$

Szeregowanie podzielnych zadań zależnych z różnymi czasami przybycia i terminami zakończenia na równoległych dowolnych maszynach (liczba procesorów jest częścią danych) w celu minimalizacji liczby zadań spóźnionych.

$1|r_j, C_j \leq d_j|-$

Pytanie o istnienie (brak kryterium kosztu, więc nic nie optymalizujemy!) uszeregowania zadań niepodzielnych i niezależnych o różnych momentach przybycia na jednej maszynie, tak by żadne zadanie nie było spóźnione.