Szeregowanie zadań

Wykład nr 5

dr Hanna Furmańczyk

Procesory dedykowane

Przypomnienie:

- 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.

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.

Flow-shop - system przepływowy

Twierdzenie

Problem $F3||C_{max}$ jest NP-trudny.

Flow-shop - system przepływowy

Twierdzenie

Problem $F3||C_{max}$ jest NP-trudny.

Dowód

Redukcja *Problemu Podziału (PP)*:
Dany jest ciąg liczb naturalnych $a_1, a_2, \ldots, a_n$ tż. $S = \sum_{i=1}^{n} a_i$ jest l. parzystą.
Pytanie: Czy istnieje jego podciąg o sumie $S/2$?

Flow-shop - system przepływowy

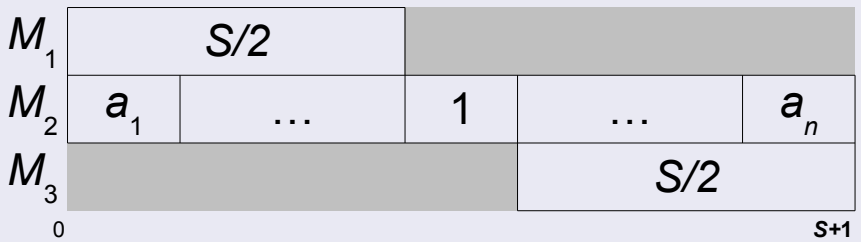
Twierdzenie

Problem $F3||C_{max}$ jest NP-trudny.

Dowód

Redukcja *Problemu Podziału (PP)*:
Dany jest ciąg liczb naturalnych $a_1, a_2, \ldots, a_n$ tż. $S = \sum_{i=1}^{n} a_i$ jest l. parzystą.
Pytanie: Czy istnieje jego podciąg o sumie $S/2$?

Redukcja: bierzemy n zadań o czasach $(0, a_i, 0)$ $i = 1, \ldots, n$ oraz jedno z czasami $(S/2, 1, S/2)$. Pytamy o istnienie uszeregowania z



System zwykły a permutacyjny (PF)

Permutacyjny system przepływowy (PF)

system przepływowy + kolejność podejmowania operacji z poszczególnych zadań musi być jednakowa na każdej maszynie (permutacja numerów zadań).

System zwykły a permutacyjny (PF)

Permutacyjny system przepływowy (PF)

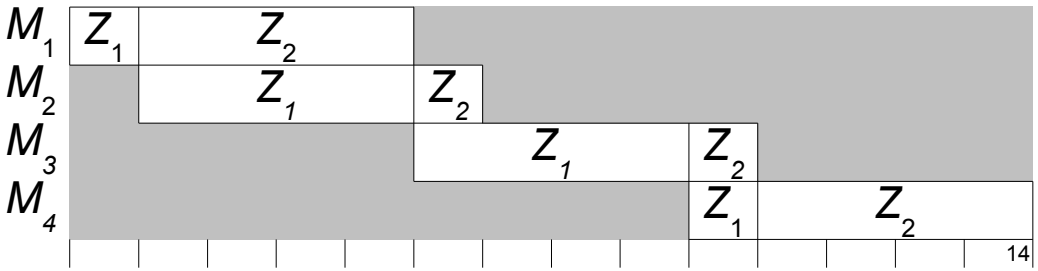
system przepływowy + kolejność podejmowania operacji z poszczególnych zadań musi być jednakowa na każdej maszynie (permutacja numerów zadań).

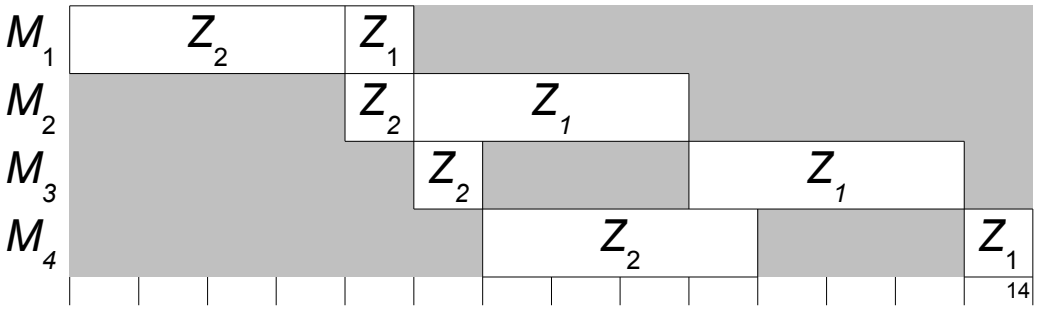
Zwykły system przepływowy

Operacje w zadaniach wykonują się w tej samej kolejności (numeracja procesorów) ale kolejność podejmowania zadań może się zmieniać pomiędzy maszynami. Jest to możliwe nawet w harmonogramie optymalnym.

Przykład

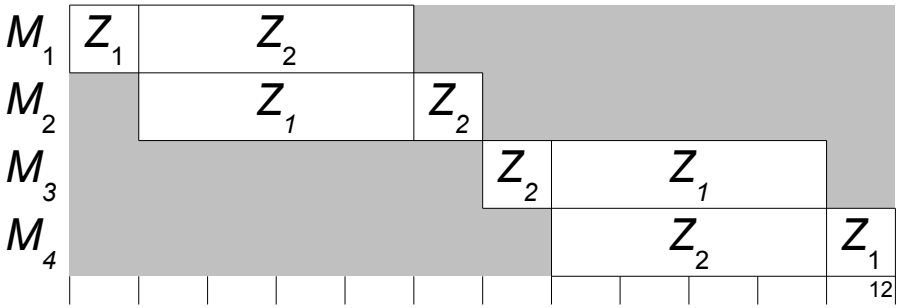
$m = 4, n = 2$. Czasy wykonania $(1, 4, 4, 1)$ dla Z_1 i $(4, 1, 1, 4)$ dla Z_2.
Harmonogramy permutacyjne:





Przykład

Harmonogram niepermutacyjny:



Obserwacje

Jeżeli $p_{ij} > 0$, to istnieje optymalne uszeregowanie flow shopu, w którym kolejność podejmowania zadań jest jednakowa na pierwszych dwóch maszynach, oraz jednakowa na ostatnich dwóch.

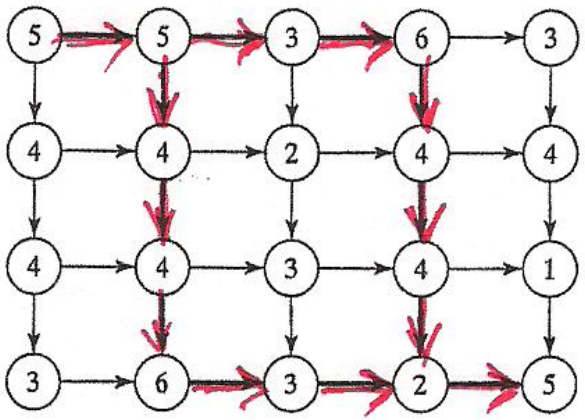
Obserwacje

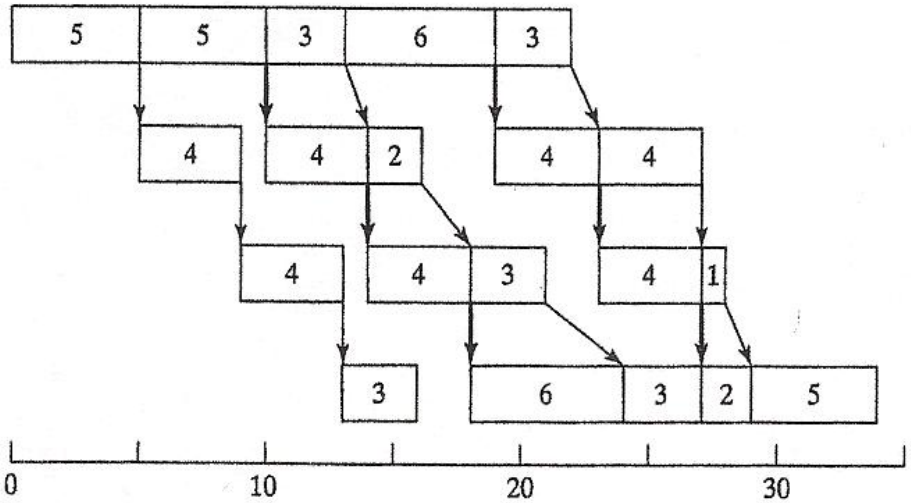
Jeżeli $p_{ij} > 0$, to istnieje optymalne uszeregowanie flow shopu, w którym kolejność podejmowania zadań jest jednakowa na pierwszych dwóch maszynach, oraz jednakowa na ostatnich dwóch.

Harmonogram optymalny dla $PFm||C_{\max}$ $(p_{ij} > 0)$ jest optymalny dla $Fm||C_{\max}$ przy $m \leq 3$ (sprawdzamy więc tylko harmonogramy permutacyjne, mniej do przeszukania!).

Szukanie C_{max} dla zadanej permutacji zadań - skan z książki M. Pinedo (Scheduling. Theory, Algorithms and Systems; Prentice Hall 2002)

jobs	j_1	j_2	j_3	j_4	j_5
p_{1,j_k}	5	5	3	6	3
p_{2,j_k}	4	4	2	4	4
p_{3,j_k}	4	4	3	4	1
p_{4,j_k}	3	6	3	2	5

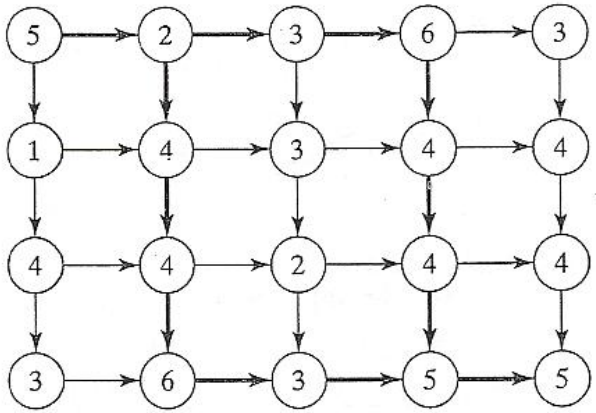


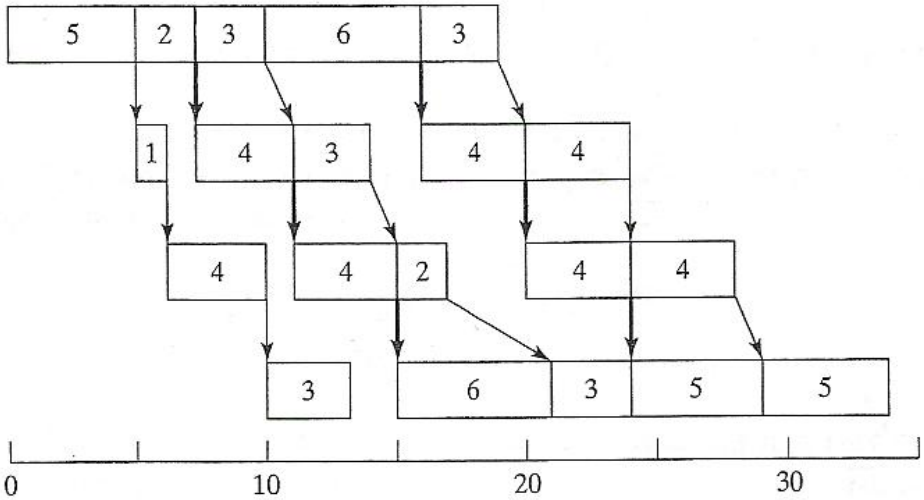


Wartość C_{max} się nie zmienia dla problemu dualnego - zadania wykonujemy w odwrotnej kolejności (permutacja $Z_n, \ldots, Z_1$), odwrotna kolejność maszyn $M_m, \ldots, M_1$ - przykład.

jobs	j_1	j_2	j_3	j_4	j_5
p_{1,j_k}	5	5	3	6	3
p_{2,j_k}	4	4	2	4	4
p_{3,j_k}	4	4	3	4	1
p_{4,j_k}	3	6	3	2	5

jobs	j_1	j_2	j_3	j_4	j_5
p_{1,j_k}	5	2	3	6	3
p_{2,j_k}	1	4	3	4	4
p_{3,j_k}	4	4	2	4	4
p_{4,j_k}	3	6	3	5	5


5 2 3 6 3
1 4 3 4 4
4 4 2 4 4
3 6 3 5 5


5 2 3 6 3
1 4 3 4 4
4 4 2 4 4
3 6 3 5 5
0
10
20
30

$F2||C_{max}$, $(F2|pmtn|C_{max})$

Jedyny problem *flow* rozwiazywalny w czasie wielomianowym, kiedy to zadania są dowolne.

Algorytm Johnsona, $O(n \log n)$

1. Podziel zadania na zbiory $N_1 = \{Z_j : p_{1j} < p_{2j}\}$, $N_2 = \{Z_j : p_{1j} \geq p_{2j}\}$.
2. Porządkuj N_1 w kolejności niemalejącej p_{1j} a N_2 według nierosnącego p_{2j}
3. Utwórz harmonogram permutacyjny (maksymalnie „przesunięty w lewo") na podstawie kolejności N_1, N_2.

Przykład - alg. Johnsona

Przykład. Algorytm Johnsona, m=2, n=5.

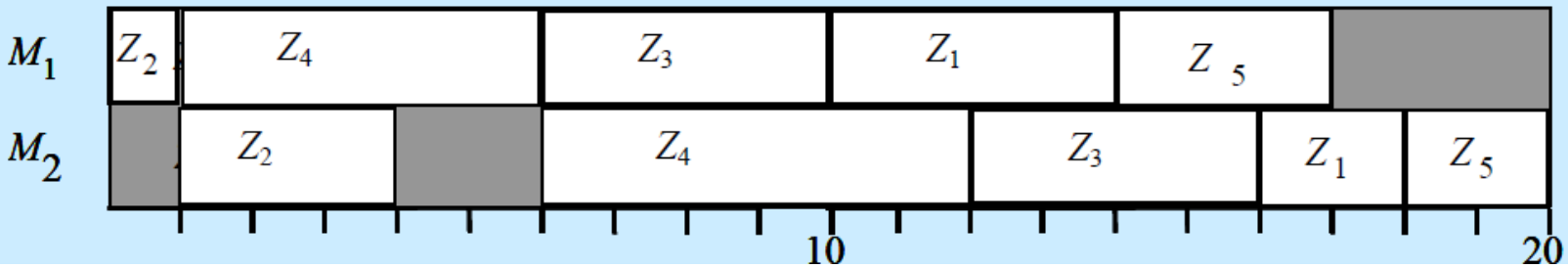
	Z_1	Z_2	Z_3	Z_4	Z_5
M_1	4	1	4	5	3
M_2	2	3	4	6	2
	N_2	N_1	N_2	N_1	N_2

N_1: Z_2 Z_4

1 5

N_2: Z_3 Z_1 Z_5

4 2 2



Zmodyfikowany algorytm Johnsona

Stosujemy, gdy M_2 jest zdominowana przez M_1 ($\forall_{i,j} p_{1i} \geq p_{2j}$) lub przez M_3 ($\forall_{i,j} p_{3i} \geq p_{2j}$) można użyć Johnsona stosując zmodyfikowane czasy wykonania $(p_{1i} + p_{2i}, p_{2i} + p_{3i})$, $i = 1, \ldots, n$.

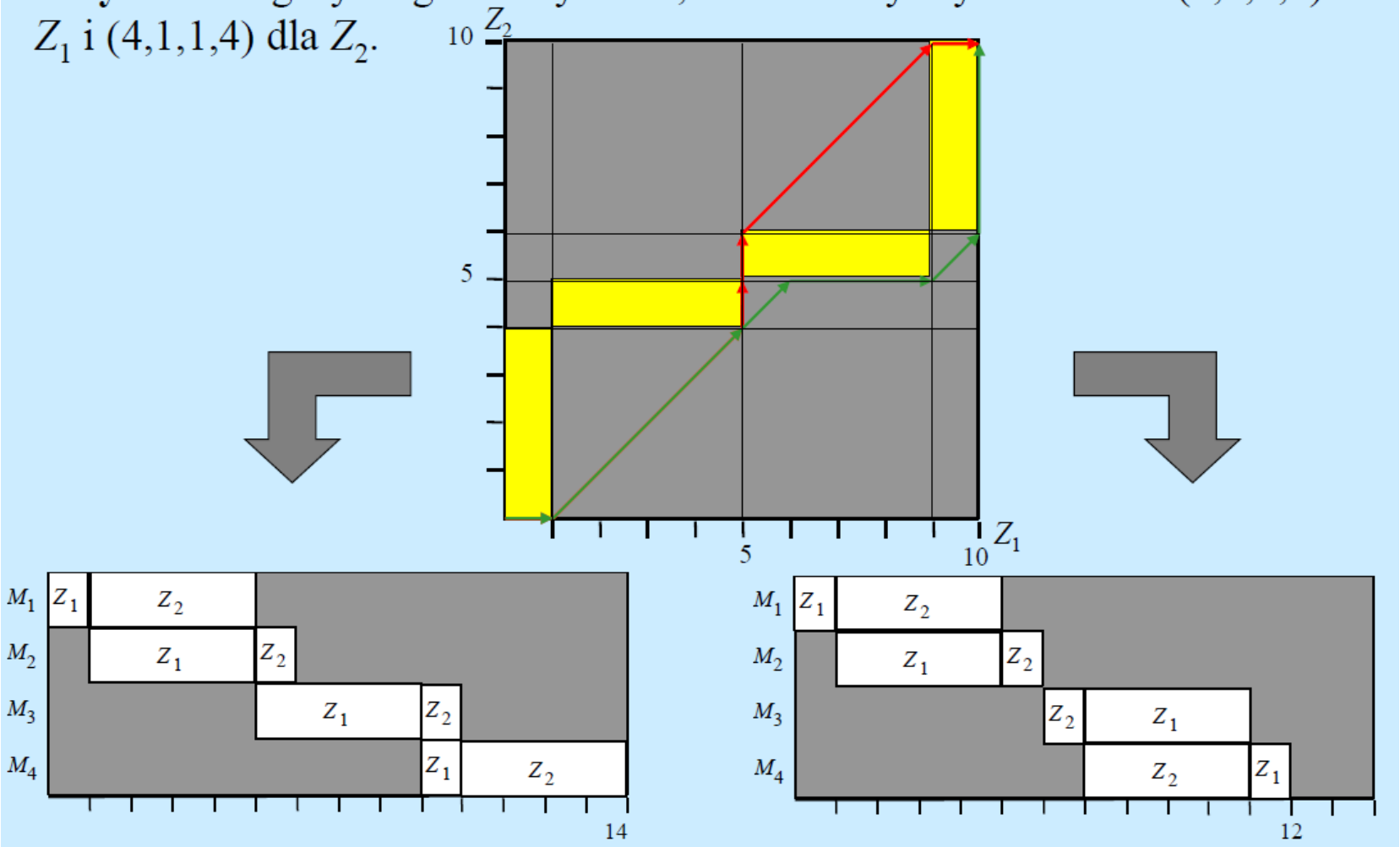
Przykład - slajdy dodatkowe

Algorytm graficzny

1. Na osi OX odkładamy kolejne odcinki o długości $p_{11}, p_{21}, \ldots, p_{m1}$ (czasy pracy maszyn nad Z_1). Na osi OY odkładamy odcinki o długości $p_{12}, p_{22}, \ldots, p_{m2}$ (czasy pracy maszyn nad Z_2).
2. Zaznaczamy obszary zakazane – wnętrza prostokątów będących iloczynami kartezjańskimi odpowiednich odcinków (ta sama maszyna nie pracuje równocześnie nad dwoma zadaniami).
3. Szukamy najkrótszej łamanej o odcinkach równoległych do osi (praca jednej maszyny) lub biegnących pod kątem $\pi/4$ (równoczesna praca obu maszyn), łączącej $(0,0)$ z $(\sum_i p_{i1}, \sum_i p_{i2})$. Jej długość to długość harmonogramu. $(d((x_1,x_2),(y_1,y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\})$.

Przykład - algorytm graficzny

Przykład. Algorytm graficzny. m=4, n=2 i czasy wykonania to (1,4,4,1) dla Z_1 i (4,1,1,4) dla Z_2.



System przepływowy permutacyjny proporcjonalny, $Fm|pmtn, p_{ij} = p_j|C_{\max}$

Reguła SPT-LPT

Zbiór zadań podzielony jest na dwie części:

1. część: $p_{j_1} \leq p_{j_2} \leq \cdots \leq p_{j_k}$
2. część: $p_{j_k} \geq p_{j_{k+1}} \geq \cdots \geq p_{j_n}$

System przepływowy permutacyjny proporcjonalny, $Fm|pmtn, p_{ij} = p_j|C_{\max}$

Reguła SPT-LPT

Zbiór zadań podzielony jest na dwie części:

1. część: $p_{j_1} \leq p_{j_2} \leq \cdots \leq p_{j_k}$
2. część: $p_{j_k} \geq p_{j_{k+1}} \geq \cdots \geq p_{j_n}$

Może być kilka takich podziałów

System przepływowy permutacyjny proporcjonalny, $Fm|pmtn, p_{ij} = p_j|C_{\max}$

Twierdzenie

Dla problemu $Fm|pmtn, p_{ij} = p_j|C_{\max}$

$$C_{\max} = \sum_{j=1}^{n} p_j + (m-1)\max\{p_1, \ldots, p_n\}$$

i nie zależy od uszeregowania.

Złożoność *flow shop* - Książka P. Bruckera

$F2 \mid p_{ij} = 1; prec; r_i \mid \sum C_i$	Baptiste & Timkowsky [23]	
$Fm \mid p_{ij} = 1; intree \mid \sum C_i$	Averbakh etal.[11]	
$F \mid p_{ij} = 1; outtree; r_i \mid C_{\max}$	Bruno et al. [60]	$O(n^2)$
$F \mid p_{ij} = 1; tree \mid C_{\max}$	Bruno et al.[60]	$O(n)$
$F2 \parallel C_{\max}$	Johnson [120]	$O(n \log n)$
	6.2.1	
$F2 \mid pmtn \mid C_{\max}$	Gonzales & Sahni [105]	$O(n \log n)$
$F \mid p_{ij} = 1; intree \mid L_{\max}$	Bruno et al. [60]	$O(n^2)$
$F2 \mid p_{ij} = 1; prec; r_i \mid L_{\max}$	Bruno et al. [60]	$O(n^3 \log n)$
$F \mid p_{ij} = 1; outtree; r_i \mid \sum C_i$	Brucker & Knust [44]	$O(n \log n)$
$F2 \mid p_{ij} = 1; prec \mid \sum C_i$	Brucker & Knust [44]	$O(n^{\log 7})$
$F \mid p_{ij} = 1; r_i \mid \sum w_i U_i$	Single machine problem	$O(n^3)$
$F \mid p_{ij} = 1; r_i \mid \sum w_i T_i$	Single machine problem	$O(n^3)$

Table 6.4: Polynomially solvable flow shop problems.

$*\ F \mid p_{ij} = 1; intree; r_i \mid C_{\max}$	Brucker & Knust [44]
$*\ F \mid p_{ij} = 1; prec \mid C_{\max}$	Leung et al. [156]
$*\ F2 \mid chains \mid C_{\max}$	Lenstra et al. [155]
$*\ F2 \mid chains; pmtn \mid C_{\max}$	Lenstra [150]
$*\ F2 \mid r_i \mid C_{\max}$	Lenstra et al. [155]
$*\ F2 \mid r_i; pmtn \mid C_{\max}$	Gonzales & Sahni [105]
$*\ F3 \parallel C_{\max}$	Garey et al. [100]
$*\ F3 \mid pmtn \mid C_{\max}$	Gonzales & Sahni [105]
$*\ F \mid p_{ij} = 1; outtree \mid L_{\max}$	Brucker & Knust [44]
$*\ F2 \parallel L_{\max}$	Lenstra et al. [155]
$*\ F2 \mid pmtn \mid L_{\max}$	Gonzales & Sahni [105]
$*\ F2 \parallel \sum C_i$	Garey et al. [100]
$*\ F2 \mid pmtn \mid \sum C_i$	Du & Leung [83]
$*\ Fm \mid p_{ij} = 1; chains \mid \sum w_i C_i$	Tanaev et al. [194]
$*\ Fm \mid p_{ij} = 1; chains \mid \sum U_i$ for each $m \geq 2$	Brucker & Knust [44]
$*\ Fm \mid p_{ij} = 1; chains \mid \sum T_i$ for each $m \geq 2$	Brucker & Knust [44]

Table 6.5: $\mathcal{NP}$-hard flow shop problems.