



Politechnika
Wrocławska

STEROWANIE PROCESAMI DYSKRETNymi

Sprawozdanie 2

Autorzy:

Adam Prystupa, 275496

Mikołaj Sobala, 275439

1 Opis problemu

W problemie $Pm \parallel C_{\max}$ rozpatrujemy m identycznych maszyn pracujących równolegle oraz n niepodzielnych zadań. Każde zadanie $j \in J = \{1, 2, \dots, n\}$ charakteryzuje się czasem wykonania p_j i musi zostać przypisane dokładnie jednej maszynie spośród zbioru $M = \{1, 2, \dots, m\}$. Celem jest takie przyporządkowanie zadań do maszyn, aby **zminimalizować czas zakończenia najpóźniej kończącego się zadania**, czyli **zminimalizować maksymalne obciążenie maszyn**. Formalnie, minimalizujemy wartość:

$$C_{\max} = \max_{j \in J} C_j,$$

gdzie C_j oznacza czas zakończenia zadania j .

2 Opis zaimplementowanych metod

W ramach projektu zrealizowano implementację następujących metod rozwiązywania problemu szeregowania zadań:

- **LSA (List Scheduling Algorithm)** – metoda przypisująca każde kolejne zadanie do maszyny o najmniejszym bieżącym czasie pracy, co pozwala na szybkie generowanie harmonogramów.
- **LPT (Longest Processing Time)** – rozszerzenie podejścia listowego, w którym zadania są najpierw posortowane malejąco według czasu wykonania, a następnie przypisywane do maszyn w celu lepszego zbalansowania obciążenia.
- **Programowanie dynamiczne dla $P2 \parallel C_{\max}$** – technika bazująca na programowaniu dynamicznym, analizująca możliwe podziały zadań między dwie maszyny w celu znalezienia optymalnego harmonogramu.
- **Przegląd zupełny dla $P2 \parallel C_{\max}$** – dokładny algorytm polegający na sprawdzeniu wszystkich możliwych konfiguracji przydziału zadań do dwóch maszyn, gwarantujący znalezienie rozwiązania optymalnego.
- **PTAS dla $P2 \parallel C_{\max}$** – algorytm przybliżony, który dla dowolnej zadanej dokładności $\varepsilon > 0$ zapewnia rozwiązanie nie gorsze niż $(1 + \varepsilon)$ razy wartość optymalna, przy czym czas działania jest wielomianowy przy stałym ε .
- **FPTAS dla $P2 \parallel C_{\max}$** – ulepszona wersja PTAS, której złożoność jest wielomianowa zarówno względem liczby zadań, jak i odwrotności dokładności $\frac{1}{\varepsilon}$, umożliwiając bardzo dokładne rozwiązania w rozsądnym czasie.

3 Wyniki

W celu porównania algorytmów wykonano testy dla różnych rozmiarów instancji. Dla instancji o rozmiarach powyżej 11 przegląd zupełny nie był wykonywany. Wartość błędu była liczona od wartości przeglądu zupełnego/programowania dynamicznego

Parametry komputera

MaxClockSpeed	Name	NumberOfCores	NumberOfLogicalProcessors
3301	AMD Ryzen 5 5600H with Radeon Graphics	6	12

Rysunek 1: Procesor

Capacity_MB	Speed_MHz
16384	3200

Rysunek 2: Pamięć RAM

3.1 LSA, LPT, PD, przegląd zupełny

W pierwszej tabeli znajdują się algorytmy: LSA, LPT, PD oraz przegląd zupełny

Rozmiar Instancji	Wartość kryterium LPT	Czas działania algorytmu LPT	Wartość kryterium LSA	Czas działania algorytmu LSA	Wartość kryterium PD	Czas działania algorytmu PD	Czas działania przeglądu zupełnego
[-]	[-]	[ns]	[-]	[ns]	[-]	[ns]	[ns]
2/10 [1-10]	29 [0%]	18 200	33 [13,793%]	3 800	29	21 200	1 810 102 600
2/10 [10-20]	88 [2,326%]	5 100	92 [6,977%]	2 800	86	45 800	1 787 156 000
2/20 [1-10]	44 [0%]	6 300	45 [2,273%]	4 300	44	36 500	-
2/20 [10-20]	149 [0%]	6 200	157 [5,369%]	3 400	149	106 700	-
2/20 [50-100]	735 [0,136%]	7 300	735 [0,136%]	3 000	734	425 100	-
2/50 [1-10]	136 [0%]	13 400	146 [7,353%]	5 600	136	299 200	-
2/50 [10-20]	367 [0%]	24 400	376 [2,452%]	5 200	367	624 500	-
2/50 [50-100]	1913 [0%]	12 900	1933 [1,045%]	4 700	1913	3 081 200	-

Rysunek 3: LSA, LPT, PD, przegląd zupełny

3.2 PTAS i FPTAS

W poniższej tabeli przedstawione zostały wartości kryteriów algorytmów PTAS oraz FPTAS, a poniżej czasy działania

Rozmiar Instancji	PTAS						FPTAS					
	K=n/2		K=2n/3		K=3n/4		K=n/2		K=2n/3		K=3n/4	
	Wartość kryterium	Czas	Wartość kryterium	Czas	Wartość kryterium	Czas	Wartość kryterium	Czas	Wartość kryterium	Czas	Wartość kryterium	Czas
[-]	[-]	[ns]	[-]	[ns]	[-]	[ns]	[-]	[ns]	[-]	[ns]	[-]	[ns]
2/10 [1-10]	29 [0%]	22 800	29 [0%]	23 600	29 [0%]	19 900	37 [27,586%]	7 600	31 [6,897%]	8 400	47 [62,069%]	6 600
2/10 [10-20]	88 [2,326%]	10 200	88 [2,326%]	10 500	88 [0%]	14 000	87 [1,163%]	12 900	87 [1,163%]	13 200	87 [1,163%]	13 300
2/20 [1-10]	44 [0%]	58 200	44 [0%]	564 500	44 [0%]	2 354 200	87 [97,727%]	11 400	87 [97,727%]	11 800	87 [97,727%]	14 800
2/20 [10-20]	149 [0%]	74 400	154 [3,356%]	561 000	154 [3,356%]	2 181 300	156 [4,698%]	17 700	187 [25,503%]	18 900	215 [44,295%]	11 200
2/20 [50-100]	735 [0,136%]	57 100	758 [3,27%]	570 800	758 [3,27%]	2 186 100	748 [1,907%]	51 000	744 [1,362%]	52 600	744 [1,362%]	36 000
2/50 [1-10]	136 [0%]	3 230 086 300	220 [61,765%]	13 900	182 [33,824%]	19 100	272 [100%]	15 400	272 [100%]	12 000	272 [100%]	12 000
2/50 [10-20]	371 [1,09%]	3 238 043 500	502 [36,785%]	18 100	425 [15,804%]	28 900	733 [99,728%]	20 500	733 [99,728%]	12 200	733 [99,728%]	13 200
2/50 [50-100]	1938 [1,307%]	3 232 104 700	2632 [37,585%]	15 000	2248 [17,512%]	21 100	1927 [0,732%]	151 800	1933 [1,045%]	93 300	1952 [2,039%]	93 100

Rysunek 4: Wartości kryterium i czasy uzyskane przez PTAS i FPTAS

4 Podsumowanie

4.1 Problemy

Podczas realizacji projektu nie napotkano znaczących problemów, wynika to prawdopodobnie z faktu, iż algorytmy takie jak przegląd zupełny były podobne do poprzedniego problemu.

4.2 Wnioski

- **LPT (Longest Processing Time):**
 - Zwykle uzyskuje bardzo dobre wyniki – w większości przypadków błąd względem optymalnego rozwiązania jest zerowy lub minimalny (maksymalnie ok. 2,326%).
 - Dłuższy czas działania w porównaniu z LSA wynika z konieczności sortowania wszystkich zadań od najdłuższego do najkrótszego oraz dynamicznego przypisywania ich do maszyn.
 - Pomimo większego kosztu obliczeniowego, LPT warto stosować, gdy zależy nam na precyzji i akceptujemy nieco dłuższy czas wykonywania.
- **LSA (List Scheduling Algorithm):**
 - Dokładność algorytmu jest na dobrym poziomie – największy odnotowany błąd (ok. 13,793%) pojawia się tylko w małych instancjach z małymi wartościami P_j .
 - Dla większych instancji lub bardziej zróżnicowanych P_j błąd spada do pojedynczych procent (zwykle poniżej 5%).
 - Bardzo szybki czas działania, ponieważ LSA przydziela zadania w kolejności dowolnej (bez sortowania), co stanowi zaletę, jeśli priorytetem jest krótki czas obliczeń, nawet kosztem nieco gorszego przybliżenia.
- **PD (Permutation–Dynamic):**
 - Algorytm wykorzystuje programowanie dynamiczne oparte na permutacjach: dla każdej podgrupy zadań decyduje, które trafiają na pierwszą maszynę, a które na drugą, budując kolejne stany DP.
 - Czas działania rośnie pseudopolinomialnie – w przypadku stałej liczby zadań wzrost wartości maksymalnych P_j prowadzi do liniowego wzrostu czasu obliczeń (z kilkudziesięciu tysięcy do kilku milionów nanosekund).
 - Dla małych lub umiarkowanych wartości P_j PD zapewnia bardzo małe błędy (często zerowe) i przewyższa LSA/LPT pod względem jakości, ale przy dużych P_j staje się mało praktyczny z uwagi na wydłużający się czas wykonania.
- **PTAS (Polynomial Time Approximation Scheme):**
 - Dla małych instancji (10–20 zadań) PTAS daje przybliżenie rzędu 1%, niezależnie od parametru K , co jest bardzo dobrym wynikiem.
 - W instancjach z 50 zadaniami widać wyraźny wpływ parametru K :
 - * Dla $K = n/2$ błąd pozostaje niski (poniżej kilku procent), ale czas wykonania rośnie bardzo mocno (rzędu kilku milionów nanosekund), co utrudnia zastosowanie w praktyce.

- * Dla $K = 2n/3$ lub $K = 3n/4$ czas działania jest znacznie krótszy, jednak błąd przybliżenia wzrasta – wynosi od kilkunastu do kilkudziesięciu procent w zależności od instancji.
- PTAS wymaga balansu między dokładnością a czasem wykonania: niższe K dają lepszą jakość, ale kosztem większego czasu obliczeń.

- **FPTAS (Fully Polynomial Time Approximation Scheme):**

- Teoretycznie FPTAS powinien zawsze zapewniać co najmniej taką samą jakość przybliżenia jak PTAS, ale w testach uzyskał gorsze wyniki w wielu przypadkach.
- Najlepsze efekty widać przy wyższych wartościach P_j , gdzie błąd potrafi być zbliżony do PTAS, natomiast w innych zakresach P_j błędy sięgają kilkunastu–kilkudziesięciu procent, co sugeruje możliwe niedociągnięcia w implementacji.
- Czas działania FPTAS jest na ogół krótszy od odpowiadających mu konfiguracji PTAS (szczególnie przy wyższych K), ale przy niektórych instancjach przyspieszenie jest niewielkie lub go brak, co wymaga dodatkowej weryfikacji poprawności kodu.