TEORIA WSPÓŁBIEŻNOŚCI

SPRAWOZDANIE: PROBLEM UCZTUJĄCYCH FILOZOFÓW

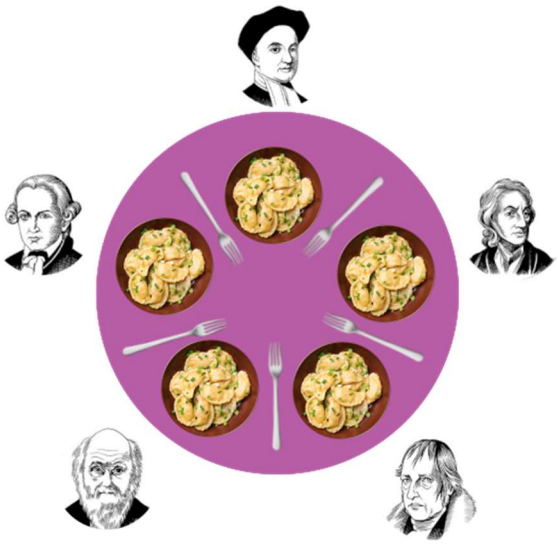
Aleksandra Smela

SPIS TREŚCI

1. OPIS ZADANIA

Problem pięciu filozofów jest jednym z klasycznych problemów teorii współbieżności. Podstawowe sformułowanie problemu jest następujące:

- N filozofów zasiada przy okrągłym stole;
- Pomiędzy sąsiednimi filozofami leży widelec (łącznie jest N widelców);
- Każdy filozof działa ciągle według schematu „myślenie - jedzenie - myślenie - jedzenie - ...”. Każdy z etapów (myślenie i jedzenie) jest skończony;
- Aby zjeść, filozof musi podnieść oba sąsiadujące widelce.



rys I: Ilustracja problemu ucztujących filozofów

Rozwiązanie problemu to algorytm, który realizuje jednoczesną alokację współdzielonych zasobów (widelce) przez konkurujące procesy (filozofowie), tak aby uniknąć zakleszczenia i zagłodzenia.

Celem ćwiczenia jest porównanie różnych rozwiązań problemu pięciu filozofów i ich implementacji.

2. UŻYTE NARZĘDZIA

Algorytmy zostały zaimplementowane w języku Java. Pomiary zapisano do plików csv, a następnie wczytano je do skryptu w języku Python. Skrypt został napisany z użyciem Jupyter Notebook i bibliotek *pandas*, *numpy* i *matplotlib*.

Wszystkie obliczenia wykonano na komputerze o specyfikacji:

- System: Windows 10
- Procesor: AMD Ryzen 7 3700X 3,6GHz
- Pamięć RAM: 32 GB

3. ROZWIĄZANIA

3.1. ROZWIĄZANIE NAIWNE

W tym podejściu każdy filozof podnosi najpierw lewy widelec, a następnie prawy. Aby podnieść widelec filozof musi poczekać aż będzie wolny, a następnie go zajmuje. W takim rozwiązaniu występuje ryzyko zakleszczenia.

3.2. ROZWIĄZANIE Z RÓWNOCZESNYM PODNOSZENIEM WIDELCÓW

Każdy filozof jednocześnie zajmuje oba widelce. Jeżeli dowolny z nich jest zajęty, to nie podnosi żadnego, a następnie ponawia próbę. W tym przypadku występuje ryzyko wystąpienia zagłodzenia – jeżeli zawsze jeden z sąsiadów jest zajęty jedzeniem, filozof nie zostanie dopuszczony do zasobu.

3.3. ROZWIĄZANIE ASYMETRYCZNE

Filozofowie są ponumerowani. Wszyscy filozofowie z parzystym numerem podnoszą zawsze prawy widelec, a ci z nieparzystym podnoszą lewy.

3.4. ROZWIĄZANIE STOCHASTYCZNE

Każdy filozof losuje kolejność podniesienia widelców i na tej podstawie decyduje czy najpierw podnieść prawy czy lewy.

3.5. ROZWIĄZANIE Z ARBITREM

Zawsze jedynie N-1 filozofów może jeść jednocześnie. Zewnętrzny arbiter powstrzymuje jednego z filozofów, aż któryś skończy jeść. Filozofowie najpierw podnoszą prawy a potem lewy widelec.

3.6. ROZWIĄZANIE Z JADALNIĄ

Filozofowie podnoszą prawy, a następnie lewy widelec tylko jeśli znajdują się w jadalni, w której jest N-1 miejsc. Pozostały filozof je na korytarzu i podnosi najpierw prawy a potem lewy widelec. Po zjedzeniu filozofowie wychodzą na korytarz i próbują ponownie dostać się do jadalni.

4. SZCZEGÓŁY TECHNICZNE IMPLEMENTACJI ROZWIĄZAŃ I DOKONYWANIA POMIARÓW

4.1. PRZYGOTOWANIE ALGORYTMÓW

Stworzono 6 klas filozofów, które rozszerzają klasę *Thread*. Każda klasa implementuje jedno z rozwiązań problemu. Każdy filozof przechowuje informację o tym, który widelec znajduje się po jego lewej i który po prawej. Widelec jest reprezentowany przez osobną klasę zawierającą informację o tym czy jest obecnie używany czy nie.

We wszystkich rozwiązaniach poza rozwiązaniem 3.2 podnoszenie widelców zrealizowano z wykorzystaniem sekcji *synchronized*, dla której monitorem był dany widelec. Filozofowie czekają na widelec przez wykorzystanie metody *wait,* a po zjedzeniu odkładają widelec i informują jeżeli inny filozof na niego czeka metodą *notify*.

Rozwiązania 3.5 i 3.6 wykorzystują ponadto semafor, determinujący o tym który z filozofów nie je albo je podnosząc widelce w odwrotnej kolejności, w zależności od rozwiązania.

W rozwiązaniu 3.2 wykorzystano *ReentrantLock* i metody *tryLock* aby zasymulować jednoczesne podniesienie dwóch widelców. Jeżeli metoda *tryLock* zwróci *false*, czyli nie uda się podnieść co najmniej jednego z widelców, to podniesiony widelec jest opuszczany metodą *unlock*.

W klasie *Parameters* umieszczono parametry symulacji takie jak:

- *eatingTime* – czas, jaki zajmuje filozofom pojedyncza akcja jedzenia: w przeprowadzonych testach **2 ms**;
- *waitingTime* – czas oczekiwania filozofów przed rozpoczęciem uczty: w przeprowadzonych testach **0,5 s**;
- *testTime* – czas każdego z testów: **20 s**,

gdzie testem nazywam działanie jednego z algorytmów dla określonej liczby *n* filozofów.

Każdy z filozofów mierzy czas oczekiwania na jedzenie. Wykonano dwa różne rodzaje pomiarów i w zależności od oczekiwanej jednostki czas zmierzono używając *System.nanoTime()* lub *System.currentTimeMillis()*.

4.2. POMIARY I

- Pomiarów dokonano dla różnych liczb ucztujących filozofów: 5, 6, 8, 10, 15, 20, 30, 50, 75, 100, 150, 200.
- Czas zmierzono w milisekundach używając *System.currentTimeMillis()*.

- Sprawdzono czy w jakimkolwiek przypadku doszło do zagłodzenia lub zakleszczenia.
- Obliczono:
  - **Średni czas oczekiwania między posiłkami.**

$$\frac{\sum_{k=0}^{N} avg\_time_k}{N}$$

gdzie:
N – liczba filozofów,
avg_time_k – średni czas oczekiwania filozofa k na pojedynczy posiłek.

  - **Średni sumaryczny czas oczekiwania na wszystkie posiłki.**

$$\frac{\sum_{k=0}^{N} sum\_time_k}{N}$$

gdzie:
N – liczba filozofów,
sum_time_k – suma czasów oczekiwania filozofa k na posiłki.

  - **Średnią liczbę skonsumowanych posiłków – ile razy wątek otrzymał zasoby.**

$$\frac{\sum_{k=0}^{N} n_k}{N}$$

gdzie:
N – liczba filozofów,
n_k – suma posiłków filozofa k.

  - **Maksymalny średni czas oczekiwania między posiłkami.**

$$\frac{max_{k=0}^{N}(avg\_time_k)}{N}$$

gdzie:
N – liczba filozofów,
avg_time_k – średni czas oczekiwania filozofa k na pojedynczy posiłek.

- Pomiary powtórzono 3 razy.

4.3. POMIARY II

- Pomiarów dokonano dla trzech różnych liczb ucztujących filozofów: 5, 20, 100.
- Czas zmierzono w nanosekundach *System.nanoTime()*.
- Dla każdej pary (algorytm, liczba filozofów) zapisano każdy zmierzony czas oczekiwania na pojedynczy posiłek.

4.4. OPRACOWANIE DANYCH

Wszystkie przygotowane dane zapisano do plików csv, a następnie pliki wczytano do skryptu w języku Python.

Ponieważ pomiary I powtórzono 3 razy, po wczytaniu zostały uśrednione z pomocą biblioteki *numpy*. Następnie stworzono struktury *DataFrame* z biblioteki *pandas* zawierające informacje o zmierzonych wartościach dla poszczególnych testów. Z pomocą biblioteki *matplotlib* stworzono wykresy słupkowe.

Z wykorzystaniem pomiarów II dla każdego testu za pomocą funkcji z biblioteki *numpy* obliczono medianę, średnią i wartość maksymalną czasów oczekiwania na pojedynczy posiłek. Następnie dane umieszczono na wykresach pudełkowych narysowanych z wykorzystaniem *matplotlib*.

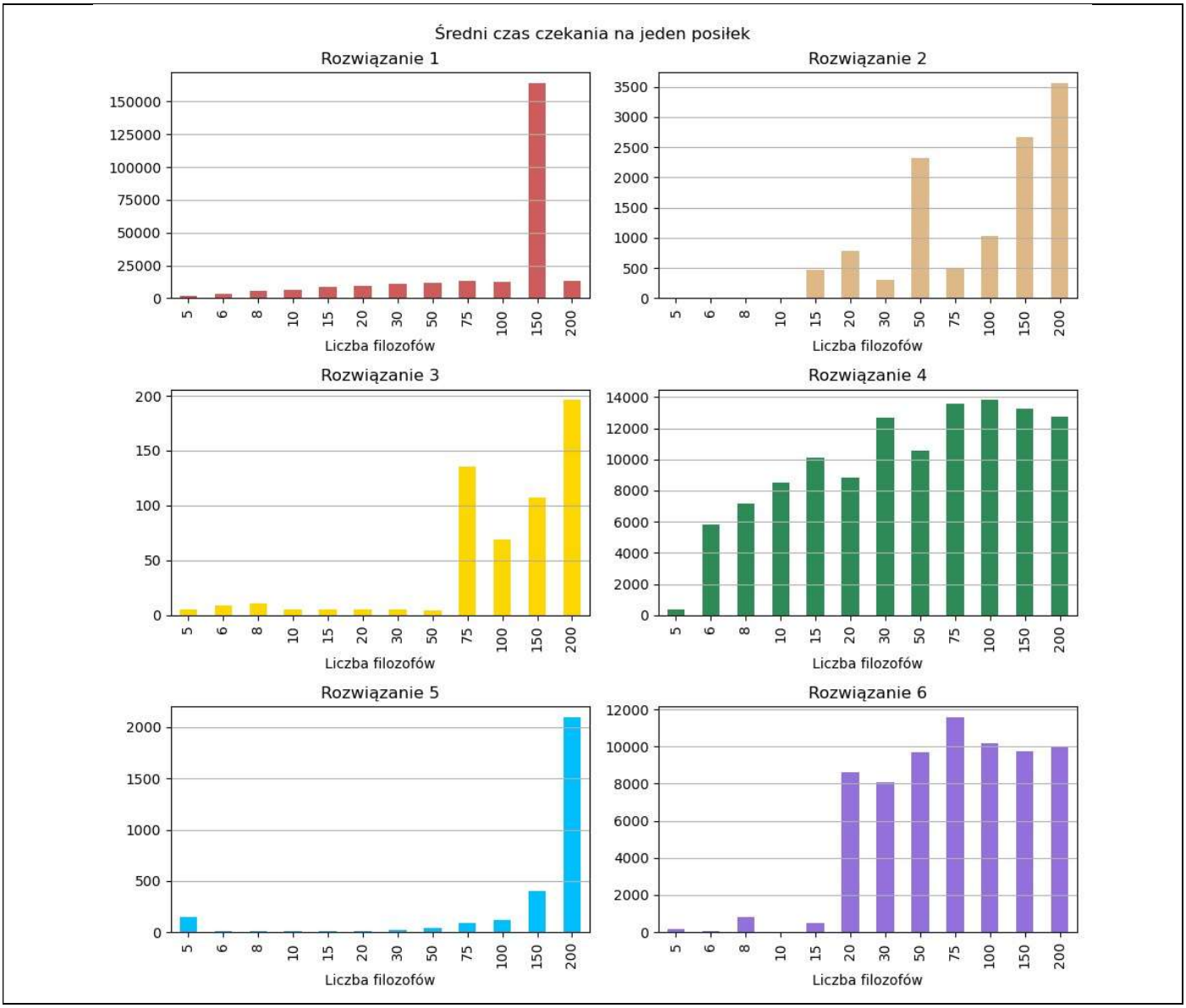
5. POMIARY I

5.1. WYNIKI POMIARÓW

5.1.1. ŚREDNI CZAS OCZEKIWANIA MIĘDZY POSIŁKAMI

liczba ucztujących filozofów	Średni czas oczekiwania między posiłkami [ms]					
	Rozwiązanie 1	Rozwiązanie 2	Rozwiązanie 3	Rozwiązanie 4	Rozwiązanie 5	Rozwiązanie 6
5	1781	9	5	369	149	153
6	3535	6	9	5831	10	52
8	5354	8	11	7166	15	837
10	6694	7	6	8534	8	9
15	8562	473	5	10102	9	516
20	9343	778	6	8841	14	8602
30	11182	299	5	12669	22	8094
50	11708	2312	5	10539	41	9675
75	13522	498	136	13577	87	11588
100	12270	1035	69	13807	116	10151
150	163876	2672	107	13274	404	9772
200	13085	3558	196	12738	2100	9961

Tabela I

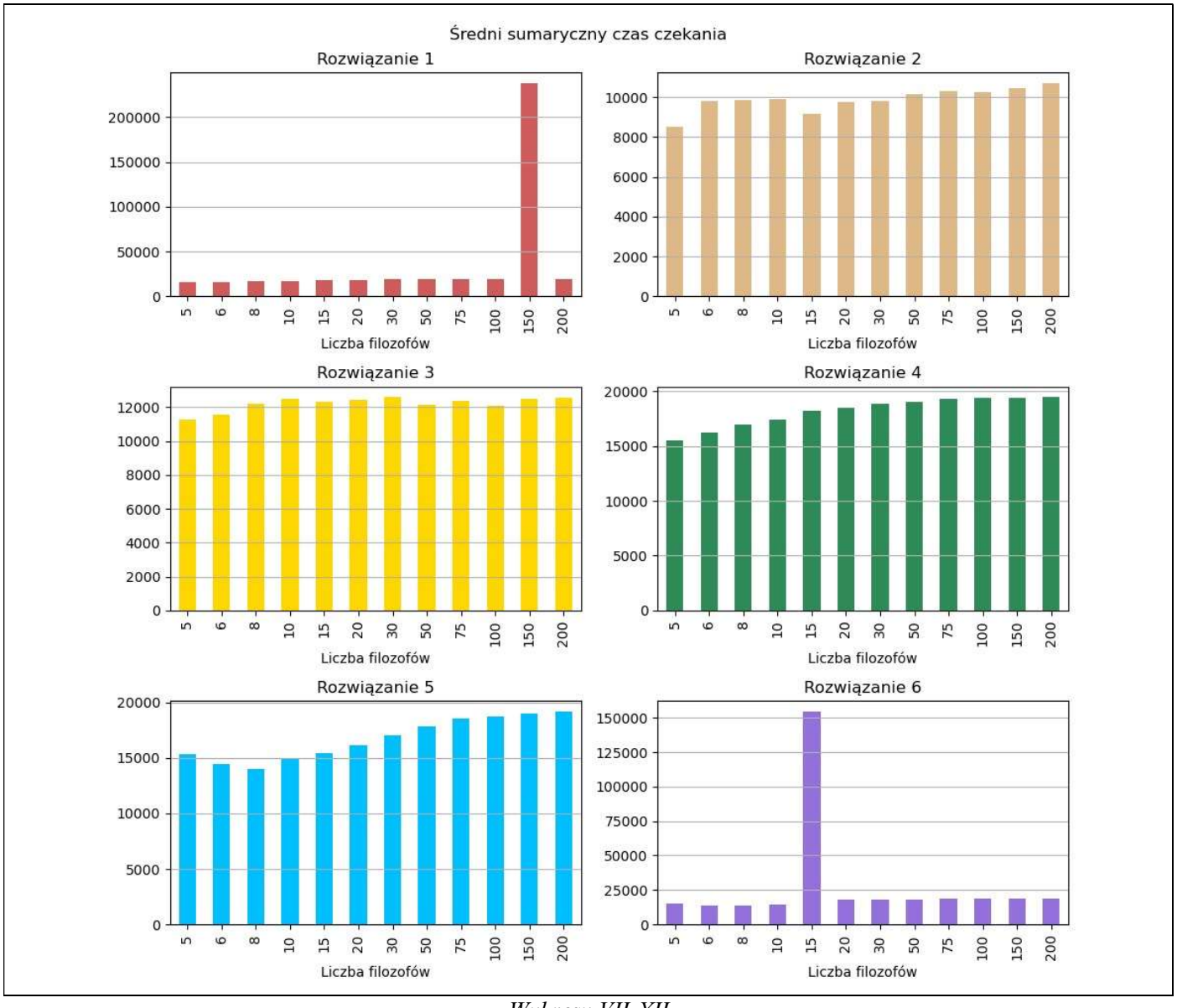


Wykresy I-VI

5.1.2. Średni sumaryczny czas oczekiwania na wszystkie posiłki

liczba ucztujących filozofów	Średni sumaryczny czas oczekiwania na wszystkie posiłki [ms]					
	Rozwiązanie 1	Rozwiązanie 2	Rozwiązanie 3	Rozwiązanie 4	Rozwiązanie 5	Rozwiązanie 6
5	15534	8526	11287	15471	15358	15290
6	16176	9785	11540	16214	14437	13974
8	17018	9869	12194	16968	13982	13595
10	17490	9918	12507	17411	14888	14280
15	18189	9178	12348	18180	15461	154640
20	18430	9775	12429	18518	16105	17901
30	18811	9790	12586	18871	17031	18162
50	19141	10144	12131	19058	17803	18123
75	19201	10298	12384	19294	18529	18435
100	19122	10260	12095	19351	18747	18590
150	237653	10468	12500	19417	19017	18667
200	19573	10694	12568	19443	19193	18721

Tabela II

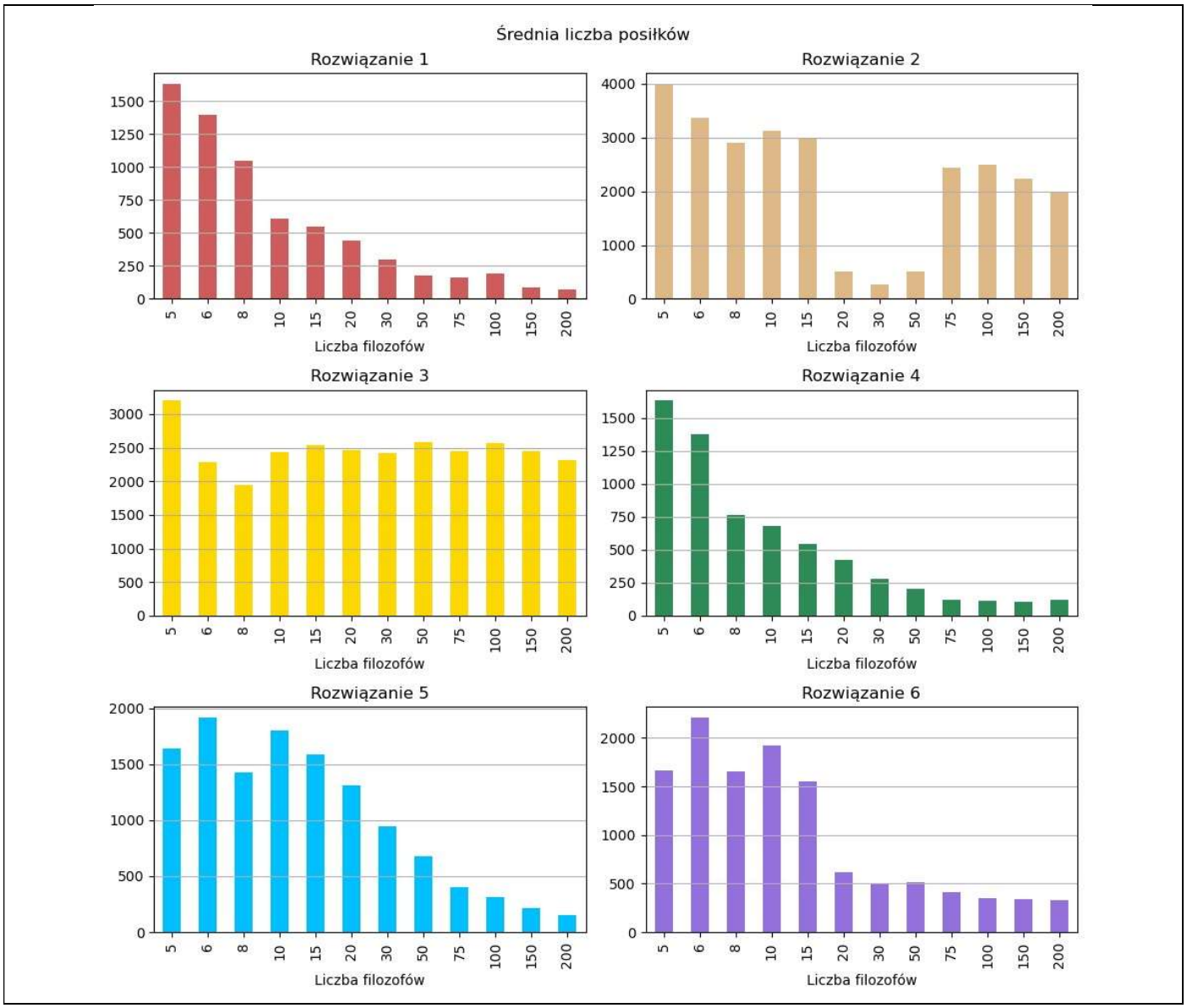


Wykresy VII-XII

5.1.3. ŚREDNIA LICZBA SKONSUMOWANYCH POSIŁKÓW

		Średnia liczba skonsumowanych posiłków					
		Rozwiązanie 1	**Rozwiązanie 2**	**Rozwiązanie 3**	**Rozwiązanie 4**	**Rozwiązanie 5**	**Rozwiązanie 6**
liczba ucztujących filozofów	**5**	1629	3998	3201	1630	1643	1662
	6	1398	3367	2286	1377	1918	2213
	8	1044	2913	1938	764	1429	1651
	10	608	3136	2428	678	1800	1925
	15	549	2997	2544	545	1586	1552
	20	443	506	2472	421	1306	614
	30	300	268	2427	279	948	506
	50	178	505	2580	203	681	512
	75	160	2437	2449	122	399	413
	100	193	2497	2564	111	310	350
	150	86	2238	2456	105	216	337
	200	72	1993	2313	116	154	329

Tabela III

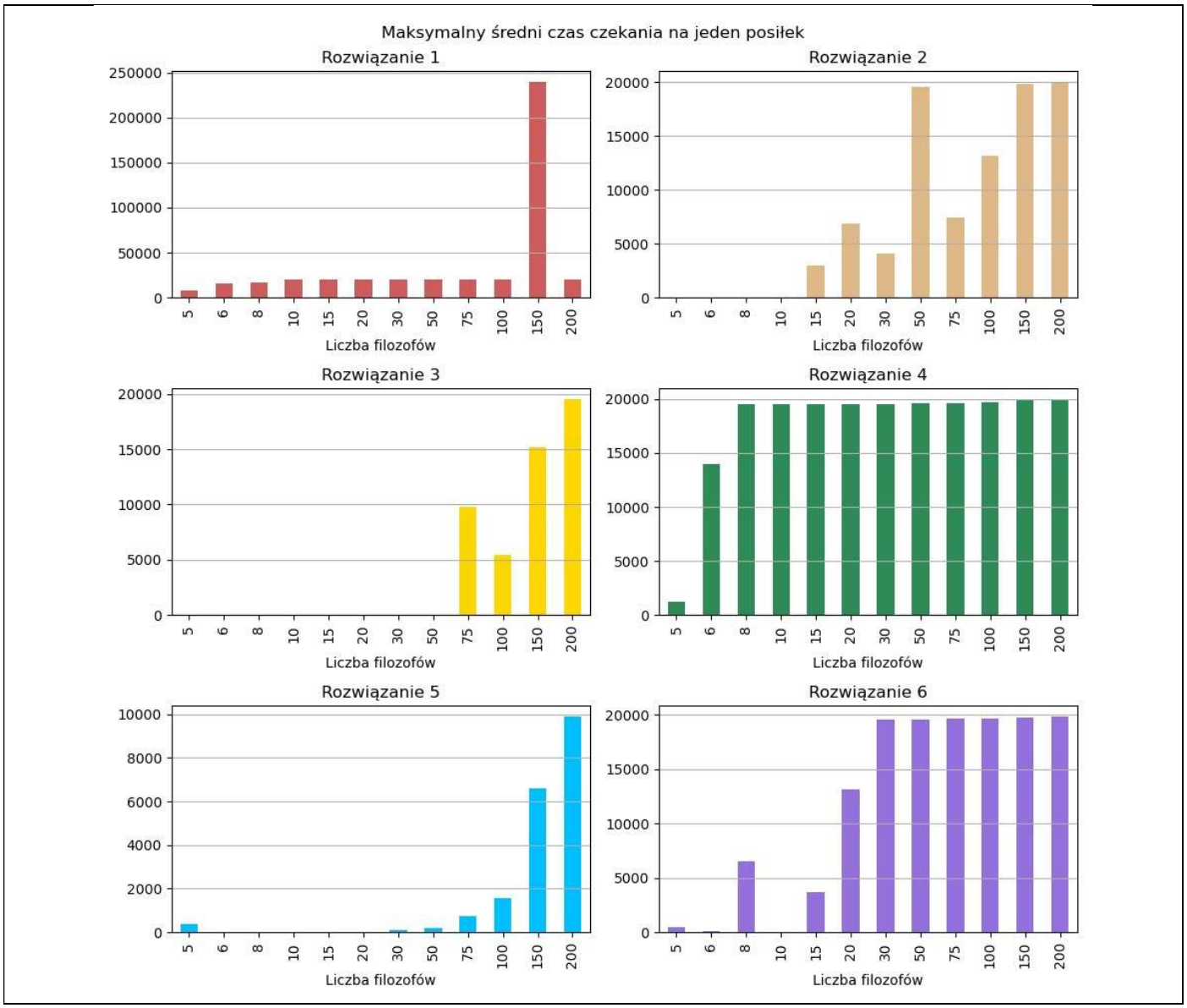


Wykresy XIII-XVIII

5.1.4. MAKSYMALNY ŚREDNI CZAS OCZEKIWANIA MIĘDZY POSIŁKAMI

liczba ucztujących filozofów	Maksymalny średni czas oczekiwania między posiłkami [ms]					
	Rozwiązanie 1	**Rozwiązanie 2**	**Rozwiązanie 3**	**Rozwiązanie 4**	**Rozwiązanie 5**	**Rozwiązanie 6**
5	7511	28	14	1184	365	509
6	15172	22	18	13933	21	125
8	16260	34	19	19531	23	6537
10	19536	38	16	19517	13	16
15	19533	3009	18	19531	14	3726
20	19540	6892	27	19536	25	13130
30	19574	4107	22	19565	113	19552
50	19635	19600	21	19610	197	19583
75	19690	7388	9760	19668	757	19630
100	19765	13150	5424	19741	1563	19625
150	239550	19864	15171	19861	6580	19724
200	20009	20006	19508	19980	9887	19838

Tabela IV



Wykresy XIX-XXIV

5.2. WNIOSKI

- Nie zaobserwowano sytuacji kompletnego zagłodzenia/zakleszczenia. Ostatecznie wszystkie wątki zakończyły pracę.
- Wartości odstające
  - Dla rozwiązania 1 i 150 filozofów zauważono bardzo duże wartości średnich i sumarycznych czasów oczekiwania na posiłek. Wynika to najprawdopodobniej z tego, że jeden lub kilka wątków zostało na dłuższy czas niedopuszczone do zasobów.
  - Dla 15 filozofów i rozwiązania 6 jest bardzo wysoki średni sumaryczny czas oczekiwania. Co ciekawe, dla tego przypadku średni czas czekania na pojedynczy posiłek i maksymalny czas oczekiwania nie są zbyt wysokie. Może to oznaczać, że czasy oczekiwania były skumulowany wokół jednej, dość wysokiej wartości, jednak nie tak wysokiej żeby wpłynąć na średnią i wartość maksymalną.
- Monotoniczność
  - Z powyższych tabel i wykresów można sformułować niewiążący wniosek, że zazwyczaj wraz ze wzrostem liczby wątków, czas oczekiwania na zasoby jest dłuższy (szczególnie dla rozwiązania 4). Nie da się jednak jednoznacznie stwierdzić monotoniczności, często występują sytuacje odstające.
- Ocena algorytmów
  - Na podstawie powyższych wykresów ciężko jest ocenić zaimplementowane algorytmy.
  - Można zauważyć dobre wyniki osiągane dla algorytmów 2, 3, 5, które przeważają nad innymi w różnych zestawieniach.

6. POMIARY II

6.1. WYNIKI POMIARÓW

	Średnia [ns]		
	5 filozofów	20 filozofów	100 filozofów
Rozwiązanie 1	9,70E+06	4,78E+07	2,24E+08
Rozwiązanie 2	1,91E+06	1,92E+07	4,63E+06
Rozwiązanie 3	3,74E+06	5,62E+06	4,82E+06
Rozwiązanie 4	9,85E+06	4,88E+07	2,42E+08
Rozwiązanie 5	9,41E+06	1,38E+07	5,38E+07
Rozwiązanie 6	9,60E+06	1,15E+07	4,36E+07

Tabela V

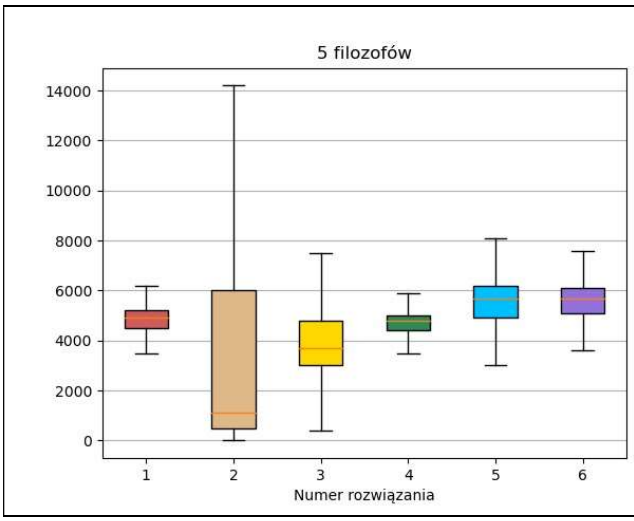
	Mediana [ns]		
	5 filozofów	20 filozofów	100 filozofów
Rozwiązanie 1	4,90E+03	4,80E+03	2,70E+03
Rozwiązanie 2	1,10E+03	3,00E+02	1,00E+02
Rozwiązanie 3	3,70E+03	3,30E+03	1,00E+03
Rozwiązanie 4	4,80E+03	4,70E+03	3,20E+03
Rozwiązanie 5	5,70E+03	3,90E+03	2,70E+03
Rozwiązanie 6	5,70E+03	3,90E+03	2,90E+03

Tabela VI

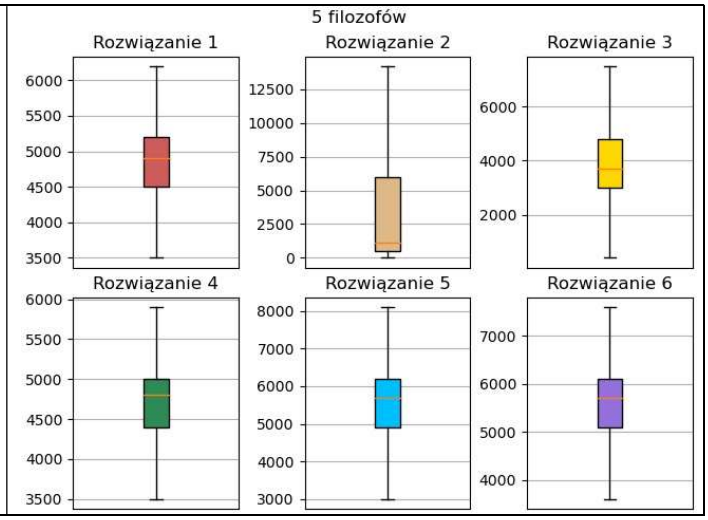
	Wartość maksymalna [ns]		
	5 filozofów	20 filozofów	100 filozofów
Rozwiązanie 1	9,71E+09	1,96E+10	1,98E+10
Rozwiązanie 2	3,22E+08	1,09E+10	1,96E+10
Rozwiązanie 3	3,54E+09	3,43E+09	1,82E+10
Rozwiązanie 4	1,90E+10	1,95E+10	1,98E+10
Rozwiązanie 5	1,04E+10	3,56E+08	1,24E+09
Rozwiązanie 6	4,69E+09	1,15E+10	1,96E+10

Tabela VII

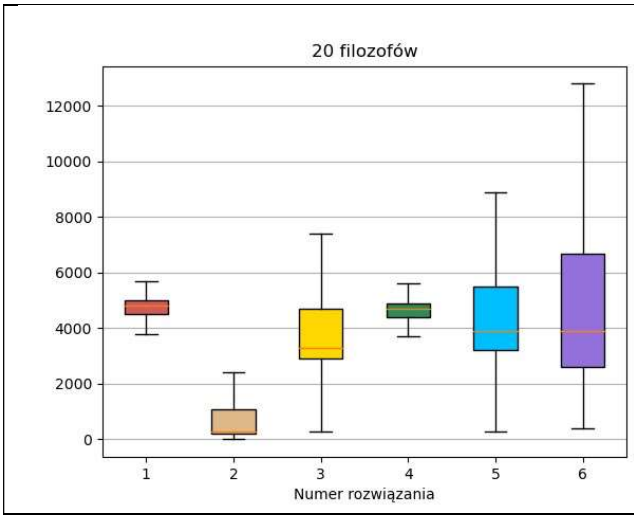
6.1.1. PORÓWNANIE DLA RÓŻNYCH ALGORYTMÓW



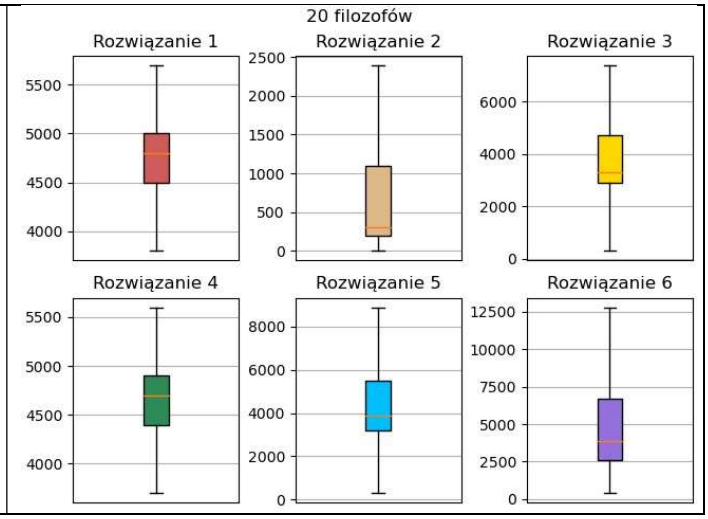
Wykres XXV



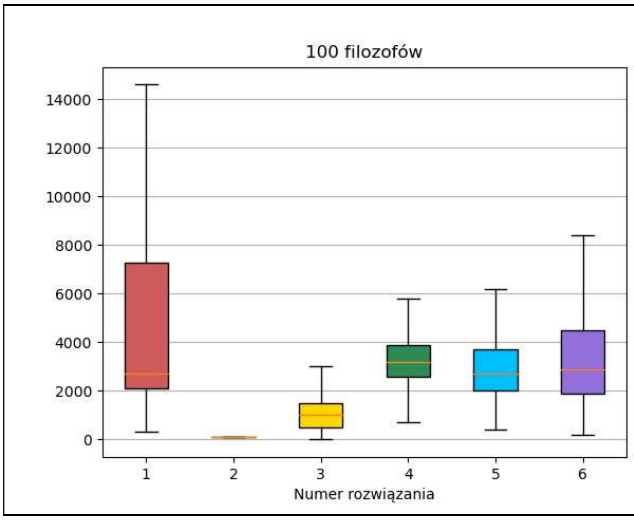
Wykres XXVI-XXXI



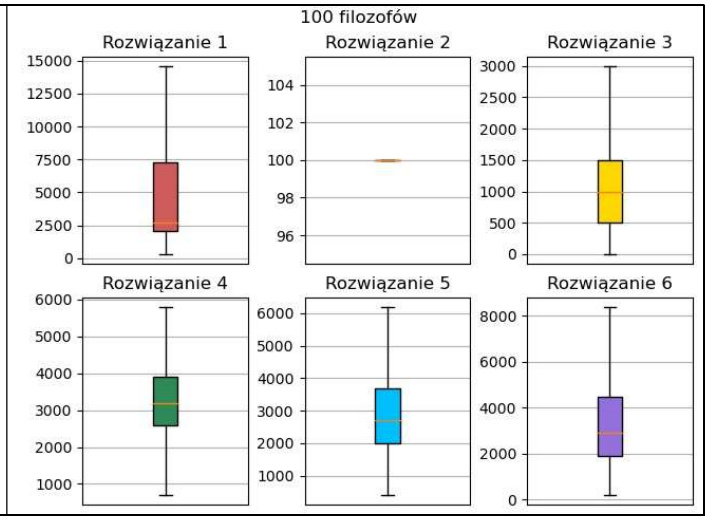
Wykres XXXII



Wykres XXXIII-XXXVIII

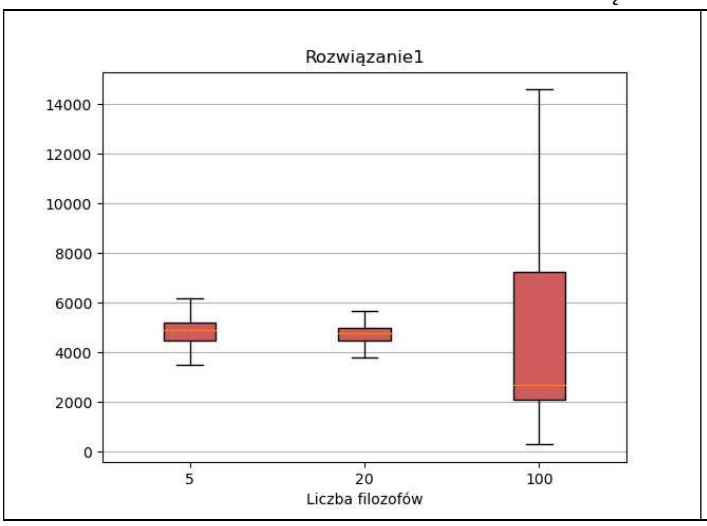


Wykres XXXIX

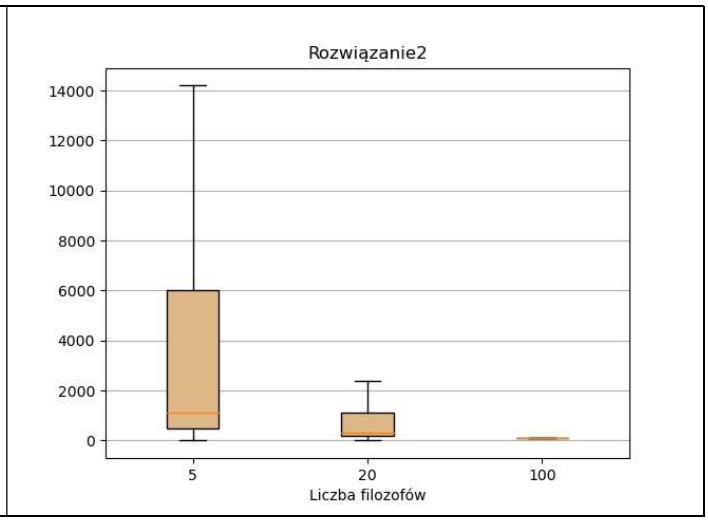


Wykres XL-XLV

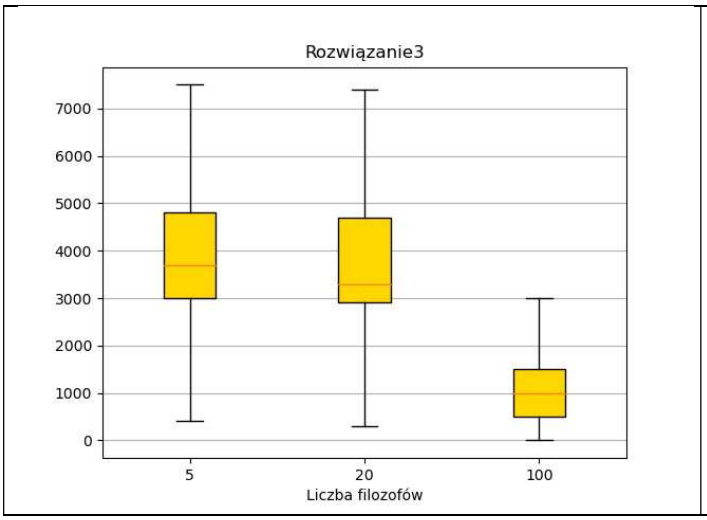
6.1.2. PORÓWNANIE DLA LICZBY UCZTUJĄCYCH FILOZOFÓW



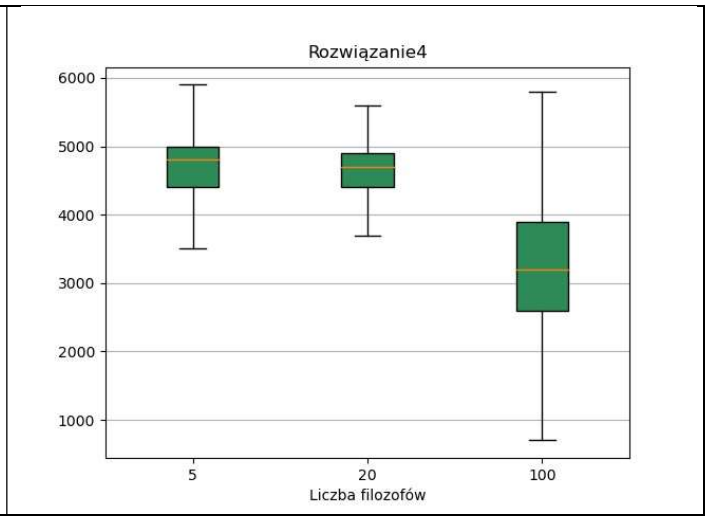
Wykres XLVI



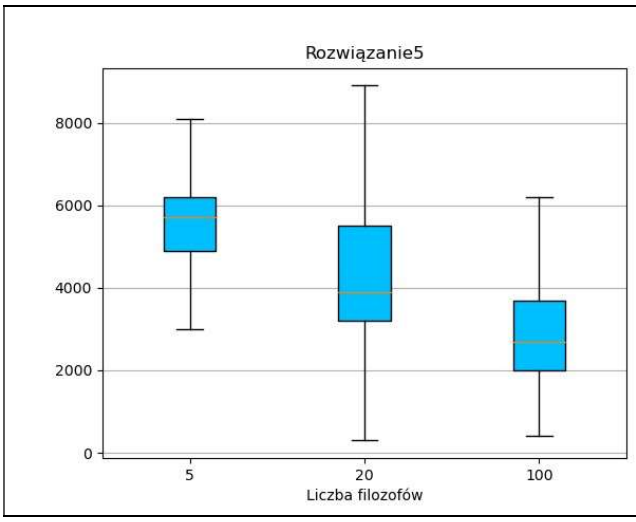
Wykres XLVII



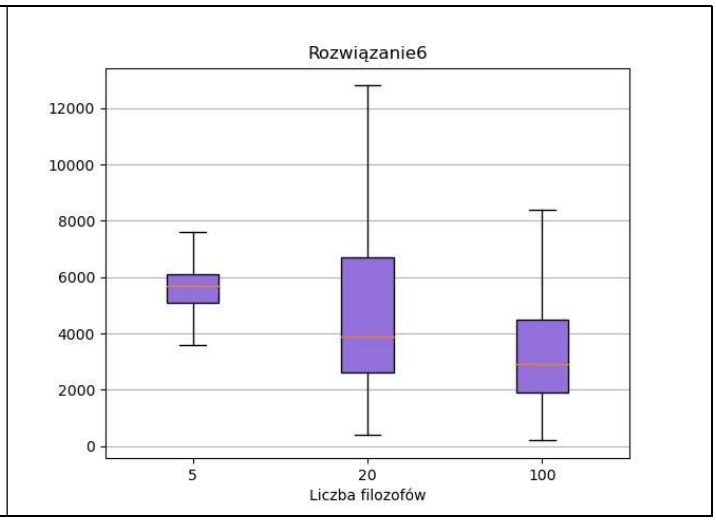
Wykres XLVIII



Wykres XLIX



Wykres XLX



Wykres LI

6.2. WNIOSKI

- W każdym testowanym przypadku mediana jest najniższa dla rozwiązania 2. Na wykresach dla rozwiązania 2 można zauważyć kilka ciekawych sytuacji.
  Dla 5 filozofów istnieje bardzo duża wariancja danych. Połowa danych znajduje się w niewielkiej części wykresu i osiąga niskie wartości, gdy druga polowa jest bardzo rozrzucona na osi. Wskazuje o tym kształt wykresu reprezentujący 3 i 4 kwartyl.
  Przeciwnie dla 100 filozofów, gdzie wariancja jest bardzo mała, a dane są skoncentrowane w jednym punkcie.
- Dla wszystkich testów średnia jest zdecydowanie większa niż mediana, co wskazuje na to, że istnieje sporo wartości odstających. Ten sam wniosek można sformułować obserwując wąsy na przedstawionych wykresach.
- Największą medianę ma zazwyczaj rozwiązanie 1, jednak nie jest to zasadą.
- Dobre wyniki są osiągane dla rozwiązania 3, gdzie zazwyczaj mediana osiąga jedną z najniższych wartości w porównaniu do innych algorytmów.
- Na podstawie wykresów XLVI-LI można stwierdzić, że nie jest prawdą, że czas oczekiwania jednoznacznie się zwiększa lub zmniejsza wraz z wzrostem liczby wątków.
- Na podstawie tych wykresów ciężko jest stwierdzić czy któryś z wątków ma przewagę nad innym.

7. PODSUMOWANIE

Przeprowadzone doświadczenia nie doprowadziły mnie do satysfakcjonujących wniosków. Nie jestem w stanie jednoznacznie stwierdzić, który z algorytmów działa najlepiej. W zestawieniu algorytm 1 zazwyczaj radził sobie najgorzej. Również pozostałe algorytmy miały przypadki działania poniżej oczekiwań. W danych można zauważyć wiele ciekawych odstających sytuacji. Prowadzi to do spostrzeżenia, że w tym który wątek otrzyma dostęp do zasobu i jak szybko, istnieje pewna doza losowości.