

CENTRAL WASHINGTON UNIVERSITY

CS 471 OPTIMIZATION

SPRING 2019

Project 5 Report

Author:

Hermann YEPDJI

Supervisor:

Dr. Donald DAVENDRA

May 31, 2019



Contents

1	Introduction	2
2	Results	3
2.1	Flow Shop Scheduling	3
2.1.1	Gantt Charts	3
2.1.2	Results for Flow Shop Scheduling	14
2.1.3	Analysis	17
2.2	Flow Shop Scheduling With Blocking	18
2.2.1	Gantt Charts	18
2.2.2	Results for Flow Shop Scheduling with Blocking	29
2.2.3	Analysis	32
2.3	Flow Shop Scheduling with No Wait	33
2.3.1	Gantt Charts	33
2.3.2	Results for Flow Shop Scheduling with No Wait	44
2.3.3	Analysis	47
3	Conclusion	48

1 Introduction

Project 5 was about optimizing 3 standard job scheduling functions namely, Flow Shop Scheduling (FSS), Flow Shop Scheduling with Blocking(FSSB) and Flow Shop Scheduling with No Wait (FSSNW). For this purpose, one optimization algorithm was to be implemented then applied to those functions. The algorithm is called NEH (Nawaz Enscore Ham) heuristic. After being implemented, the it was run on each of the 3 functions using as input, some data that was provided along with the project in the form of text files. Important details were recorded during the experimentation process then stored in a tabular format and they will be discussed then analyzed later on in this report. However, in order to visually observe how the jobs should be scheduled according to the NEH heuristic, Gantt charts were generated for some of the results and they are also discussed later on in this report.

2 Results

2.1 Flow Shop Scheduling

2.1.1 Gantt Charts

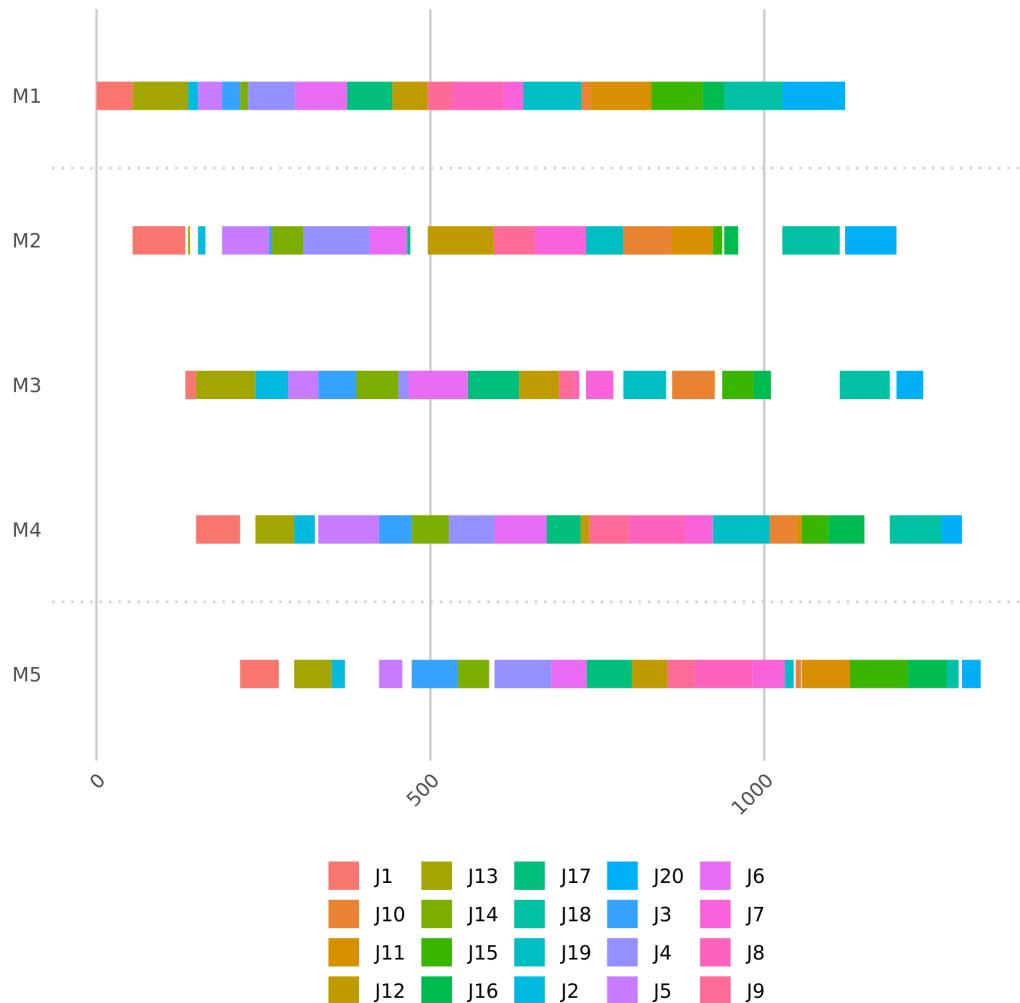


Figure 1: Gantt Chart for file 1.txt

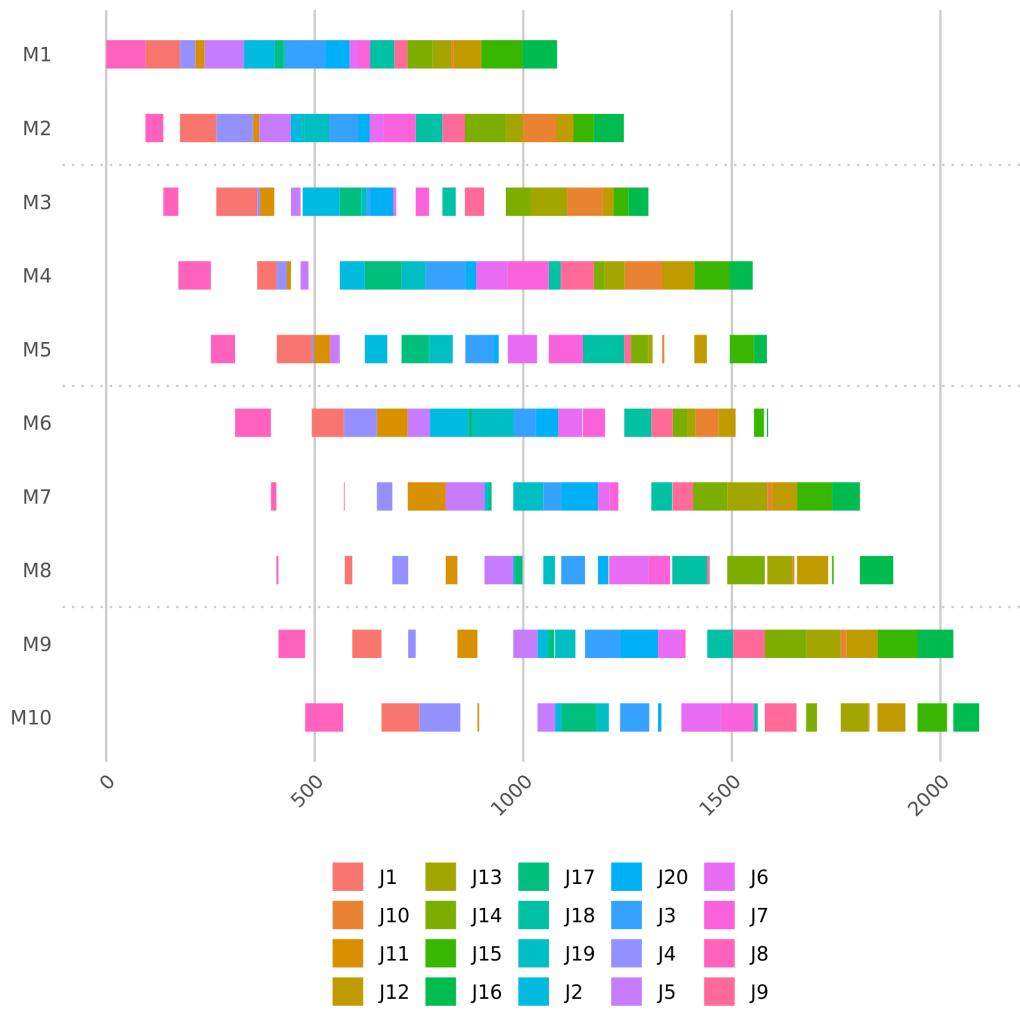


Figure 2: Gantt Chart for file 11.txt

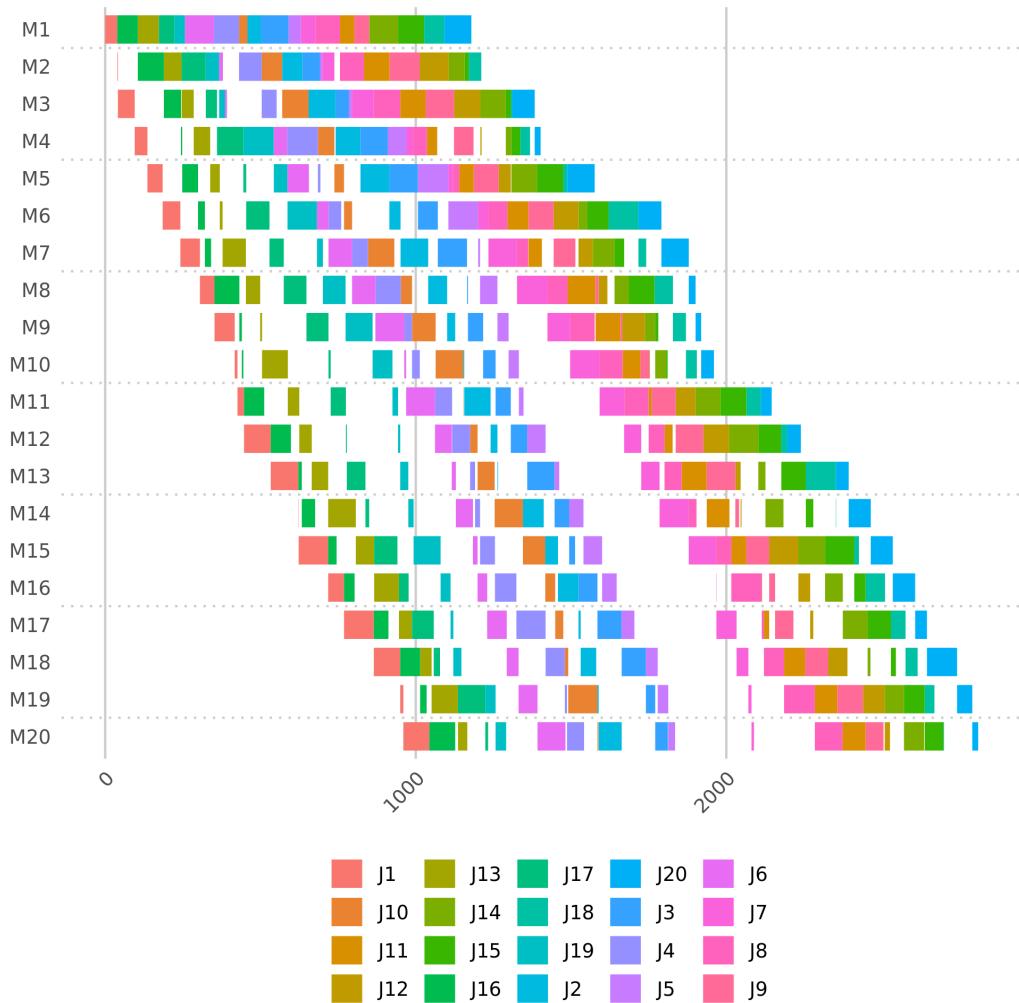


Figure 3: Gantt Chart for file 21.txt

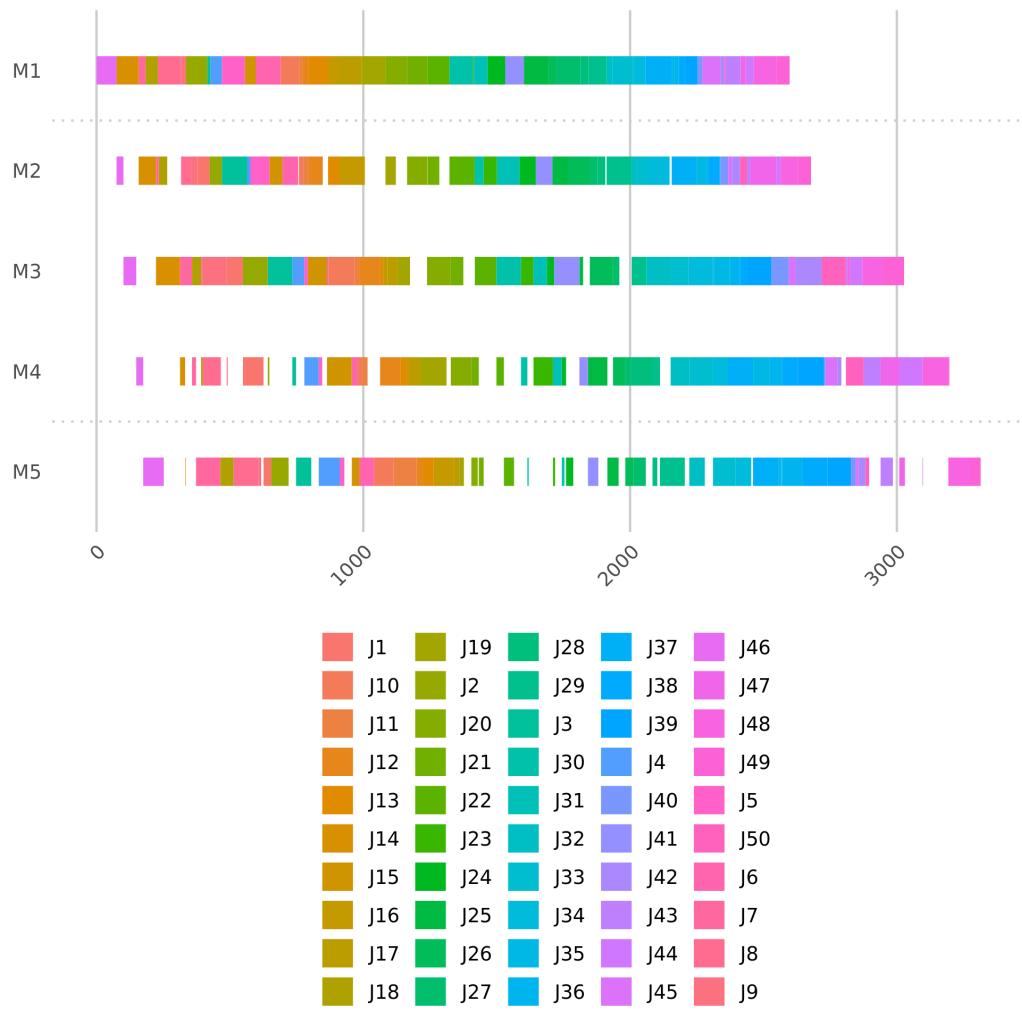


Figure 4: Gantt Chart for file 31.txt

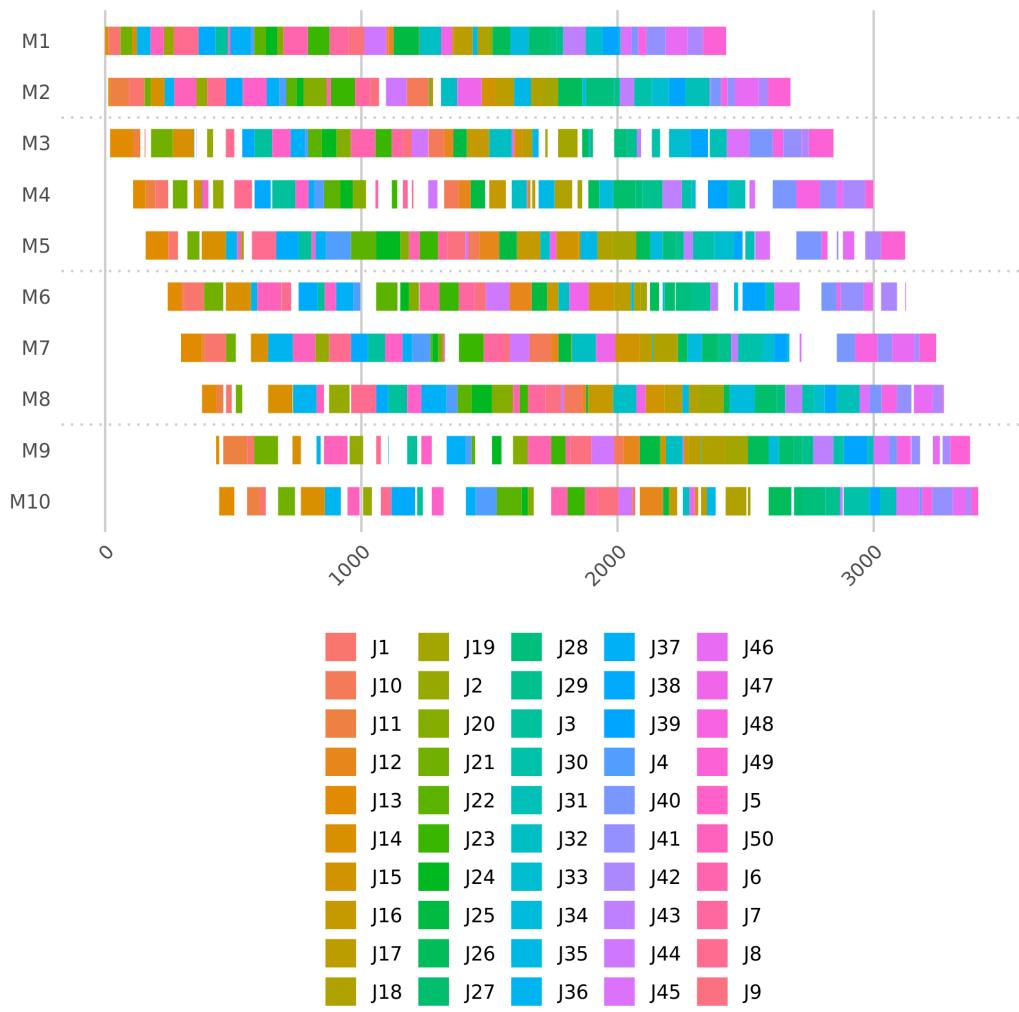


Figure 5: Gantt Chart for file 41.txt

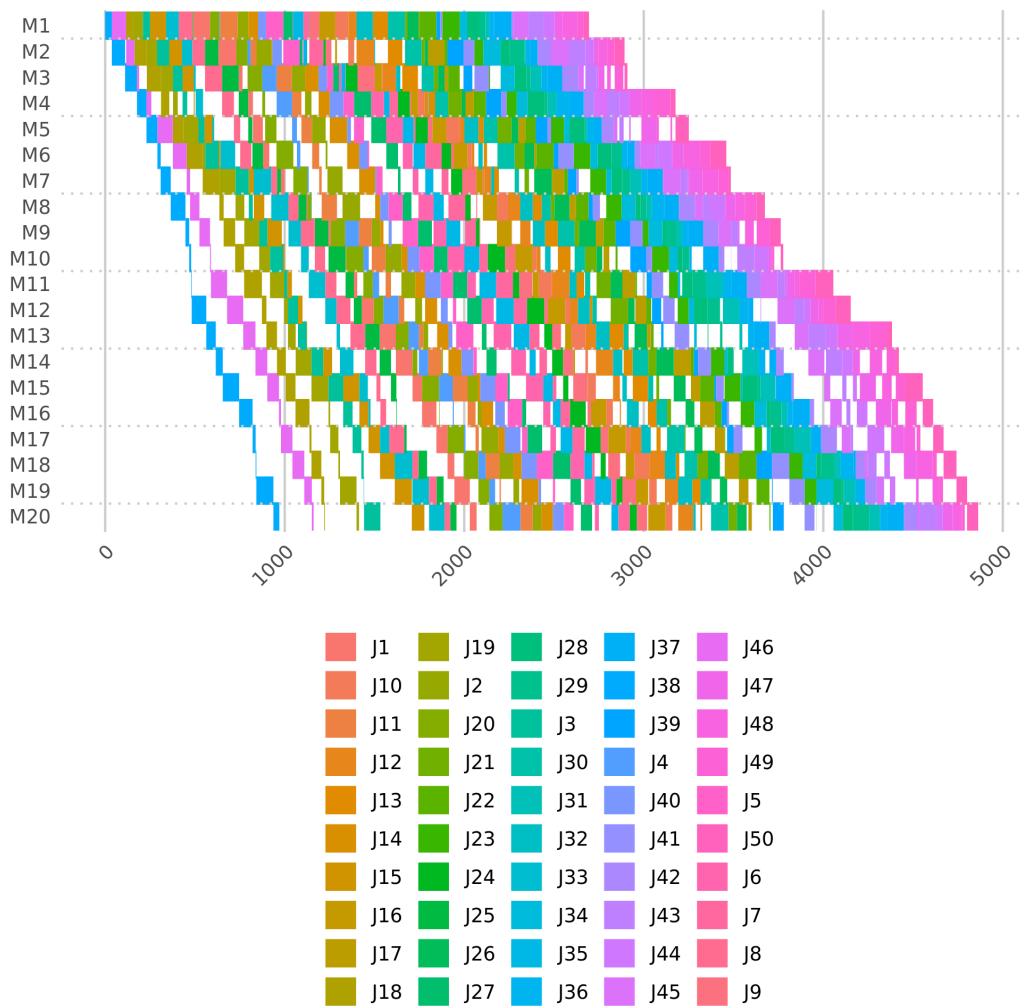


Figure 6: Gantt Chart for file 51.txt

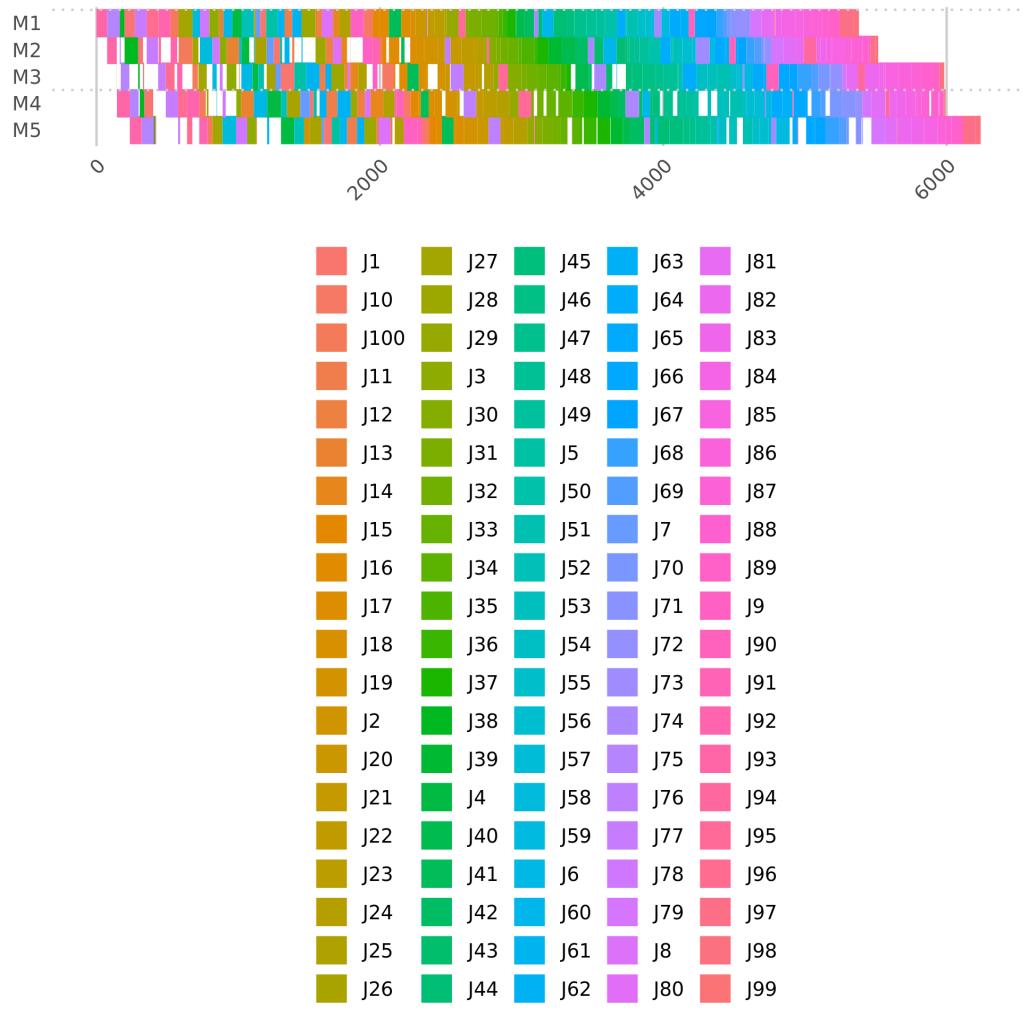


Figure 7: Gantt Chart for file 61.txt

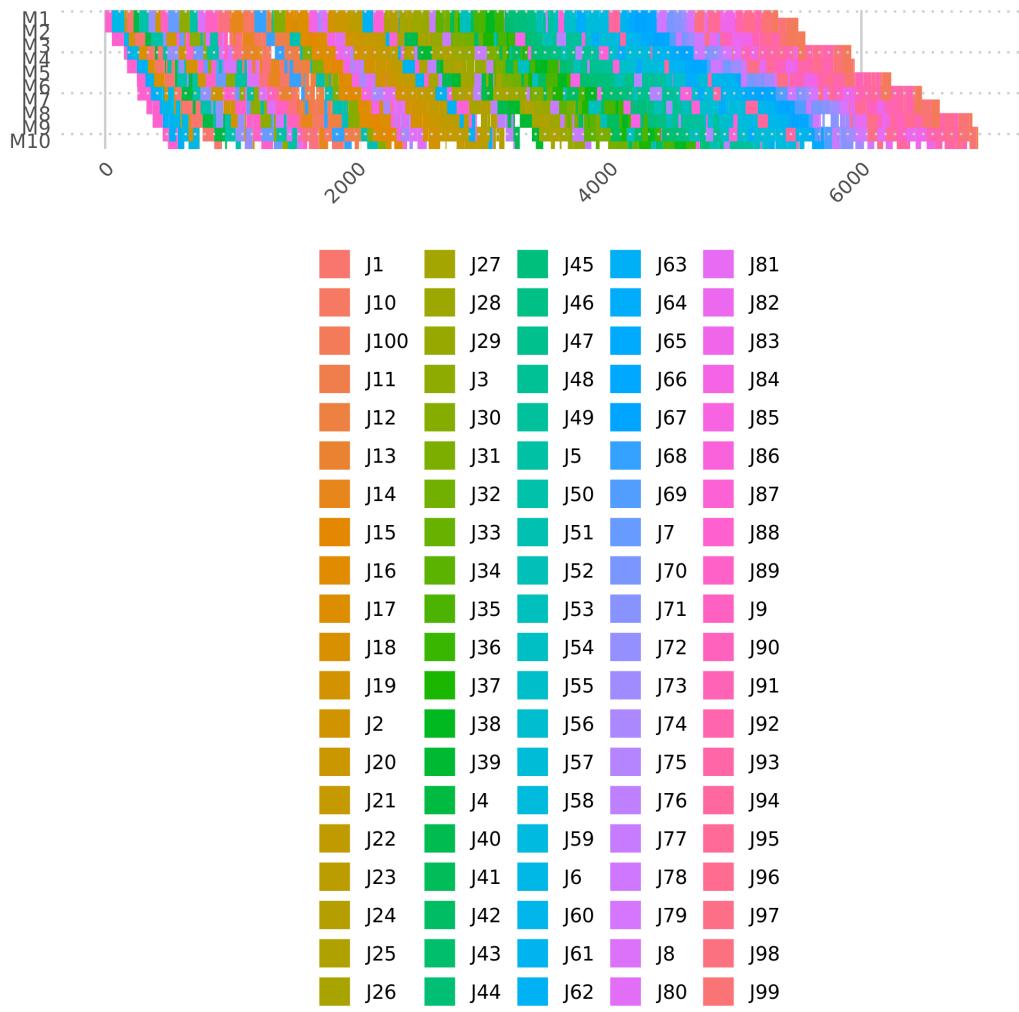


Figure 8: Gantt Chart for file 71.txt

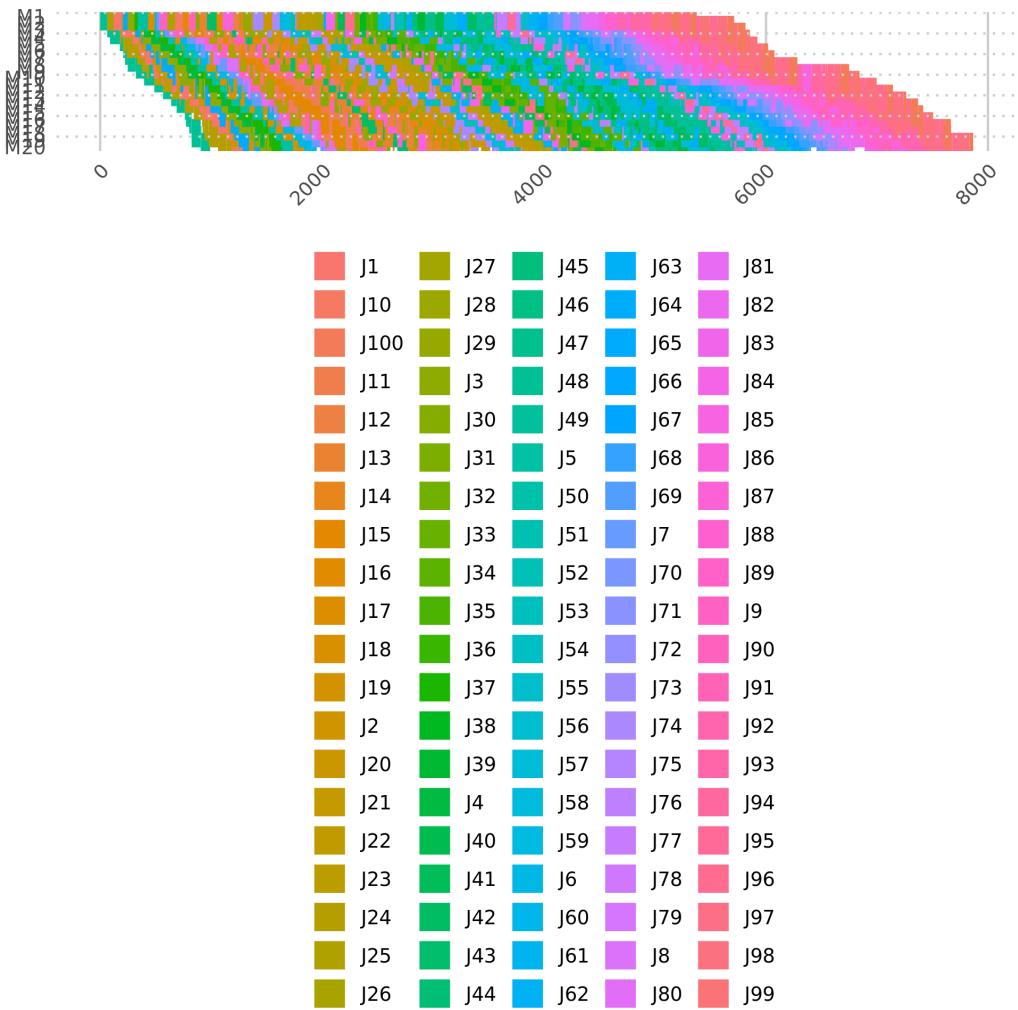


Figure 9: Gantt Chart for file 81.txt

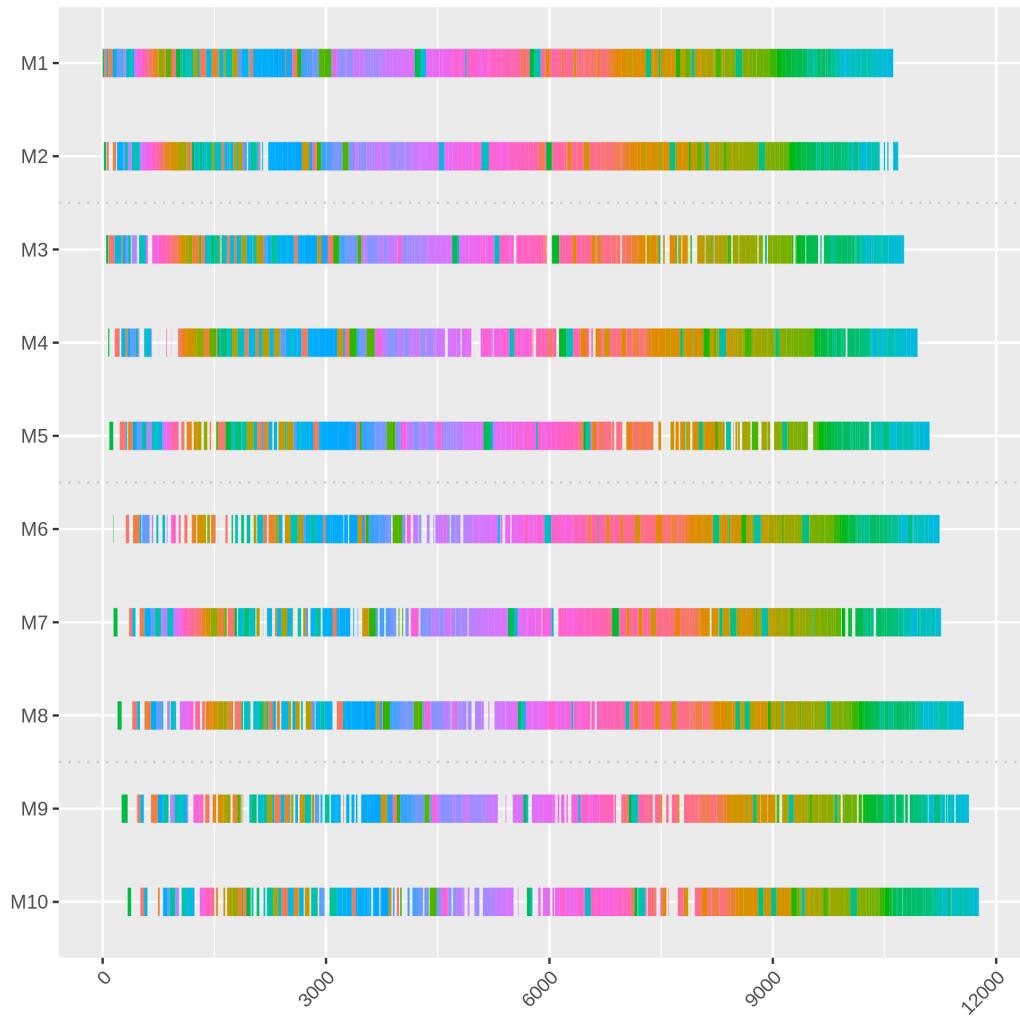


Figure 10: Gantt Chart for file 91.txt

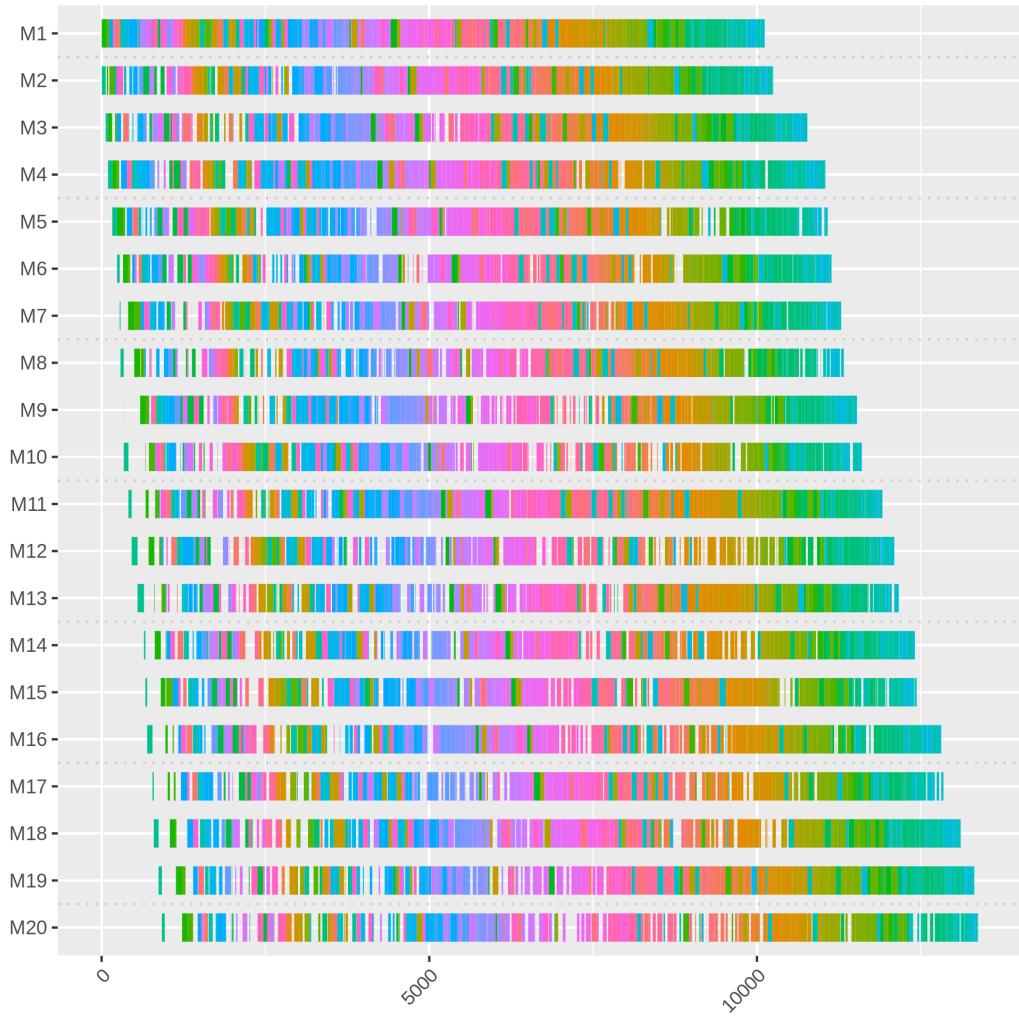


Figure 11: Gantt Chart for file 101.txt

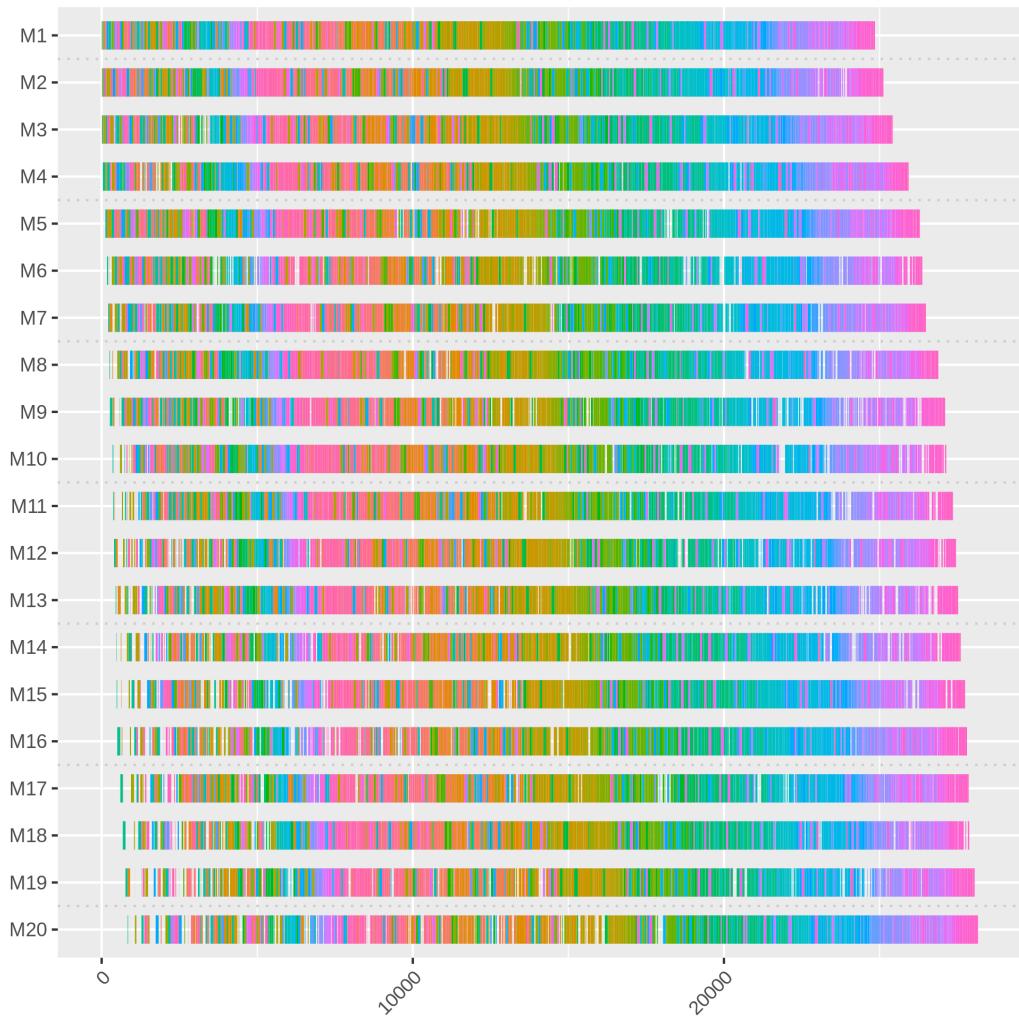


Figure 12: Gantt Chart for file 111.txt

2.1.2 Results for Flow Shop Scheduling

	File.name	Best.makespan	# func calls
1/DataFiles/1.txt	1324	558
2/DataFiles/10.txt	1250	562
3/DataFiles/2.txt	1396	566
4/DataFiles/3.txt	1197	565
5/DataFiles/4.txt	1449	569
6/DataFiles/5.txt	1360	568
7/DataFiles/6.txt	1347	569
8/DataFiles/7.txt	1286	562
9/DataFiles/8.txt	1380	565
10/DataFiles/9.txt	1353	567
11/DataFiles/11.txt	1714	564
12/DataFiles/12.txt	1806	567
13/DataFiles/13.txt	1684	563
14/DataFiles/14.txt	1606	570
15/DataFiles/15.txt	1638	569
16/DataFiles/16.txt	1624	569
17/DataFiles/17.txt	1611	566
18/DataFiles/18.txt	1786	565
19/DataFiles/19.txt	1748	570
20/DataFiles/20.txt	1827	565
21/DataFiles/21.txt	2502	570
22/DataFiles/22.txt	2392	567
23/DataFiles/23.txt	2532	566
24/DataFiles/24.txt	2461	567
25/DataFiles/25.txt	2537	562
26/DataFiles/26.txt	2436	565
27/DataFiles/27.txt	2464	569
28/DataFiles/28.txt	2392	567
29/DataFiles/29.txt	2486	568
30/DataFiles/30.txt	2394	568
31/DataFiles/31.txt	2859	3668
32/DataFiles/32.txt	2942	3645
33/DataFiles/33.txt	2743	3644
34/DataFiles/34.txt	2931	3664
35/DataFiles/35.txt	2980	3656
36/DataFiles/36.txt	2987	3650
37/DataFiles/37.txt	2845	3657
38/DataFiles/38.txt	2866	3647
39/DataFiles/39.txt	2653	3642
40/DataFiles/40.txt	2822	3597
41/DataFiles/41.txt	3408	3657
42/DataFiles/42.txt	3445	3665
43/DataFiles/43.txt	3253	3662
44/DataFiles/44.txt	3365	3665
45/DataFiles/45.txt	3396	3668

	File.name	Best.makespan	# func calls
1/DataFiles/46.txt	3365	3659
2/DataFiles/47.txt	3495	3656
3/DataFiles/48.txt	3309	3659
4/DataFiles/49.txt	3176	3664
5/DataFiles/50.txt	3476	3645
6/DataFiles/51.txt	4285	3663
7/DataFiles/52.txt	4172	3659
8/DataFiles/53.txt	4130	3649
9/DataFiles/54.txt	4225	3648
10/DataFiles/55.txt	4163	3659
11/DataFiles/56.txt	4228	3661
12/DataFiles/57.txt	4262	3655
13/DataFiles/58.txt	4213	3660
14/DataFiles/59.txt	4141	3648
15/DataFiles/60.txt	4218	3651
16/DataFiles/61.txt	5660	14801
17/DataFiles/62.txt	5345	14799
18/DataFiles/63.txt	5455	14801
19/DataFiles/64.txt	5205	14794
20/DataFiles/65.txt	5486	14794
21/DataFiles/66.txt	5314	14796
22/DataFiles/67.txt	5530	14795
23/DataFiles/68.txt	5179	14813
24/DataFiles/69.txt	5518	14782
25/DataFiles/70.txt	5459	14810
26/DataFiles/71.txt	6285	14792
27/DataFiles/72.txt	5822	14815
28/DataFiles/73.txt	6178	14803
29/DataFiles/74.txt	6311	14817
30/DataFiles/75.txt	5956	14812
31/DataFiles/76.txt	5774	14824
32/DataFiles/77.txt	5869	14815
33/DataFiles/78.txt	6105	14799
34/DataFiles/79.txt	6209	14801
35/DataFiles/80.txt	6273	14801
36/DataFiles/81.txt	6954	14749
37/DataFiles/82.txt	7138	14733
38/DataFiles/83.txt	6978	14740
39/DataFiles/84.txt	6882	14717
40/DataFiles/85.txt	7168	14733
41/DataFiles/86.txt	7283	14724
42/DataFiles/87.txt	6957	14749
43/DataFiles/88.txt	7350	14702
44/DataFiles/89.txt	16 7098	14723
45/DataFiles/90.txt	7217	14661
46/DataFiles/100.txt	11336	59401

	File.name	Best.makespan	# func calls
1/DataFiles/91.txt	11542	59416
2/DataFiles/92.txt	11290	59441
3/DataFiles/93.txt	11731	59253
4/DataFiles/94.txt	11222	59446
5/DataFiles/95.txt	11217	59494
6/DataFiles/96.txt	11166	59382
7/DataFiles/97.txt	11302	59553
8/DataFiles/98.txt	11465	59313
9/DataFiles/99.txt	11178	59426
10/DataFiles/101.txt	12338	58906
11/DataFiles/102.txt	12498	59106
12/DataFiles/103.txt	12678	59219
13/DataFiles/104.txt	12417	59099
14/DataFiles/105.txt	12501	58971
15/DataFiles/106.txt	12356	58916
16/DataFiles/107.txt	12658	58986
17/DataFiles/108.txt	12678	59048
18/DataFiles/109.txt	12465	59170
19/DataFiles/110.txt	12498	59202
20/DataFiles/111.txt	28166	371362
21/DataFiles/112.txt	28615	371897
22/DataFiles/113.txt	28151	371500
23/DataFiles/114.txt	28286	371772
24/DataFiles/115.txt	28567	371442
25/DataFiles/116.txt	28718	371436
26/DataFiles/117.txt	28195	371245
27/DataFiles/118.txt	28803	370891
28/DataFiles/119.txt	27873	371911
29/DataFiles/120.txt	28385	371341

Table 1: Results for Flow Shop Scheduling

2.1.3 Analysis

As seen from table1 above, both the makespan and the number of function calls increases as the number of machines or the number of jobs increases. The table also shows that the best makespan found are not really close to what is reported on Taillard website. For example the table shows that the best makespan found for file 54.txt is 3723

2.2 Flow Shop Scheduling With Blocking

2.2.1 Gantt Charts

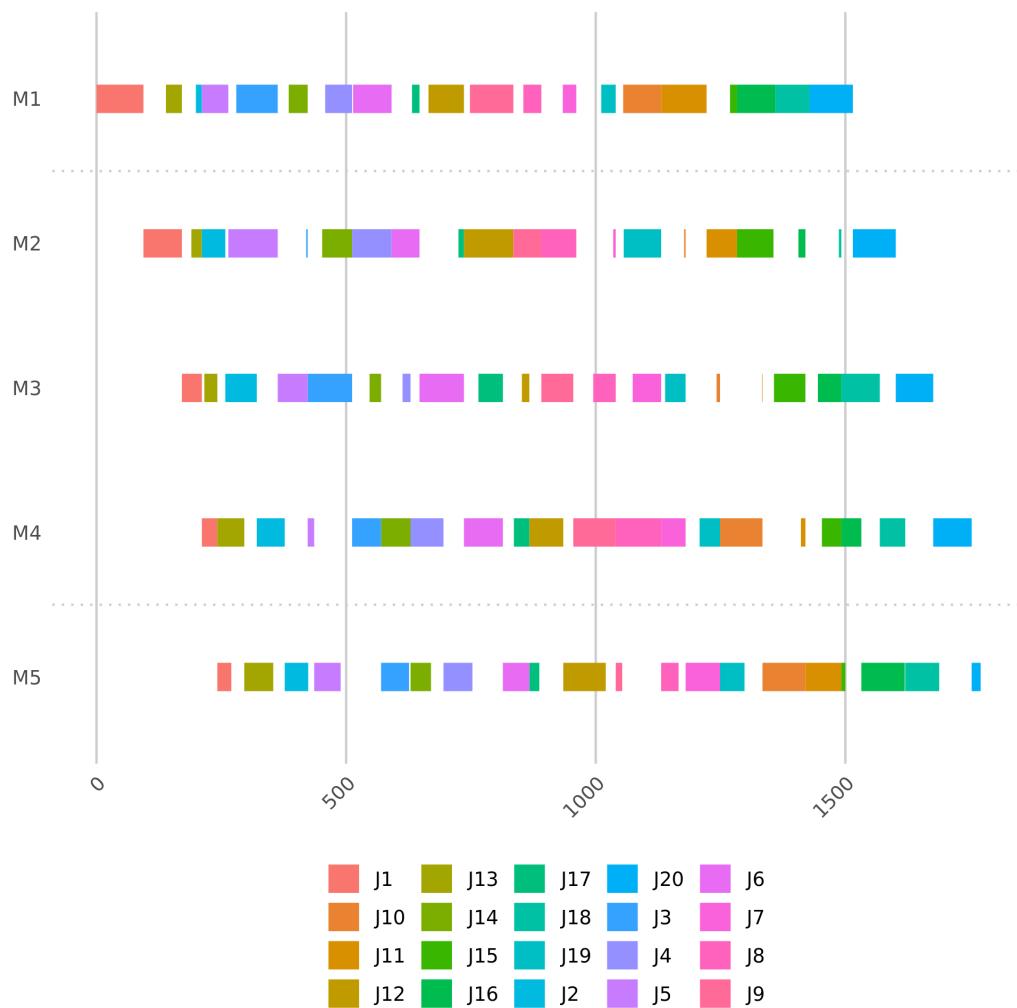


Figure 13: Gantt Chart for file 1.txt

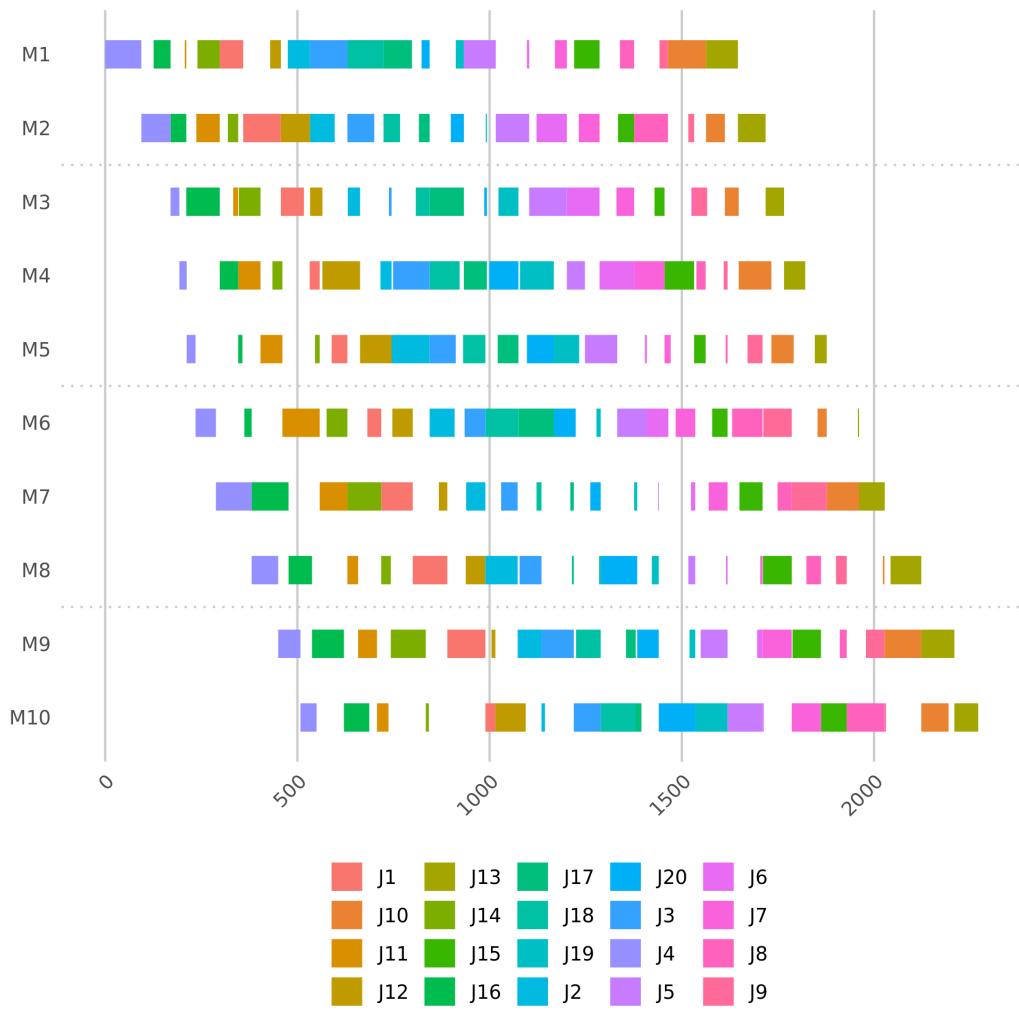


Figure 14: Gantt Chart for file 11.txt

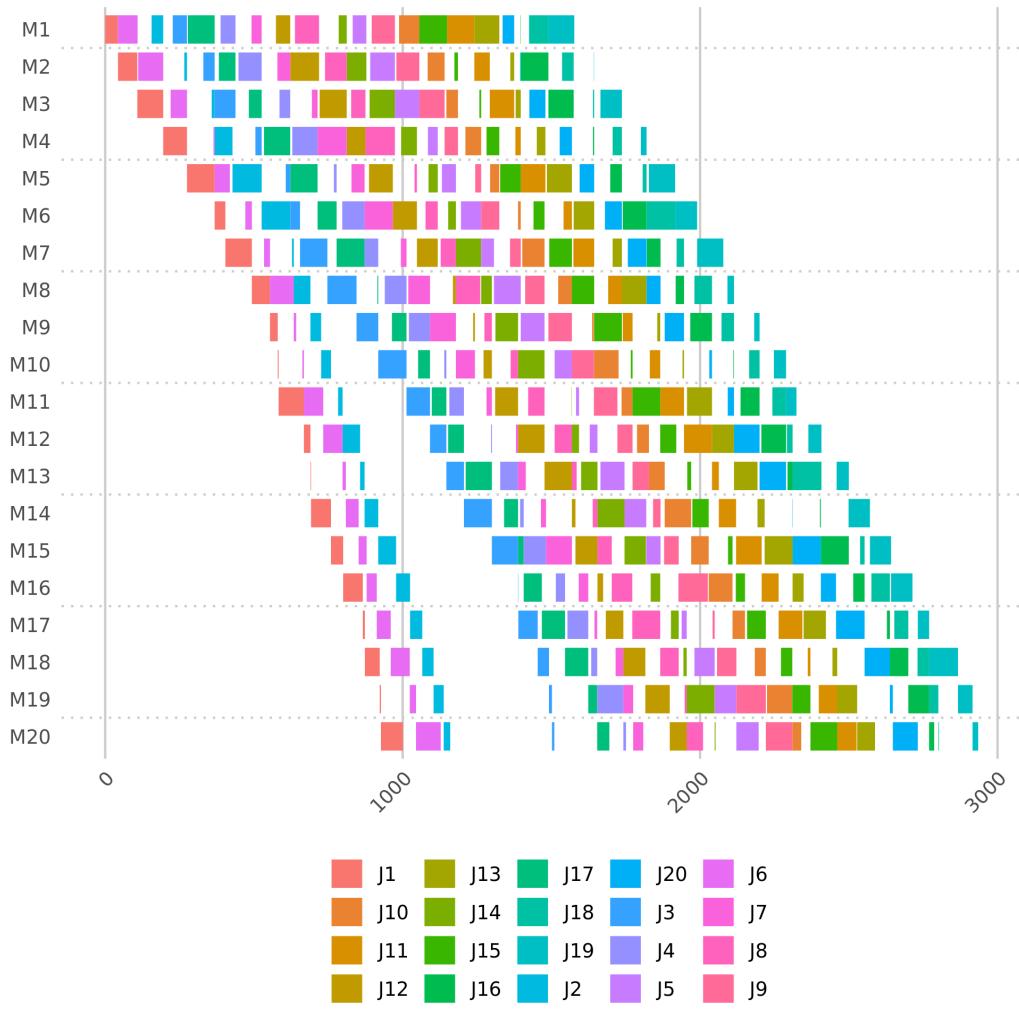


Figure 15: Gantt Chart for file 21.txt

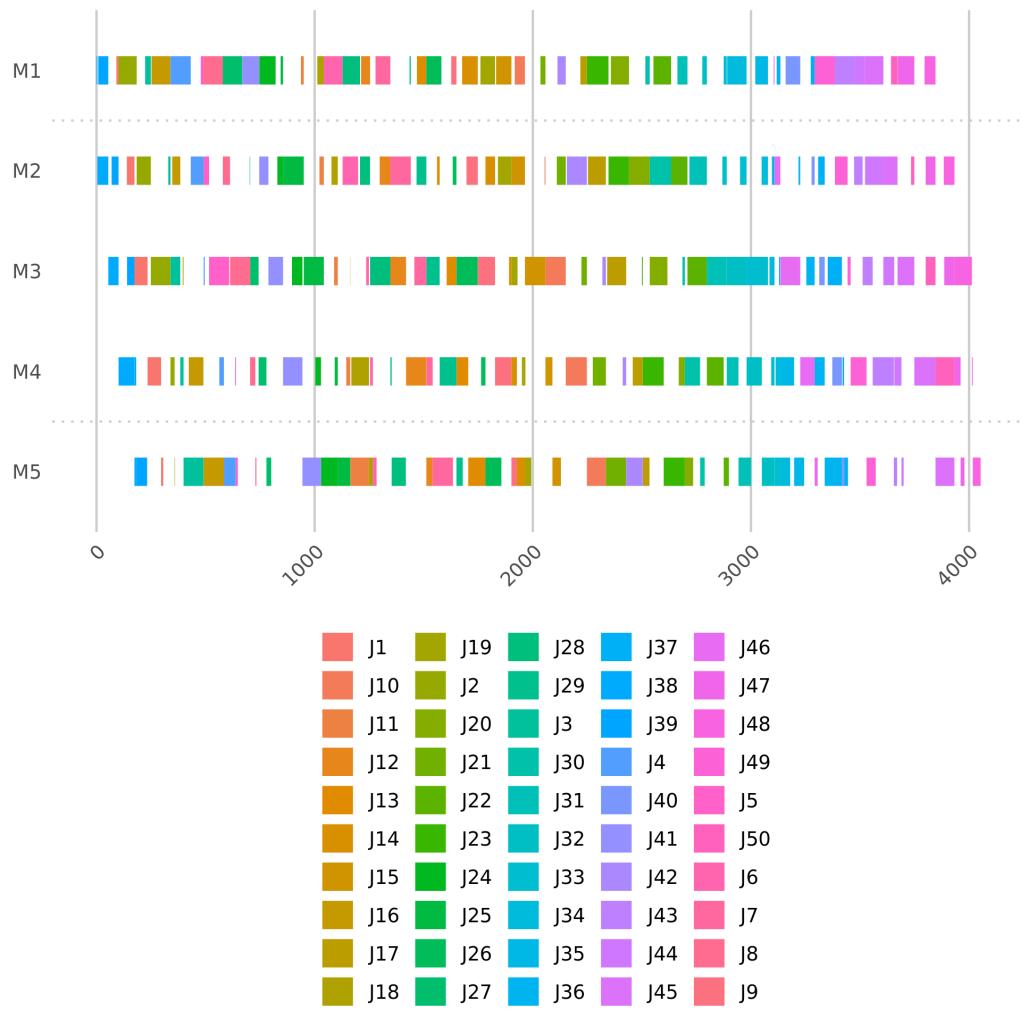


Figure 16: Gantt Chart for file 31.txt



Figure 17: Gantt Chart for file 41.txt

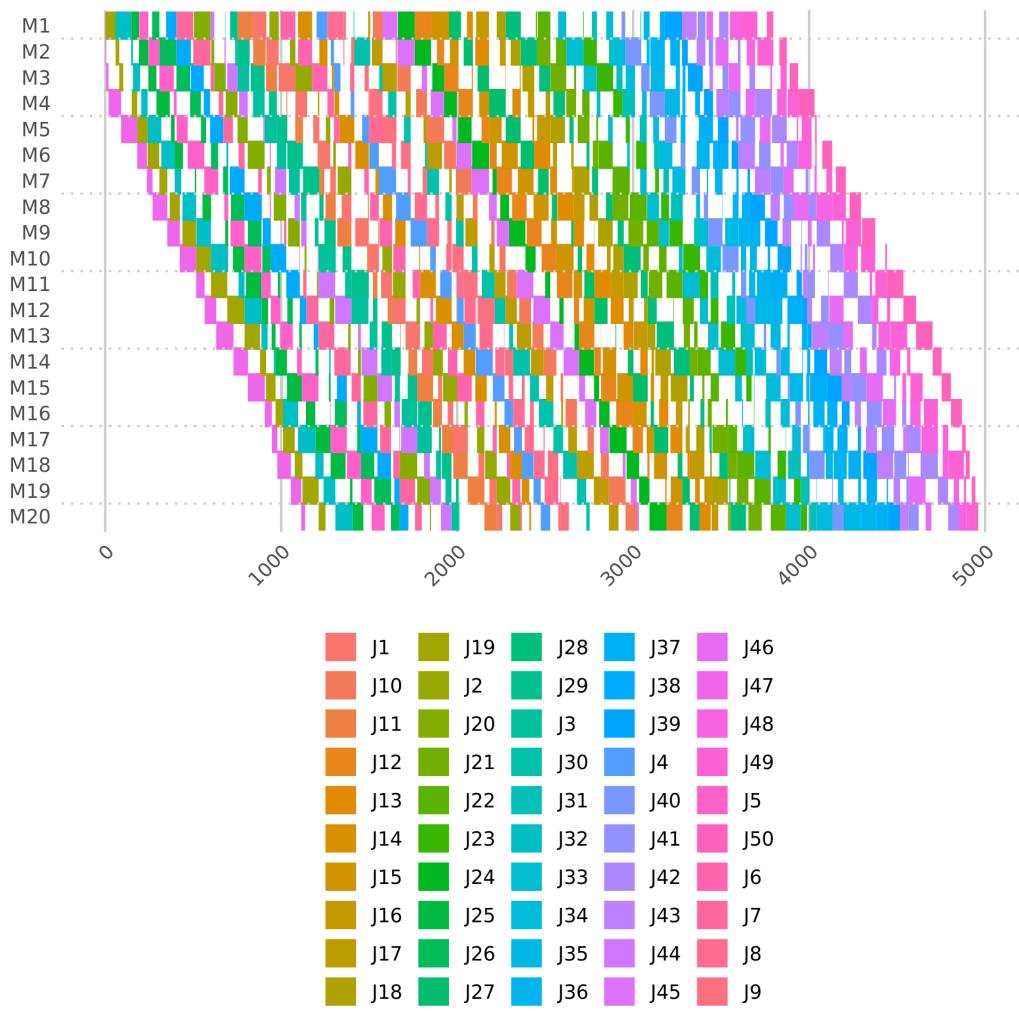


Figure 18: Gantt Chart for file 51.txt

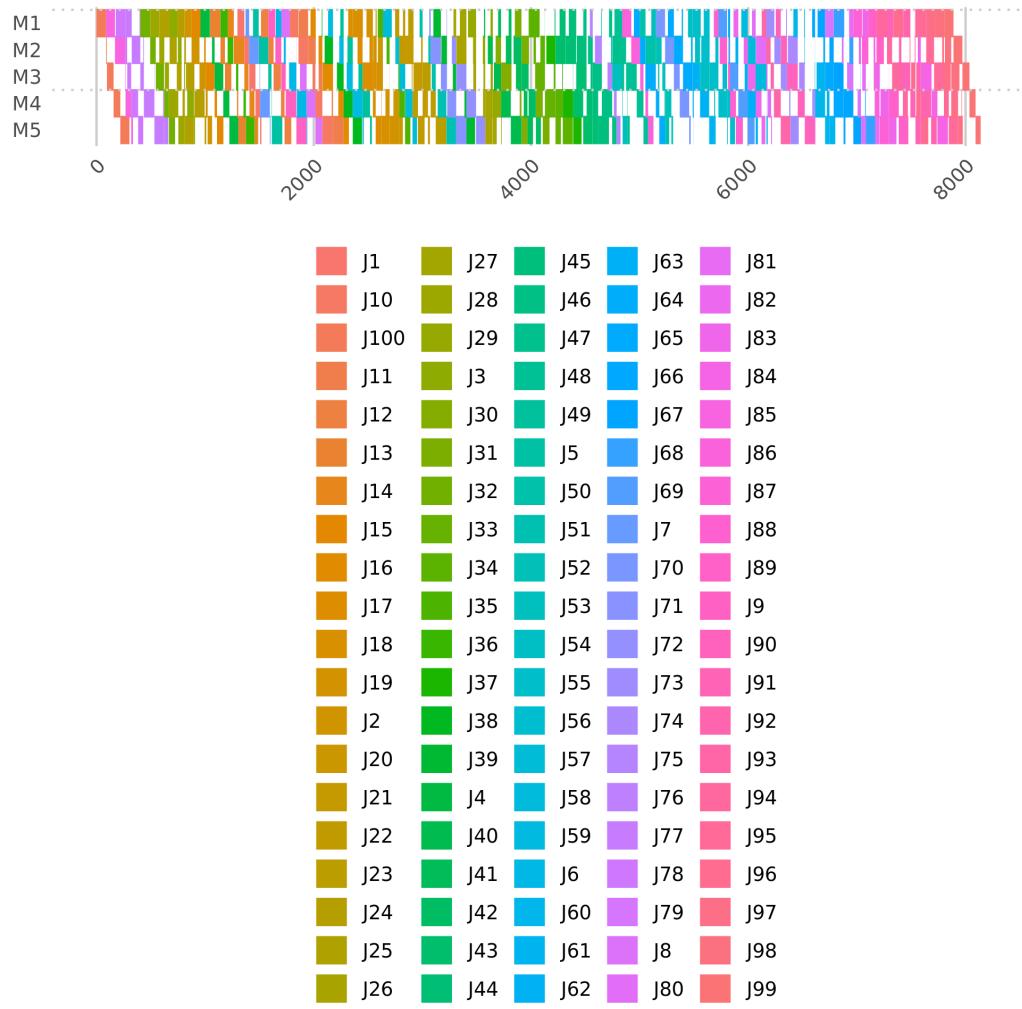


Figure 19: Gantt Chart for file 61.txt

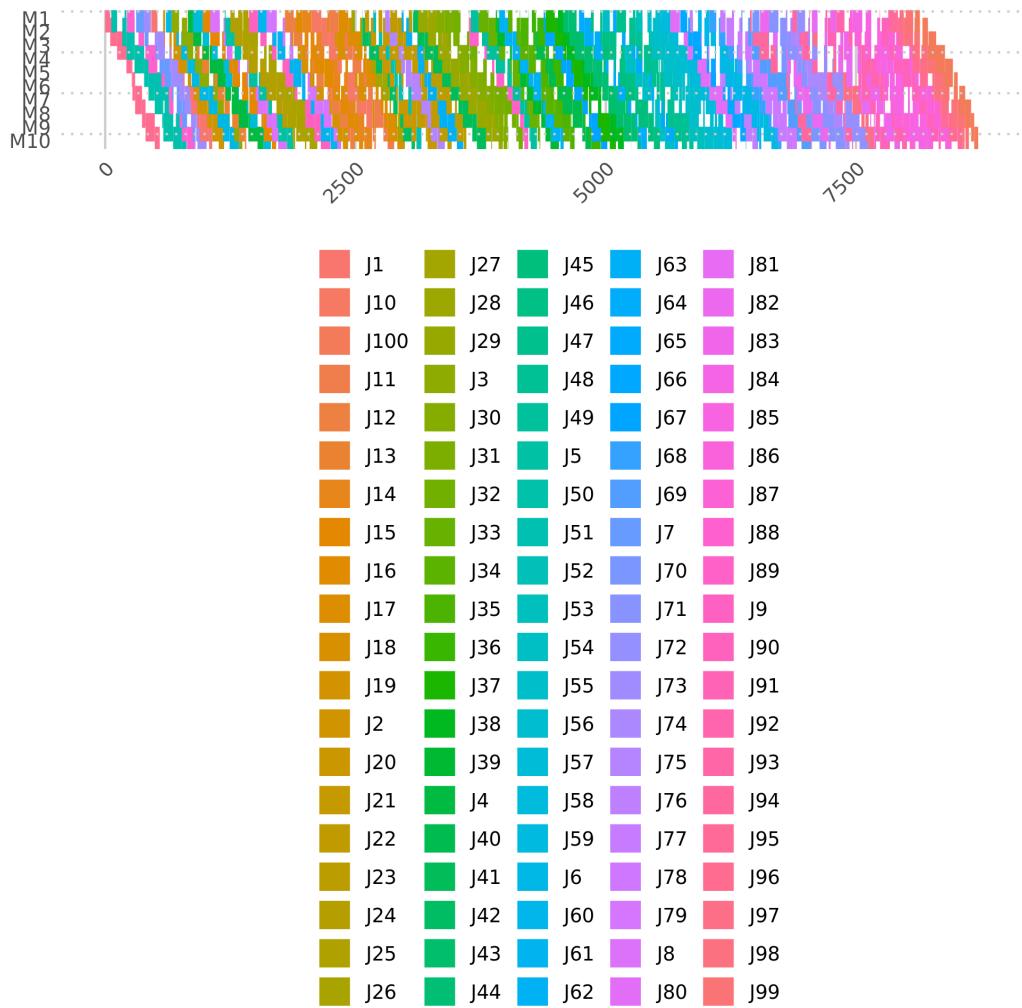


Figure 20: Gantt Chart for file 71.txt

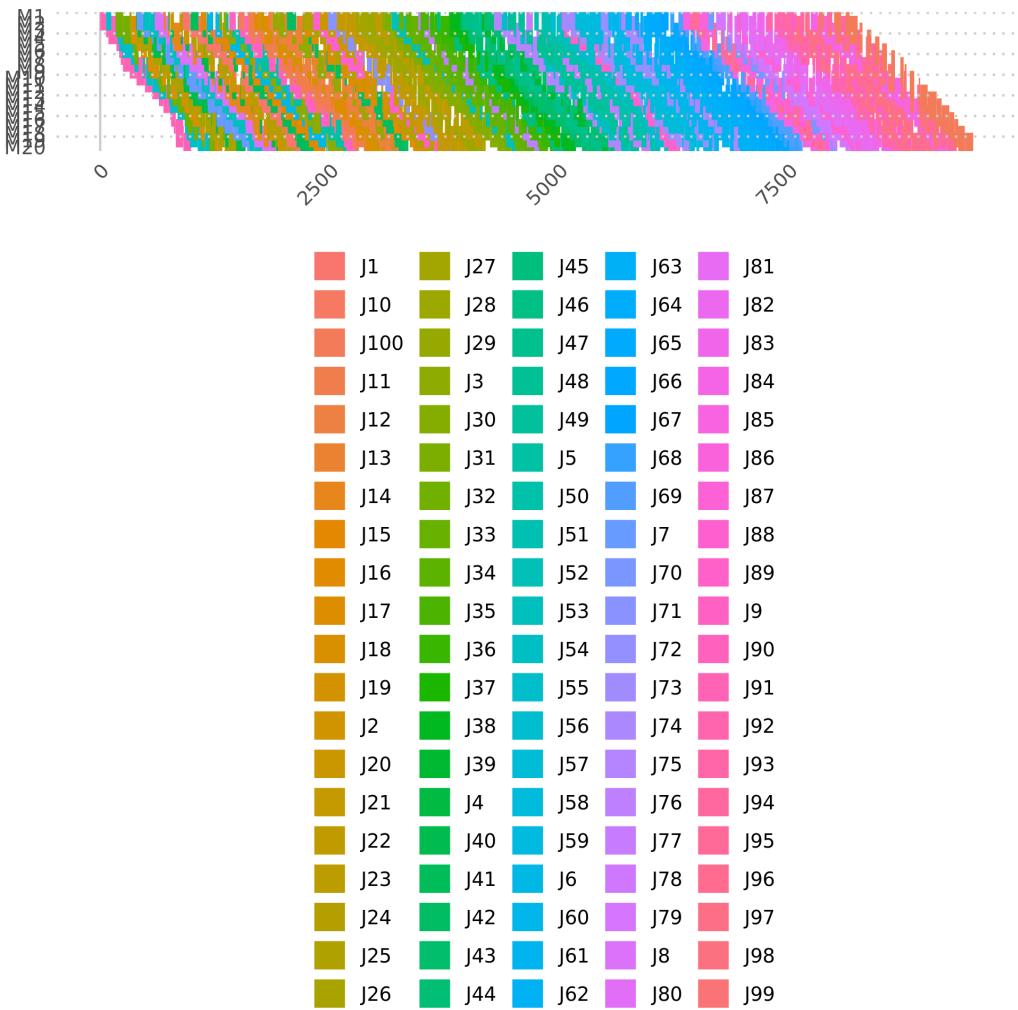


Figure 21: Gantt Chart for file 81.txt

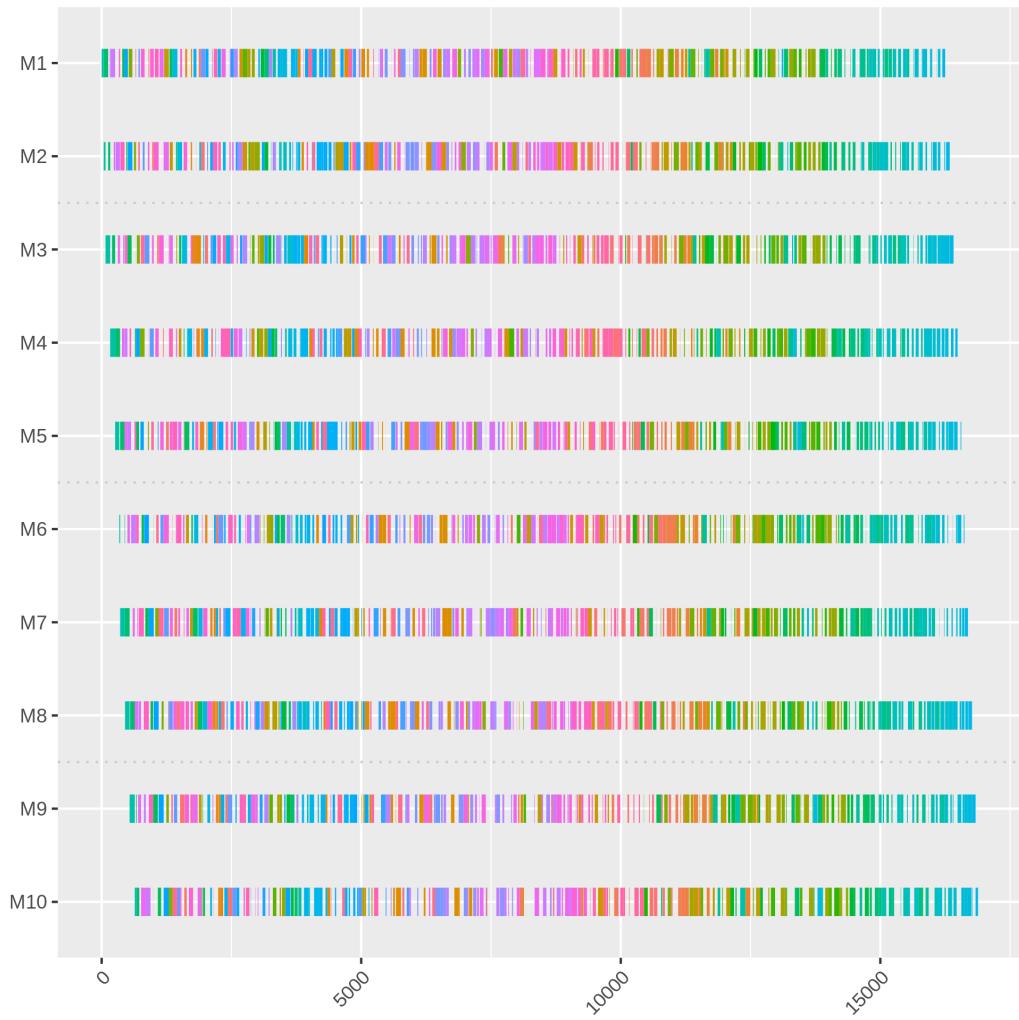


Figure 22: Gantt Chart for file 91.txt

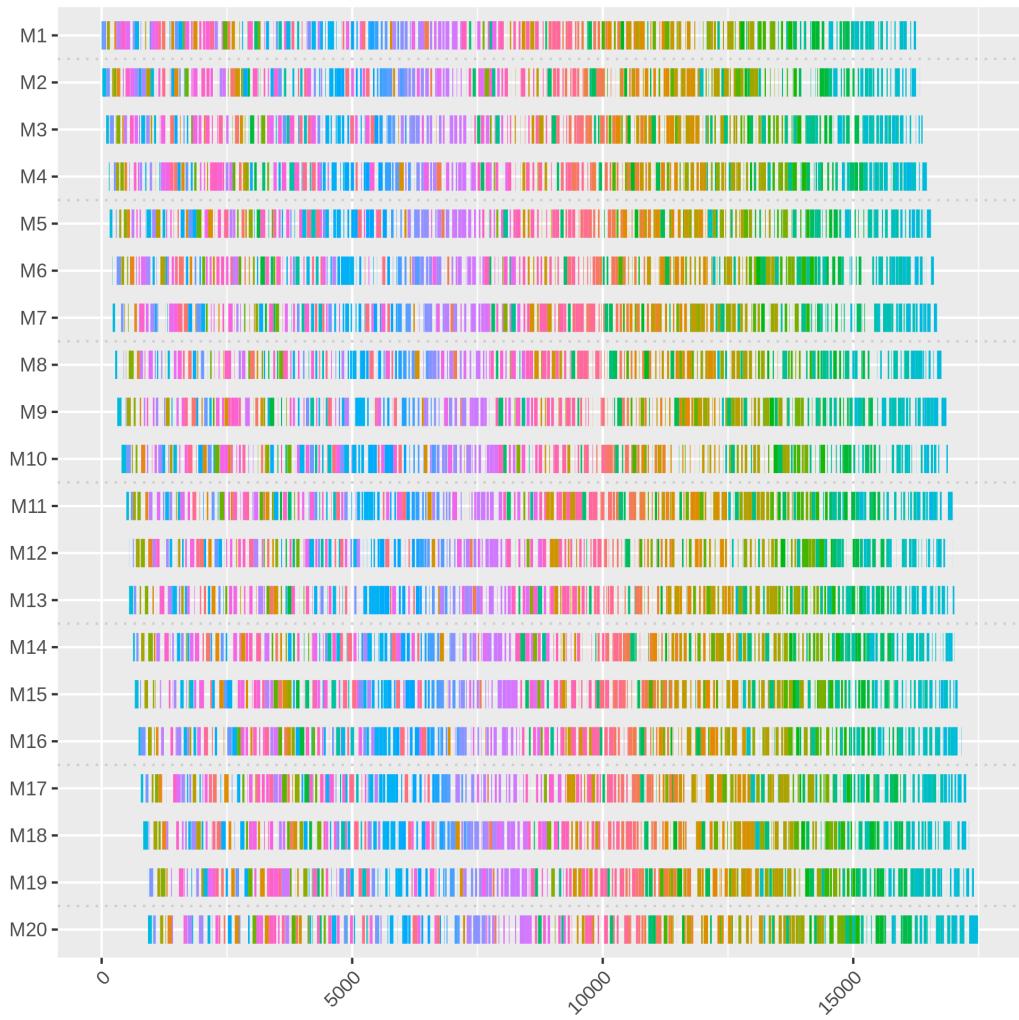


Figure 23: Gantt Chart for file 101.txt

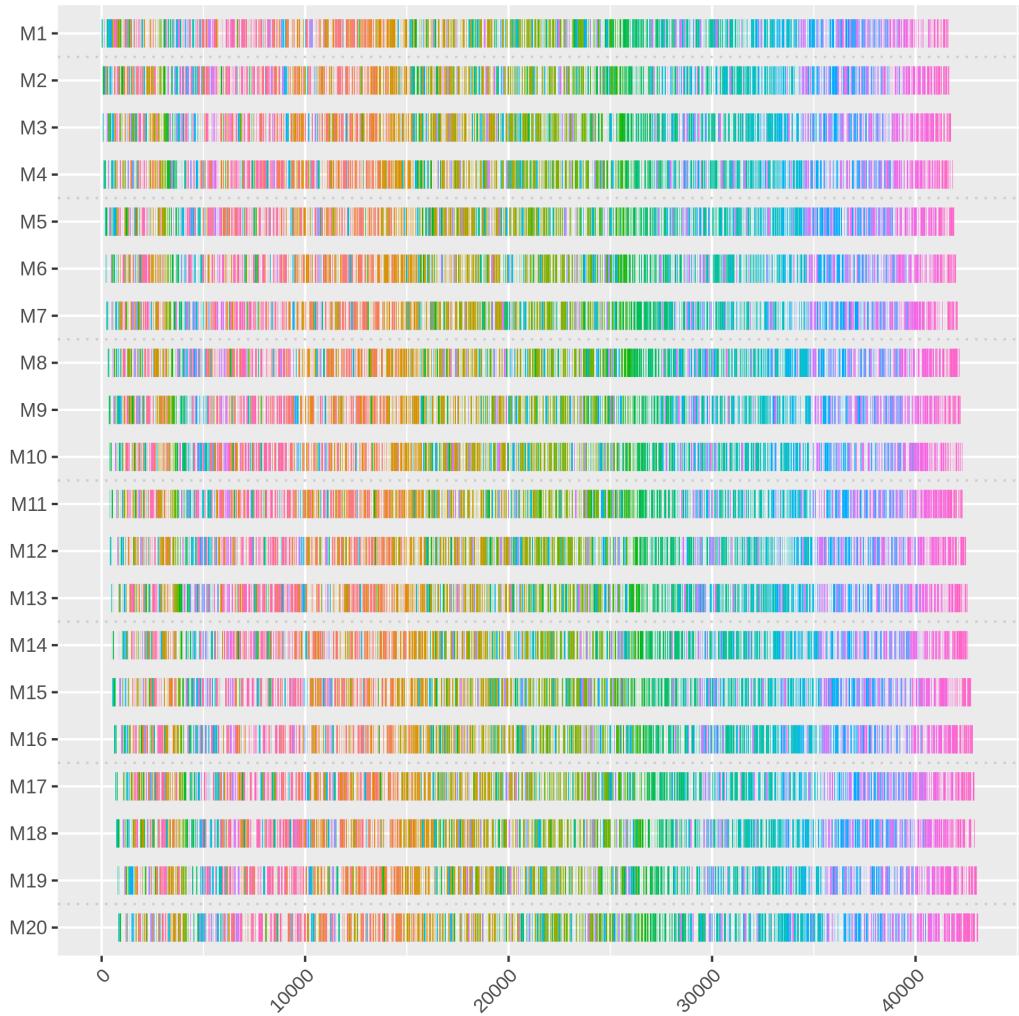


Figure 24: Gantt Chart for file 111.txt

2.2.2 Results for Flow Shop Scheduling with Blocking

	File.name	Best.makespan	# func calls
1/DataFiles/1.txt	1459	558
2/DataFiles/10.txt	1420	568
3/DataFiles/2.txt	1501	554
4/DataFiles/3.txt	1447	565
5/DataFiles/4.txt	1608	567
6/DataFiles/5.txt	1472	565
7/DataFiles/6.txt	1526	559
8/DataFiles/7.txt	1485	62
9/DataFiles/8.txt	1495	61
10/DataFiles/9.txt	1521	59
11/DataFiles/11.txt	1876	568
12/DataFiles/12.txt	1957	565
13/DataFiles/13.txt	1828	566
14/DataFiles/14.txt	1726	569
15/DataFiles/15.txt	1849	564
16/DataFiles/16.txt	1776	562
17/DataFiles/17.txt	1865	563
18/DataFiles/18.txt	1946	568
19/DataFiles/19.txt	1962	569
20/DataFiles/20.txt	1942	566
21/DataFiles/21.txt	2640	570
22/DataFiles/22.txt	2386	570
23/DataFiles/23.txt	2666	567
24/DataFiles/24.txt	2641	568
25/DataFiles/25.txt	2708	570
26/DataFiles/26.txt	2554	569
27/DataFiles/27.txt	2611	569
28/DataFiles/28.txt	2413	570
29/DataFiles/29.txt	2650	568
30/DataFiles/30.txt	2572	568
31/DataFiles/31.txt	3429	3645
32/DataFiles/32.txt	3536	3653
33/DataFiles/33.txt	3356	3646
34/DataFiles/34.txt	3493	3654
35/DataFiles/35.txt	3564	3659
36/DataFiles/36.txt	3486	3654
37/DataFiles/37.txt	3317	3642
38/DataFiles/38.txt	3380	3666
39/DataFiles/39.txt	3163	3663
40/DataFiles/40.txt	3507	3646
41/DataFiles/41.txt	4164	3664
42/DataFiles/42.txt	3834	3663
43/DataFiles/43.txt	4028	3658
44/DataFiles/44.txt	3011	3670
45/DataFiles/45.txt	4099	3666

	File.name	Best.makespan	# func calls
1/DataFiles/46.txt	3985	3664
2/DataFiles/47.txt	4062	3655
3/DataFiles/48.txt	3879	3650
4/DataFiles/49.txt	3837	3661
5/DataFiles/50.txt	3967	3653
6/DataFiles/51.txt	4961	3601
7/DataFiles/52.txt	4672	3648
8/DataFiles/53.txt	4830	3630
9/DataFiles/54.txt	4721	3664
10/DataFiles/55.txt	4863	3664
11/DataFiles/56.txt	4711	3664
12/DataFiles/57.txt	4867	3634
13/DataFiles/58.txt	4803	3610
14/DataFiles/59.txt	4760	3666
15/DataFiles/60.txt	4907	3664
16/DataFiles/61.txt	6917	14807
17/DataFiles/62.txt	6551	14805
18/DataFiles/63.txt	6585	14788
19/DataFiles/64.txt	6356	14753
20/DataFiles/65.txt	6629	14785
21/DataFiles/66.txt	6470	14775
22/DataFiles/67.txt	6637	14792
23/DataFiles/68.txt	6504	14763
24/DataFiles/69.txt	6741	14801
25/DataFiles/70.txt	6748	14765
26/DataFiles/71.txt	7769	14798
27/DataFiles/72.txt	7469	14795
28/DataFiles/73.txt	7610	14784
29/DataFiles/74.txt	7775	14758
30/DataFiles/75.txt	7627	14785
31/DataFiles/76.txt	7413	14786
32/DataFiles/77.txt	7463	14770
33/DataFiles/78.txt	7664	14469
34/DataFiles/79.txt	7802	14541
35/DataFiles/80.txt	7608	14695
36/DataFiles/81.txt	8441	14562
37/DataFiles/82.txt	8652	14350
38/DataFiles/83.txt	8757	14751
39/DataFiles/84.txt	8536	14709
40/DataFiles/85.txt	8683	14751
41/DataFiles/86.txt	8533	14515
42/DataFiles/87.txt	8608	14754
43/DataFiles/88.txt	8711	14578
44/DataFiles/89.txt	8581	14495
45/DataFiles/90.txt	8675	14736
46/DataFiles/100.txt	14786	58742

	File.name	Best.makespan	# func calls
1/DataFiles/91.txt	14734	57677
2/DataFiles/92.txt	14792	59020
3/DataFiles/93.txt	14937	58896
4/DataFiles/94.txt	14838	58610
5/DataFiles/95.txt	14722	58556
6/DataFiles/96.txt	14513	58662
7/DataFiles/97.txt	14802	58847
8/DataFiles/98.txt	14989	58235
9/DataFiles/99.txt	14696	59014
10/DataFiles/101.txt	15995	58936
11/DataFiles/102.txt	16268	58787
12/DataFiles/103.txt	16482	58806
13/DataFiles/104.txt	16348	58975
14/DataFiles/105.txt	16157	58951
15/DataFiles/106.txt	16132	58804
16/DataFiles/107.txt	16297	58903
17/DataFiles/108.txt	16268	58752
18/DataFiles/109.txt	16230	58960
19/DataFiles/110.txt	16306	59090
20/DataFiles/111.txt	39101	370915
21/DataFiles/112.txt	39163	370431
22/DataFiles/113.txt	39046	370045
23/DataFiles/114.txt	39260	370815
24/DataFiles/115.txt	38970	370413
25/DataFiles/116.txt	39389	371219
26/DataFiles/117.txt	38974	370676
27/DataFiles/118.txt	39075	372572
28/DataFiles/119.txt	39191	372435
29/DataFiles/120.txt	39396	372202

Table 2: Results for Flow Shop Scheduling with Blocking

2.2.3 Analysis

As seen from table1 above, both the makespan and the number of function calls increases as the number of machines or the number of jobs increases. Here results could not be compared to those on Taillard's website as the algorithm used to compute the makespan is different.

2.3 Flow Shop Scheduling with No Wait

2.3.1 Gantt Charts

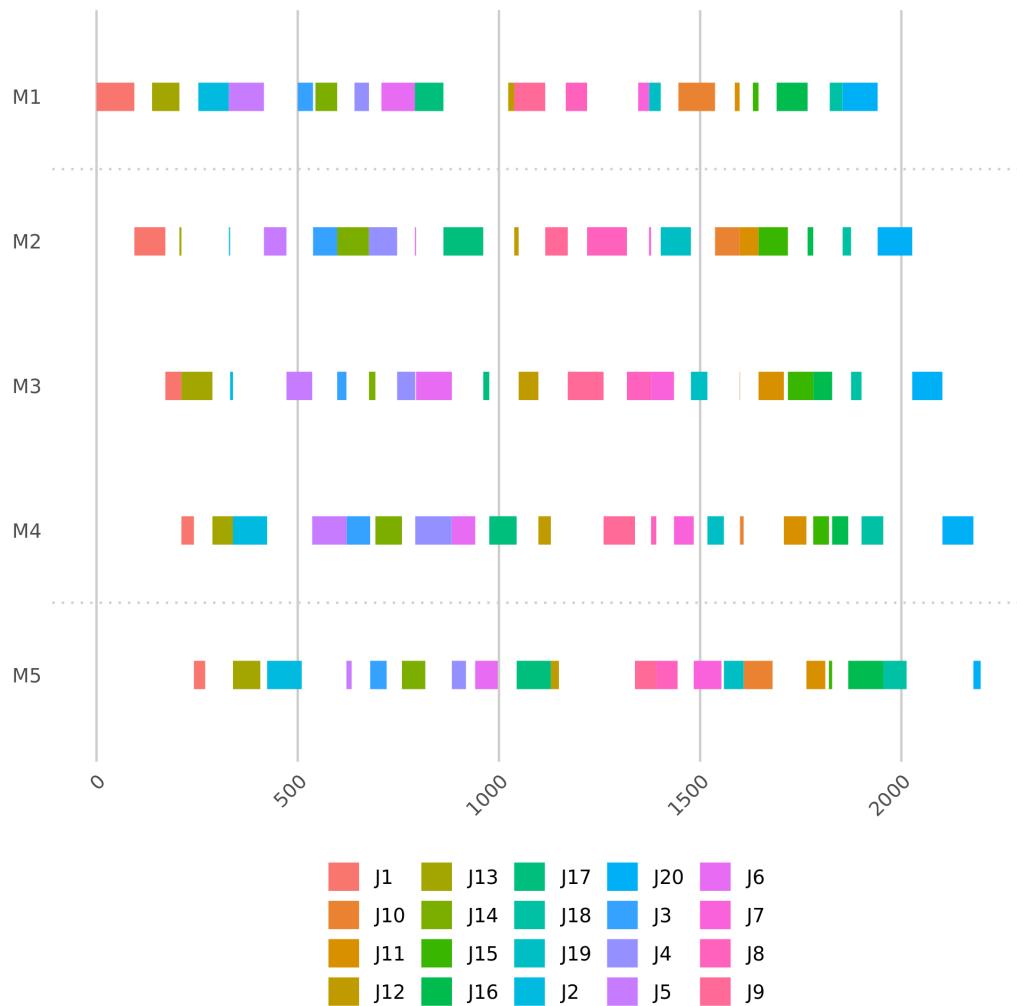


Figure 25: Gantt Chart for file 1.txt

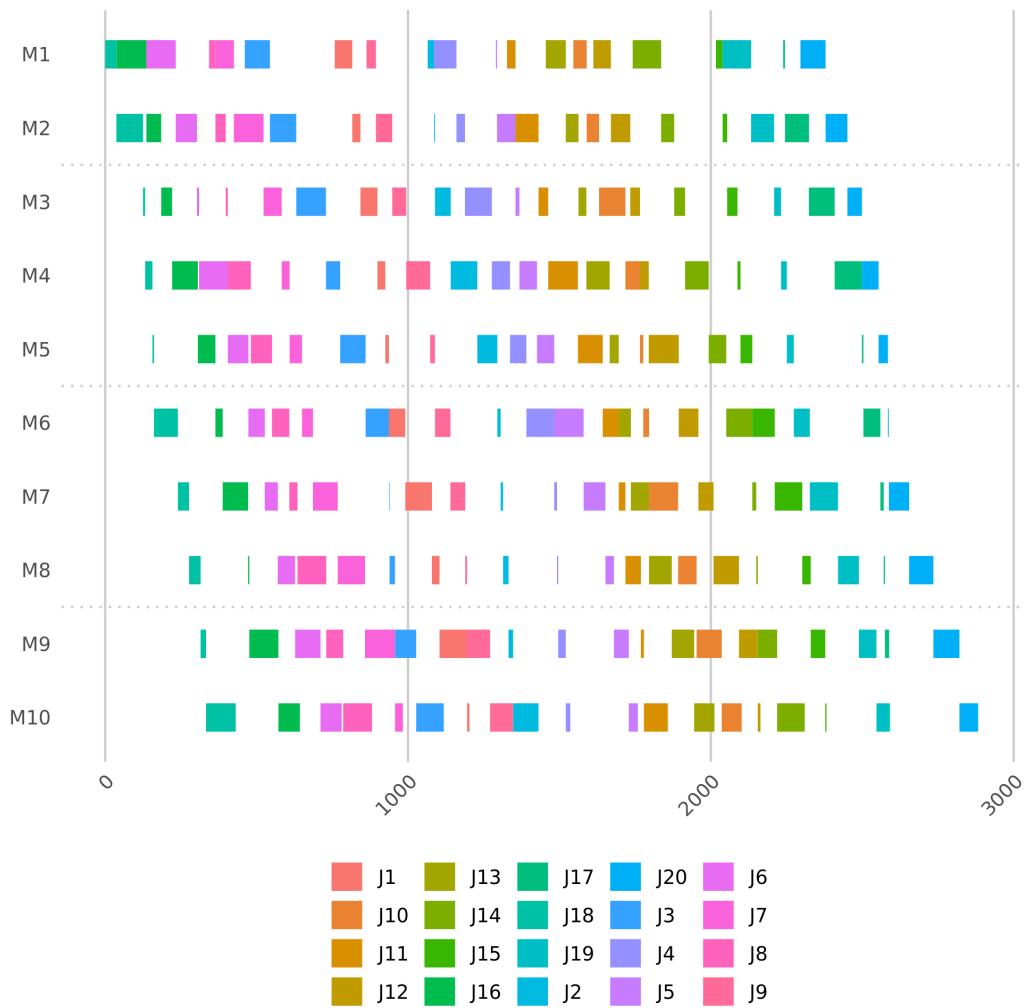


Figure 26: Gantt Chart for file 11.txt

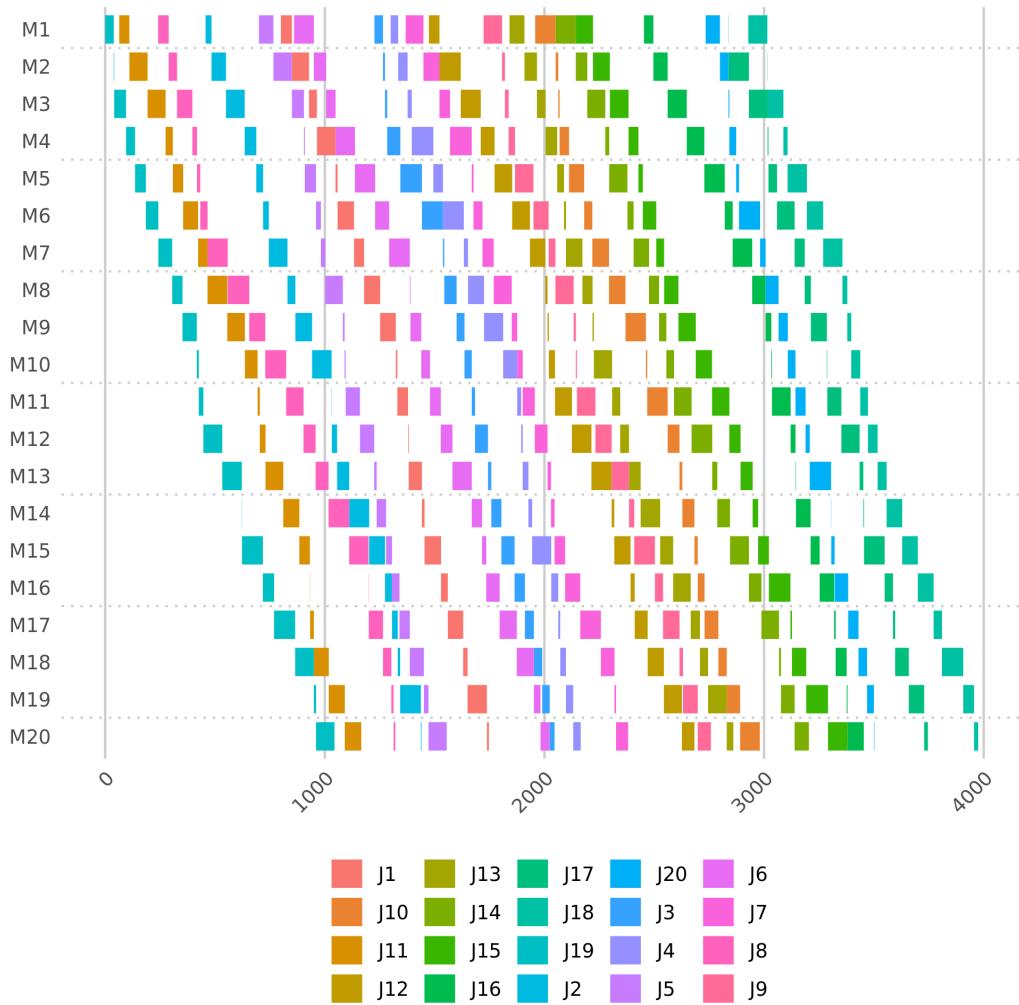


Figure 27: Gantt Chart for file 21.txt

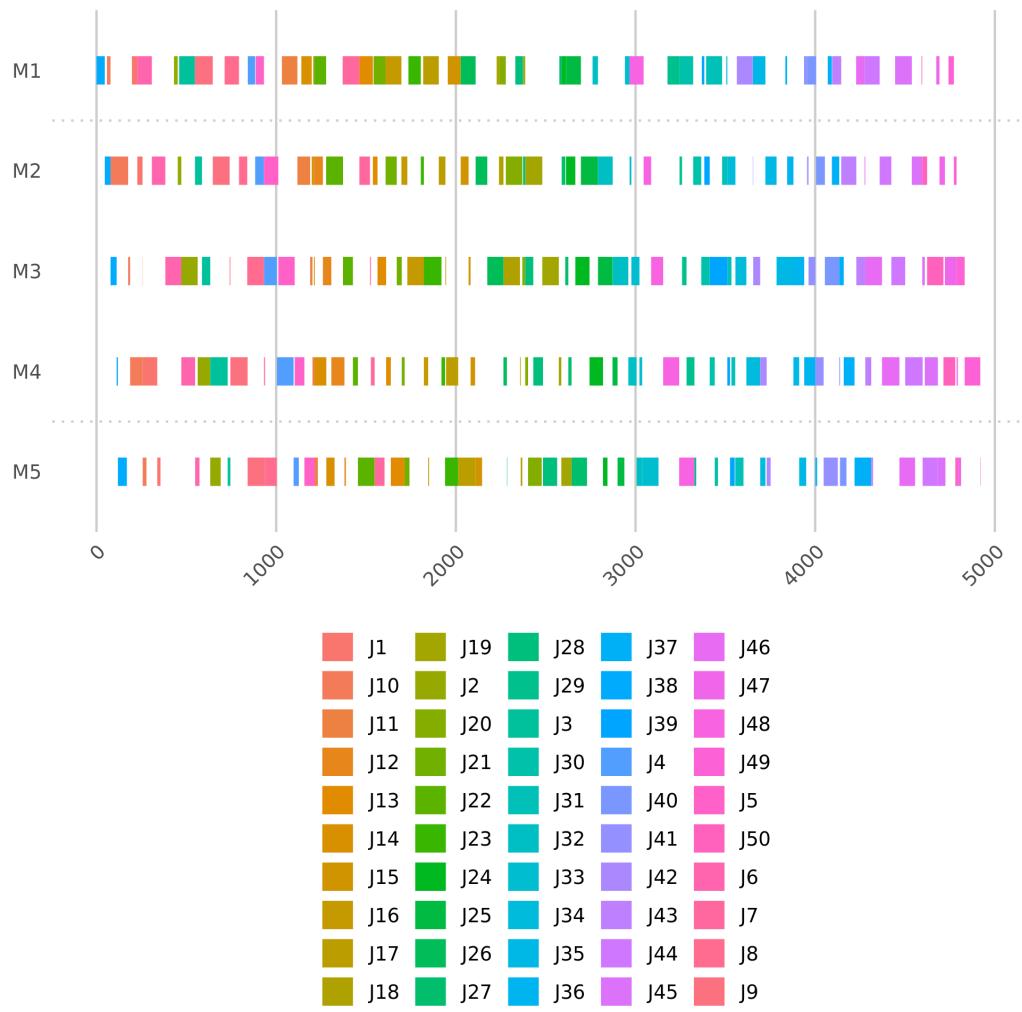


Figure 28: Gantt Chart for file 31.txt

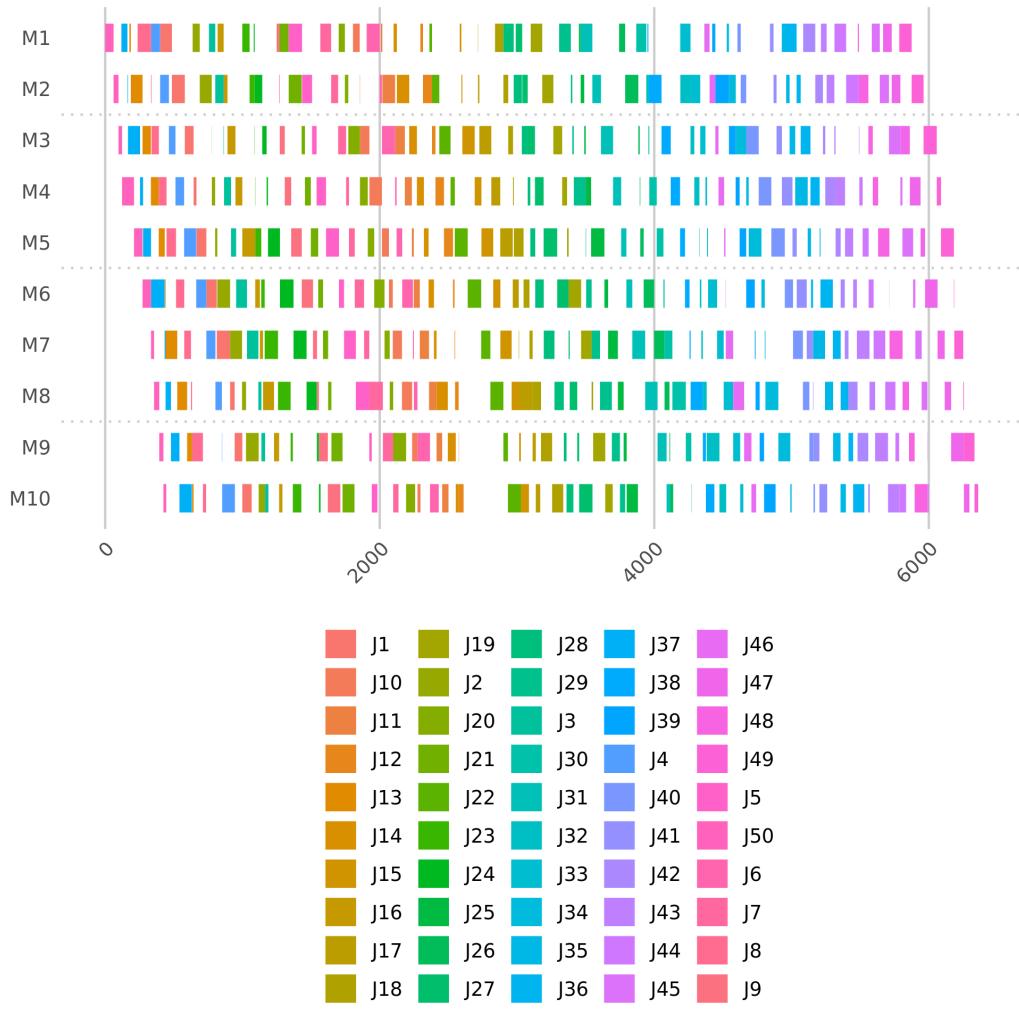


Figure 29: Gantt Chart for file 41.txt

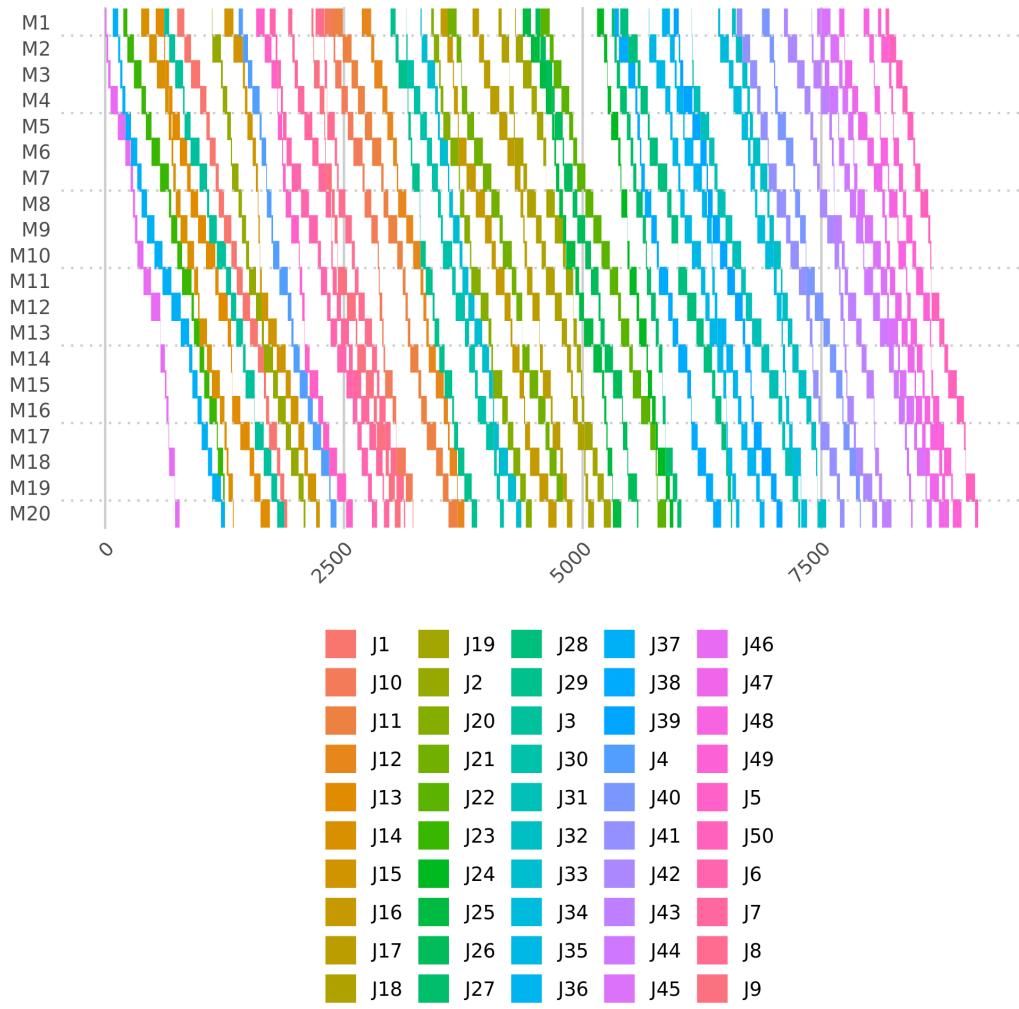


Figure 30: Gantt Chart for file 51.txt

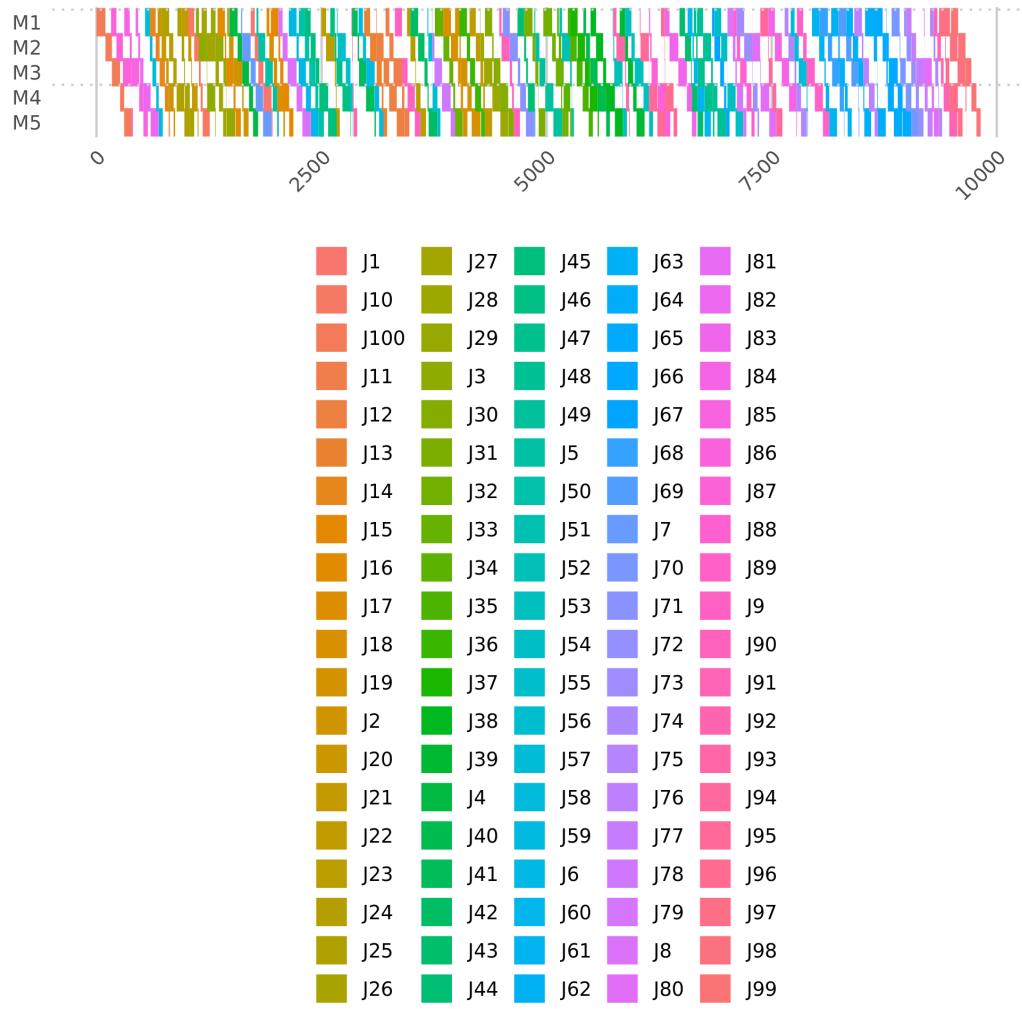


Figure 31: Gantt Chart for file 61.txt

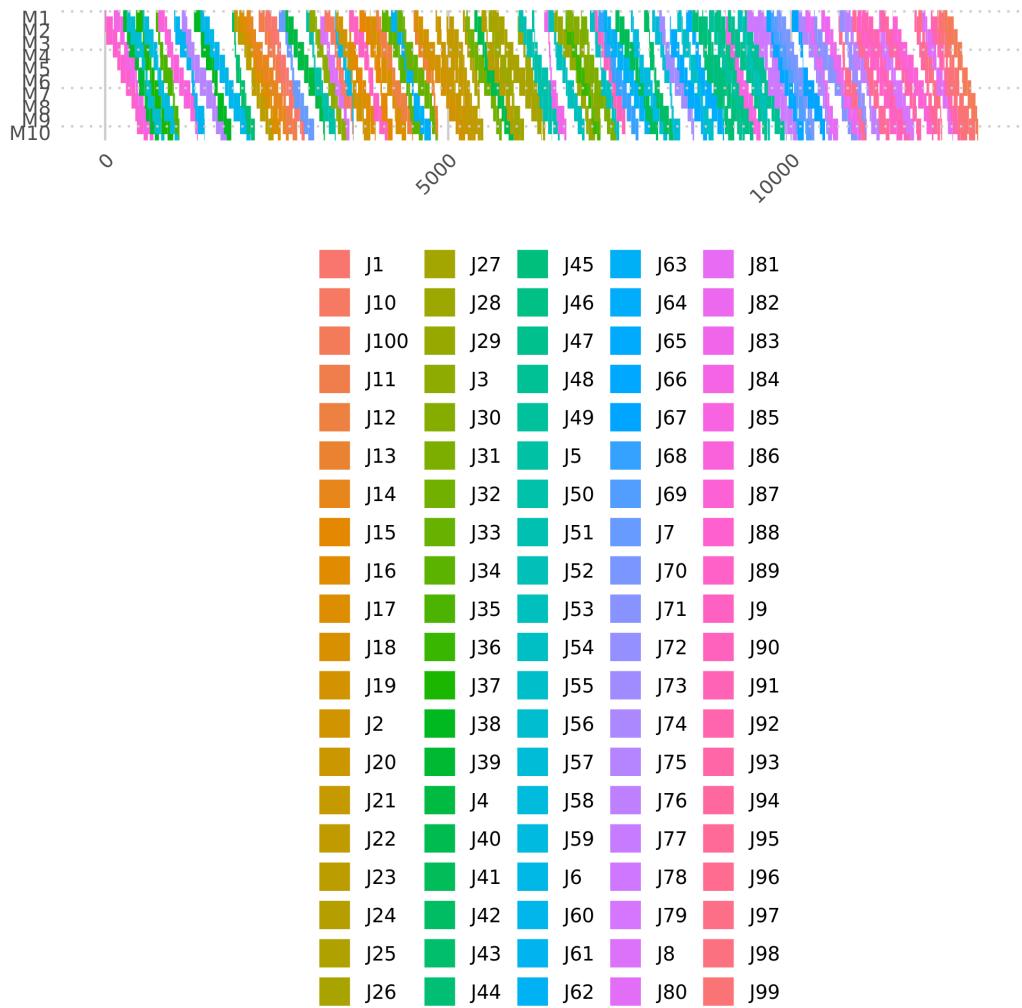


Figure 32: Gantt Chart for file 71.txt

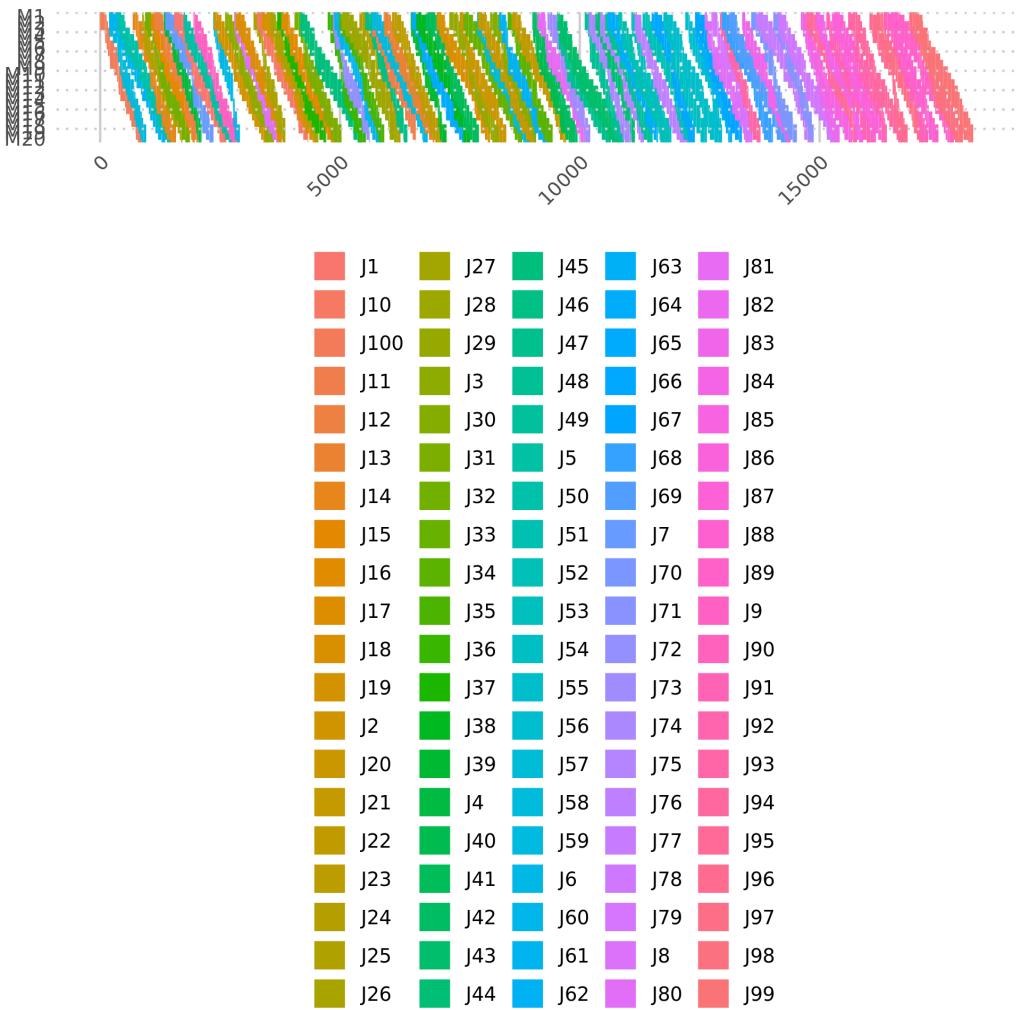


Figure 33: Gantt Chart for file 81.txt

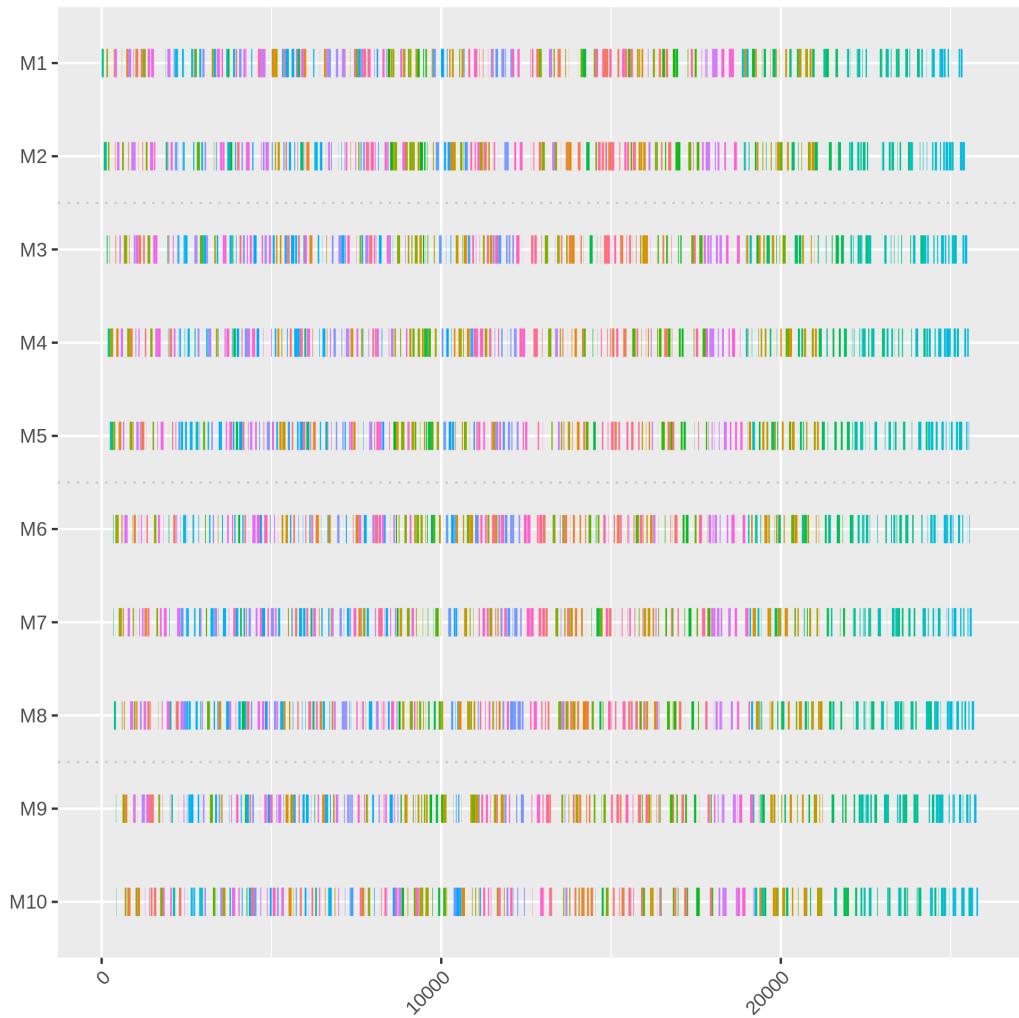


Figure 34: Gantt Chart for file 91.txt

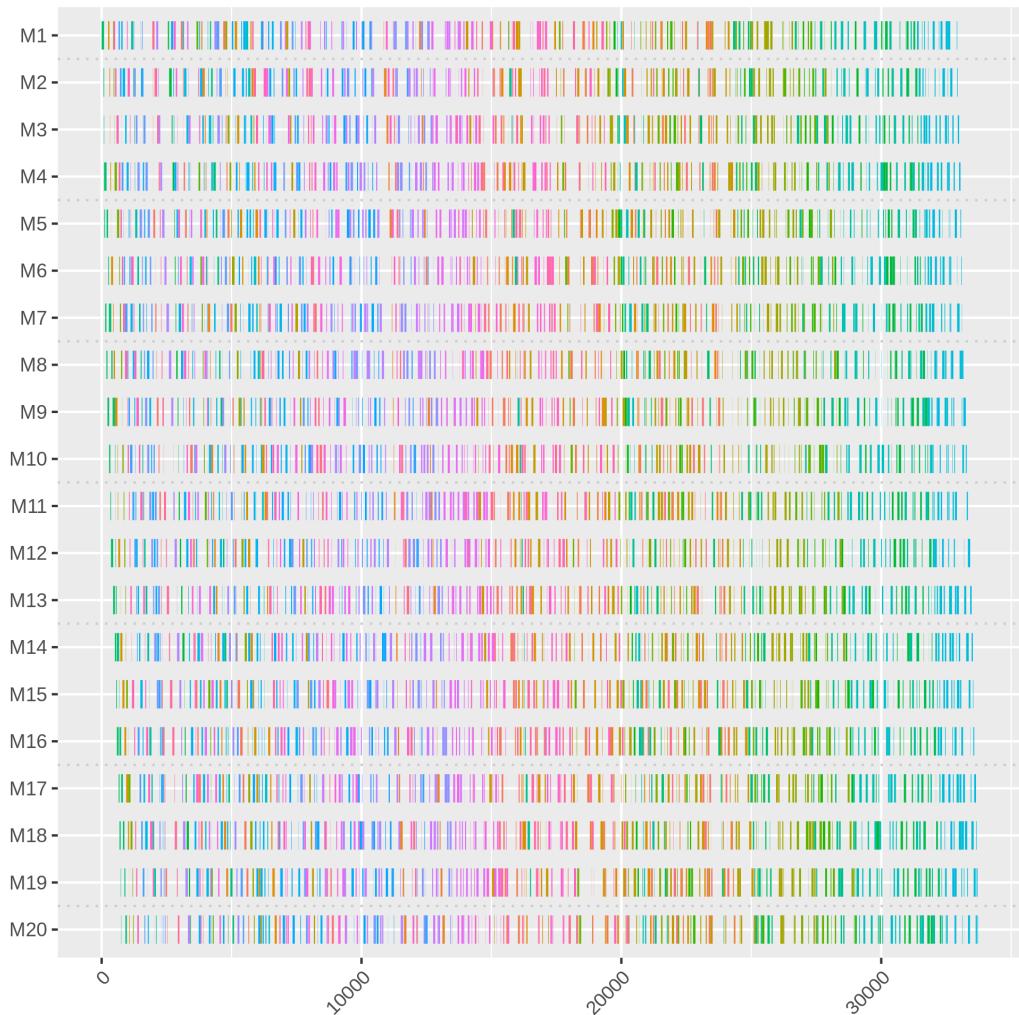


Figure 35: Gantt Chart for file 101.txt

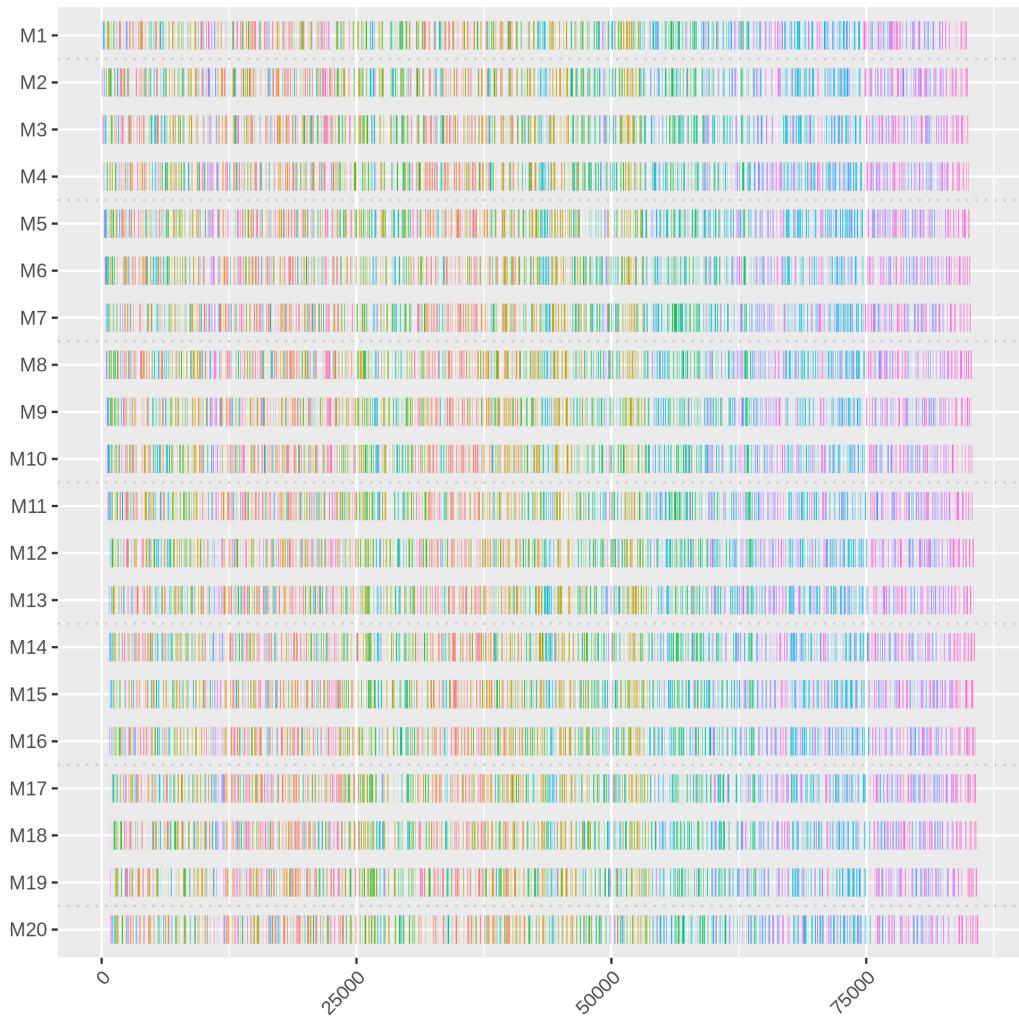


Figure 36: Gantt Chart for file 111.txt

2.3.2 Results for Flow Shop Scheduling with No Wait

	File.name	Best.total.flow.time	# func calls
1/DataFiles/1.txt	1614	567
2/DataFiles/10.txt	1494	562
3/DataFiles/2.txt	1644	565
4/DataFiles/3.txt	1581	566
5/DataFiles/4.txt	1676	564
6/DataFiles/5.txt	1585	566
7/DataFiles/6.txt	1577	565
8/DataFiles/7.txt	1548	564
9/DataFiles/8.txt	1634	561
10/DataFiles/9.txt	1649	570
11/DataFiles/11.txt	2339	570
12/DataFiles/12.txt	2286	566
13/DataFiles/13.txt	2089	566
14/DataFiles/14.txt	1997	566
15/DataFiles/15.txt	2249	570
16/DataFiles/16.txt	2146	566
17/DataFiles/17.txt	2206	569
18/DataFiles/18.txt	2223	562
19/DataFiles/19.txt	2147	570
20/DataFiles/20.txt	2140	566
21/DataFiles/21.txt	3224	567
22/DataFiles/22.txt	2990	565
23/DataFiles/23.txt	3235	570
24/DataFiles/24.txt	3270	568
25/DataFiles/25.txt	3231	570
26/DataFiles/26.txt	3218	569
27/DataFiles/27.txt	3267	567
28/DataFiles/28.txt	2978	568
29/DataFiles/29.txt	3260	568
30/DataFiles/30.txt	3218	568
31/DataFiles/31.txt	3639	3657
32/DataFiles/32.txt	3837	3662
33/DataFiles/33.txt	3625	3641
34/DataFiles/34.txt	3636	3616
35/DataFiles/35.txt	3727	3650
36/DataFiles/36.txt	3573	3661
37/DataFiles/37.txt	3542	3645
38/DataFiles/38.txt	3601	3658
39/DataFiles/39.txt	3472	3662
40/DataFiles/40.txt	3694	3639
41/DataFiles/41.txt	4727	3647
42/DataFiles/42.txt	4630	3657
43/DataFiles/43.txt	4736	3657
44/DataFiles/44.txt	4868	3659
45/DataFiles/45.txt	4874	3662

	File.name	Best.total.flow.time	# func calls
1/DataFiles/46.txt	4744	3658
2/DataFiles/47.txt	4901	3664
3/DataFiles/48.txt	4782	3663
4/DataFiles/49.txt	4705	3651
5/DataFiles/50.txt	4728	3663
6/DataFiles/51.txt	6828	3664
7/DataFiles/52.txt	6593	3661
8/DataFiles/53.txt	6590	3662
9/DataFiles/54.txt	6539	3667
10/DataFiles/55.txt	6642	3671
11/DataFiles/56.txt	6604	3665
12/DataFiles/57.txt	6655	3667
13/DataFiles/58.txt	6549	3668
14/DataFiles/59.txt	6692	3664
15/DataFiles/60.txt	6618	3664
16/DataFiles/61.txt	7328	14782
17/DataFiles/62.txt	7078	14785
18/DataFiles/63.txt	6850	14802
19/DataFiles/64.txt	6725	14788
20/DataFiles/65.txt	6879	14781
21/DataFiles/66.txt	6966	14797
22/DataFiles/67.txt	6976	14807
23/DataFiles/68.txt	6991	14813
24/DataFiles/69.txt	7242	14811
25/DataFiles/70.txt	7136	14802
26/DataFiles/71.txt	8949	14797
27/DataFiles/72.txt	8857	14817
28/DataFiles/73.txt	9170	14826
29/DataFiles/74.txt	9571	14808
30/DataFiles/75.txt	8982	14815
31/DataFiles/76.txt	8866	14821
32/DataFiles/77.txt	8869	14808
33/DataFiles/78.txt	8983	14792
34/DataFiles/79.txt	9277	14791
35/DataFiles/80.txt	9304	14826
36/DataFiles/81.txt	11927	14811
37/DataFiles/82.txt	12517	14780
38/DataFiles/83.txt	12077	14830
39/DataFiles/84.txt	12406	14715
40/DataFiles/85.txt	12102	14800
41/DataFiles/86.txt	12059	14812
42/DataFiles/87.txt	12339	14820
43/DataFiles/88.txt	12308	14802
44/DataFiles/89.txt	12163	14802
45/DataFiles/90.txt	12442	14719
46/DataFiles/100.txt	17519	59058

	File.name	Best.total.flow.time	# func calls
1/DataFiles/91.txt	17564	59189
2/DataFiles/92.txt	17628	59302
3/DataFiles/93.txt	17359	59187
4/DataFiles/94.txt	17692	59050
5/DataFiles/95.txt	17286	59357
6/DataFiles/96.txt	17595	59192
7/DataFiles/97.txt	17663	59317
8/DataFiles/98.txt	17637	59444
9/DataFiles/99.txt	17506	59137
10/DataFiles/101.txt	22958	59075
11/DataFiles/102.txt	23402	59198
12/DataFiles/103.txt	23207	59241
13/DataFiles/104.txt	23354	59225
14/DataFiles/105.txt	22958	59214
15/DataFiles/106.txt	23103	59333
16/DataFiles/107.txt	23099	59272
17/DataFiles/108.txt	23382	59291
18/DataFiles/109.txt	23084	59329
19/DataFiles/110.txt	22944	59354
20/DataFiles/111.txt	54322	371479
21/DataFiles/112.txt	55440	371041
22/DataFiles/113.txt	54803	370990
23/DataFiles/114.txt	54526	371147
24/DataFiles/115.txt	54944	371190
25/DataFiles/116.txt	55518	371742
26/DataFiles/117.txt	54347	371507
27/DataFiles/118.txt	54567	372950
28/DataFiles/119.txt	54161	372592
29/DataFiles/120.txt	55313	373286

Table 3: Results for Flow Shop Scheduling with No Wait

2.3.3 Analysis

As seen from table1 above, both the total flow time and the number of function calls increases as the number of machines or the number of jobs increases. Here results could not be compared to those on Taillard's website as the algorithm used to compute the total flow time is different.

3 Conclusion

After implementing, experimenting with and analyzing the results of the NEH algorithm, it can be determined that even though NEH is reported to be the best algorithm for solving flow shop type of problems, its efficiency highly depends on how it is implemented. For example, the results of this project are different from those reported on Taillard's website even though the same concept was used to perform the optimization.