CS471 Project5

JuneYeob Lee(2462 9603)

May 31, 2019

1 Introduction

Project 5 is aiming to get the shortest makespan with Flowshop (FSS), Flowshop with blocking (FSSB), AND Flowshop with no-wait (FSSNW). NEH heuristic algorithm had been invented by Nawaz, Enscore, and Ham and published in 1983. NEH is regarded as one of the most effective heuristics to solve permutative flow shop ploblems. For example, the results of project 5 from Taillard data sets. The Taillard data sets has 120 different data set with 12 different number of machines and jobs. The sequence of producing makespan from Taillard data sets is to first get the best job ordering through NEH heuristic algorithm from Taillard data sets and then calculate makespan using the FSS, FSSB, and FSSNW. Project 5 report includes one tabulated table about average and standard deviation of a makespan and average execution time for 12 different machine and job environment of Taillard data sets. The following are 18 time plot graphs. These time plots graphs are the 1st, 11th, 21th, 31th, 41th, 51th, and 61th Taillard data sets with FSS, FSSB, and FSSNW.

2 ANALYSIS

According to Table 3.1 and figure 3.1, Flowshop scheduling has the most optimized makespan among three algorithms. Basically, makespan is determined by the number of jobs and machines. If the number of machines and jobs are same, FSS gives the smallest value, followed by FSSB, and then FSSNW. For the data sets between 1 and 10 (5 machines and 20 jobs), FSS has the shortest makespan average, but FSSB has the shortest execution time average. For the data sets between 11 and 20 (10 machines and 20 jobs), FSS has the shortest makespan average. Overall, FSS has the shortest and the most optimized makespan value among three algorithms. According to the figure 3.2, 3.3, and 3.4, using FSS algorithm is the most effective compare to FSSB and FSSNW because FSS algorithm's first machine doesn't have waiting time to process next jobs. Due to FSSB and FSSNW requiring extra wait time means FSSB and FSSNW need more total time to complete all jobs. According to all tables FSS has the shortest makespan among three- scheduling algorithms. Makespan increased by number of jobs and machine because if the number of jobs are increased completion time will be higher. Also, makespan is proportional to number of machines.

3 CONCLUSION

In conclusion, makespan is proportional to the number of jobs and machines, this fact will not be changed. According to result tables, FSS has the shortest makespan which means FSS algorithm need the shortest time to complete offered jobs, however, in a practical setting, this cannot be implemented due to the fact that machines overheat from overwork, thus need breaks. FSS algorithm requires that first machine should work all jobs without wait time , this would cause unavoidable to overheating. Even though FSS has the best result among three algorithm, FSSB would be better be used in the industry. In my opinion, FSS, FSSB, and FSSNW has limitation to be optimized because all machines should work on job in same order. I would like to say if each machine has each optimized schedule without job conflict, makespan will be shorter. Next experiment would be scheduling jobs for each machine to get shorter makespan value.

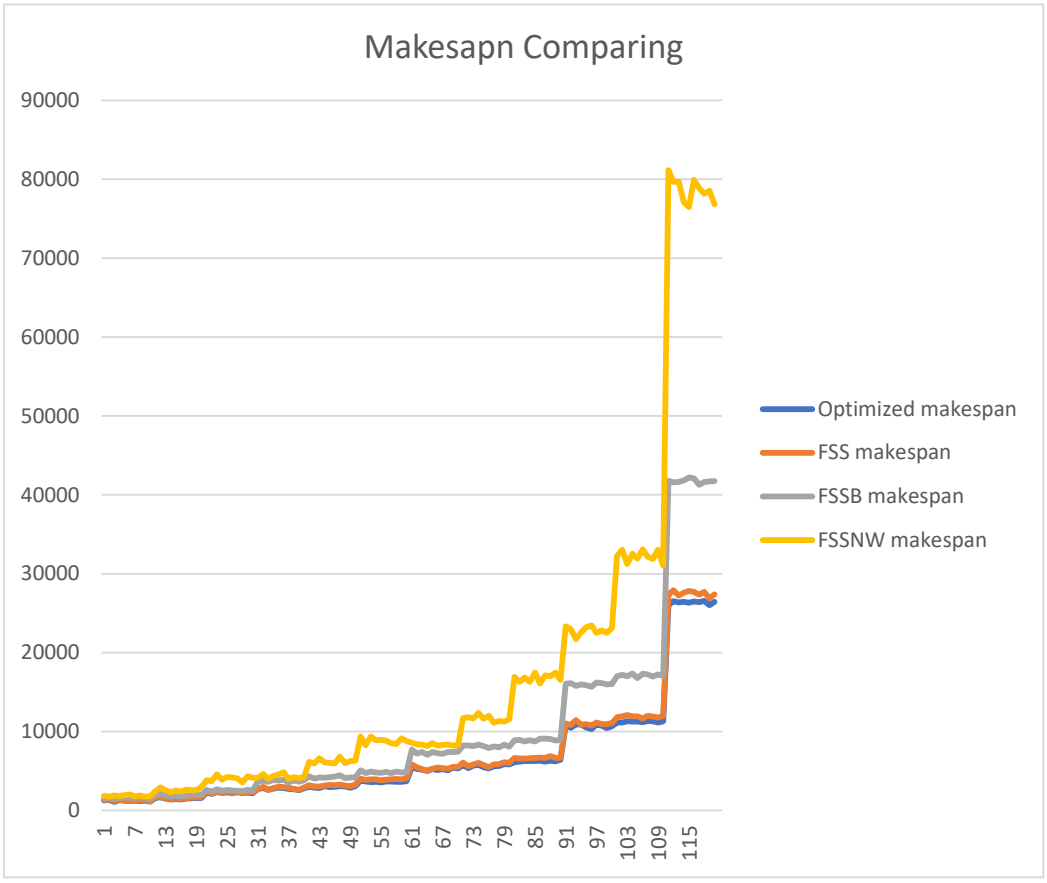

Figure 3.1: Make span graph with 3 algorithms for 120 Data files

Table 3.1: Total result(make span and execution time

	Optimized		FSS			FSSB			FSSNW		
Data File	AVG	STD	AVG	STD	TIME(m/s)	AVG	STD	TIME(m/s)	AVG	STD	TIME(m/s)
1~10	1193	120.2	1221.5	96.87	1.5	1488.5	82.73	1	1834	50.9	1.5
11~20	1513.6	94.9	1588.3	109.37	0	1891.5	101.17	0	2583.5	222.1	0
21~30	2235	66.6	2330	92.24	1	2532.2	89.88	1	4052.9	303.4	1
31~40	98.24934266	98.2	2819.6	158.58	3.5	3745	120.29	10	4309.9	280.6	7
41~50	2983.4	52.3	3162.7	40.31	3	4208.9	136.47	7	6212.2	118.1	11.5
51~60	3652.5	53.0	3939.5	36.77	7	4856.6	134.35	7.5	8852.7	417.9	5.5
61~70	5244.5	572.0	5399.6	595.38	12	7346.9	978.64	14.5	8335.2	5313.2	11
71~80	5627.4	53.0	5858.3	31.82	21	8150.8	108.89	18.5	11643	83.4	16
81~90	6246.6	210.7	6659.5	26.16	41.5	8924.3	43.84	35	16801.2	282.8	41
91~100	10668.5	145.0	10996.9	24.75	241.5	15977.8	21.92	198.5	22827	166.2	205
101~110	11235.3	93.3	11897.6	118.09	322	17112.9	41.01	291	32218.2	828.0	265
111~120	26358.1	294.9	27485.4	21.92	4662	41746.4	32.53	4567	78652.6	3050.5	4105.5

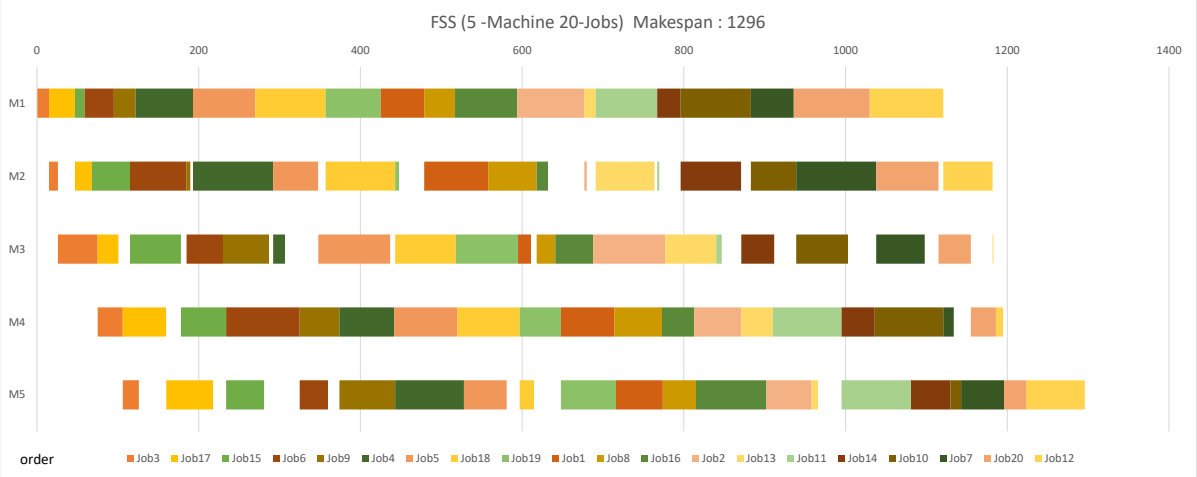

Figure 3.2: FSS with 5 machines and 20 jobs

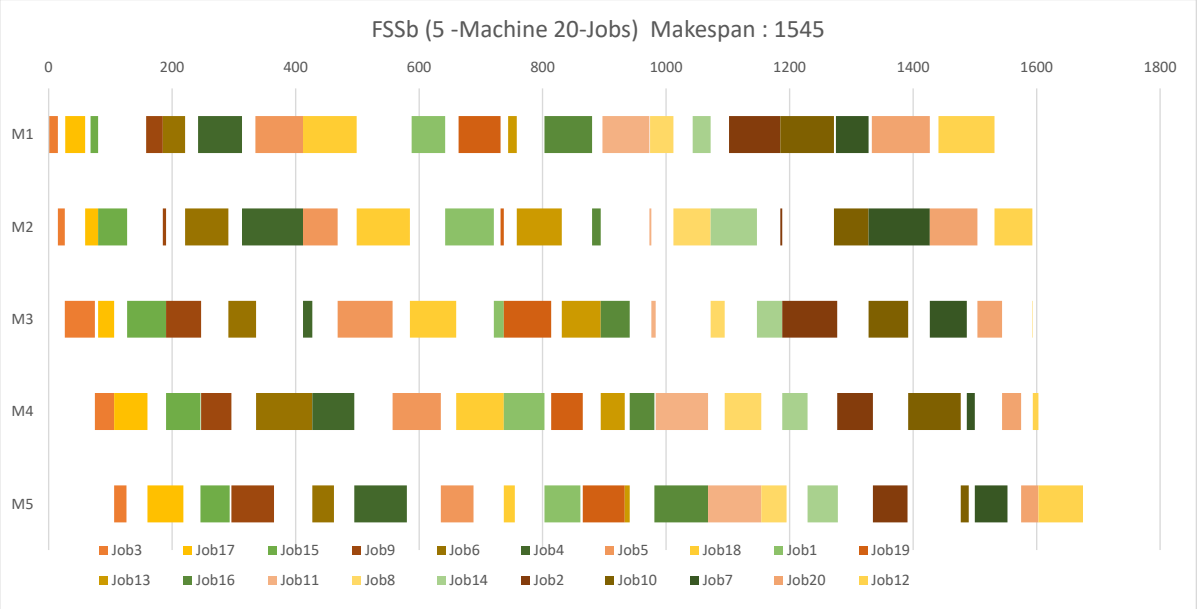

Figure 3.3: FSSB with 5 machines and 20 jobs

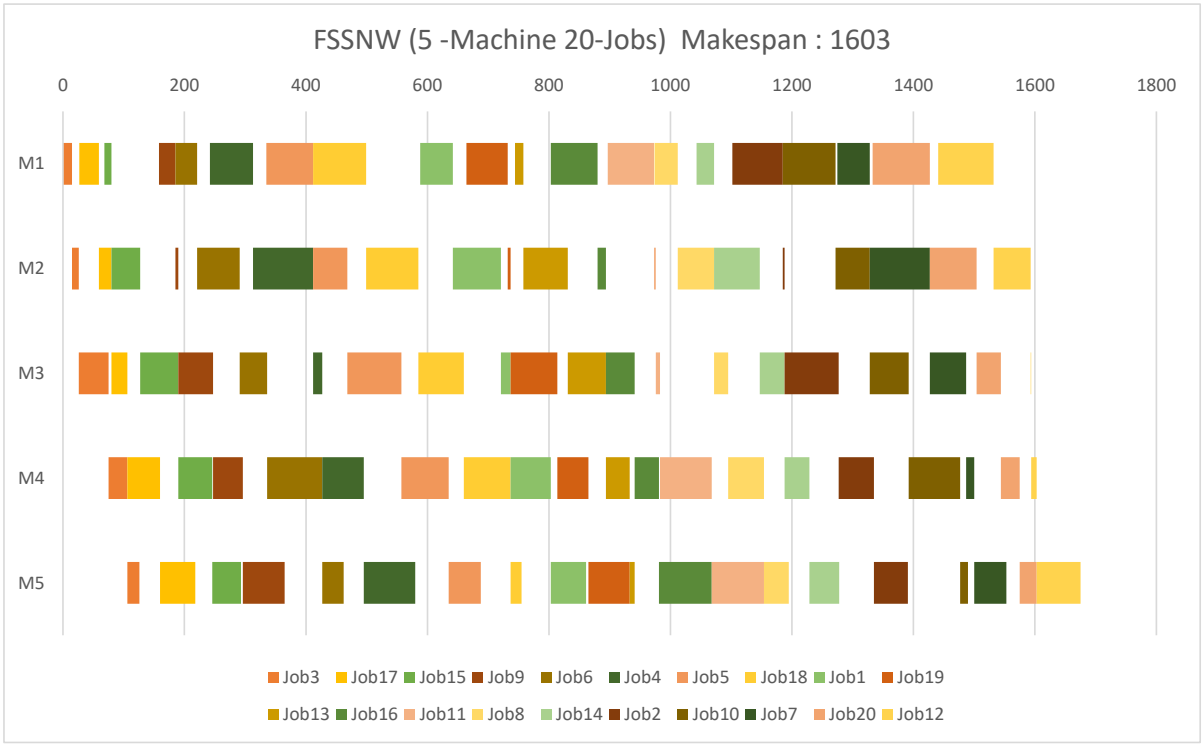

Figure 3.4: FSSNW with 5 machines and 20 jobs

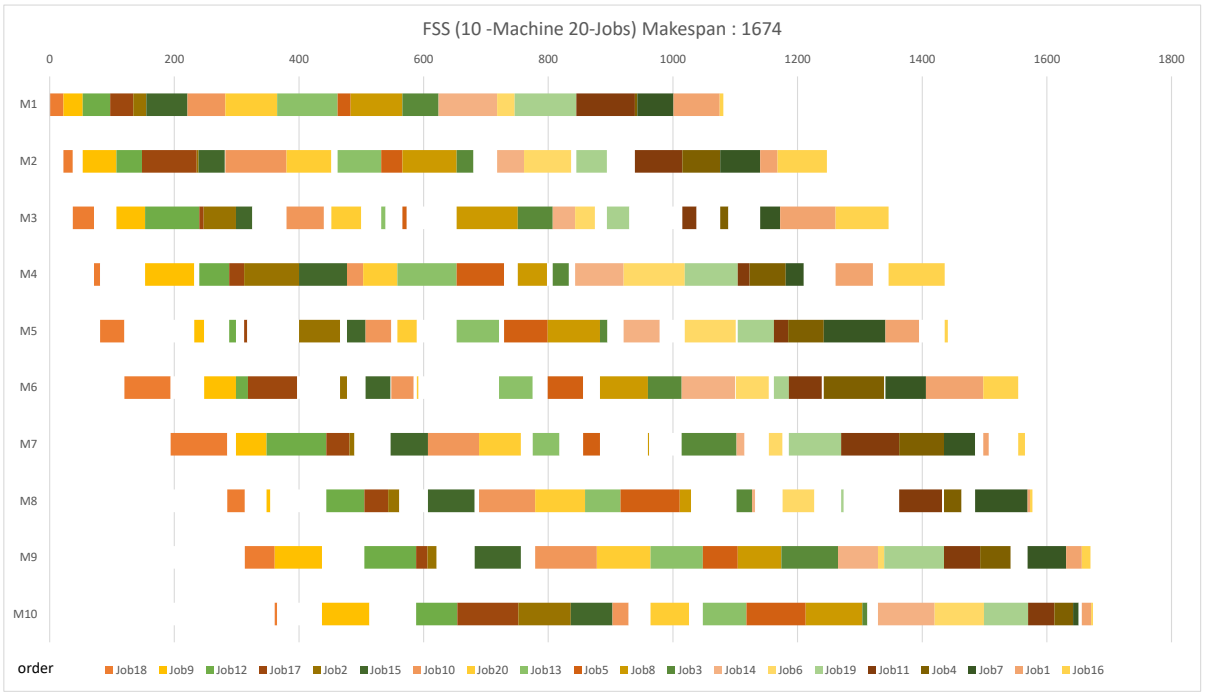

Figure 3.5: FSS with 10 machines and 20 jobs

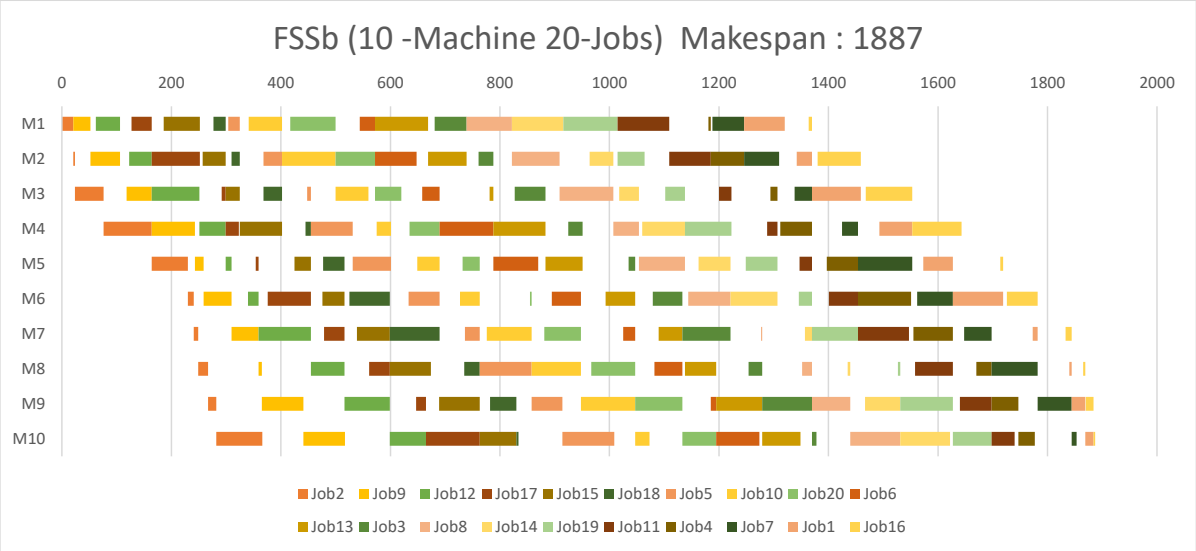

Figure 3.6: FSSB with 10 machines and 20 jobs

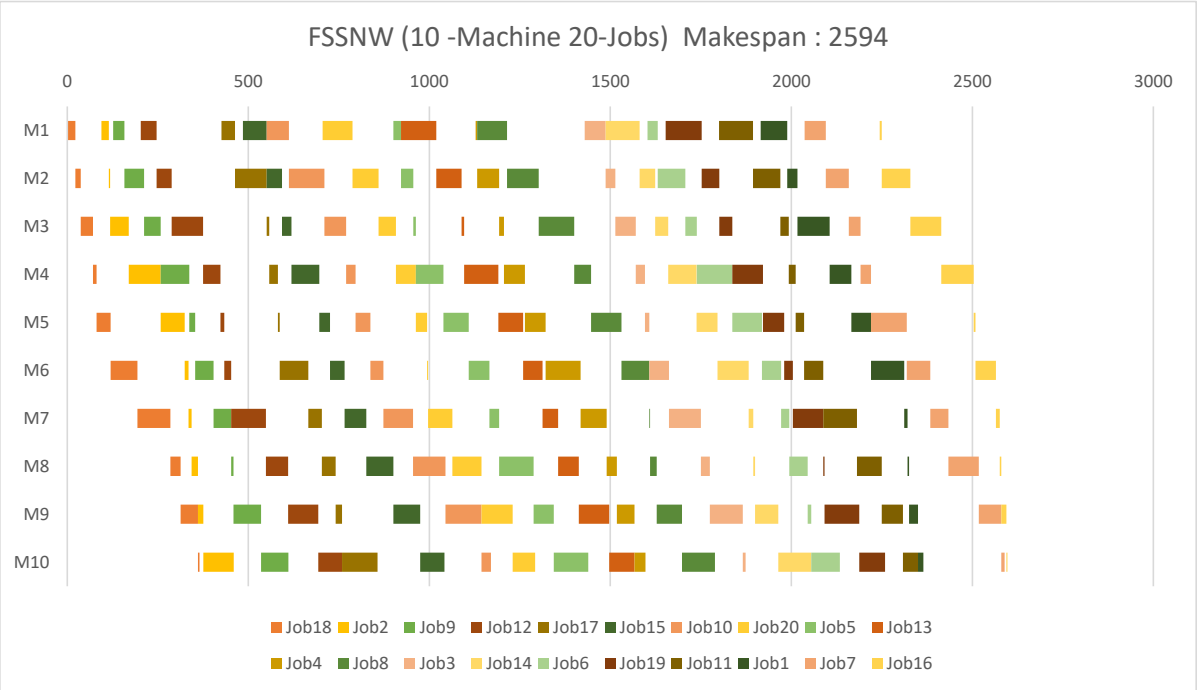

Figure 3.7: FSSNW with 10 machines and 20 jobs

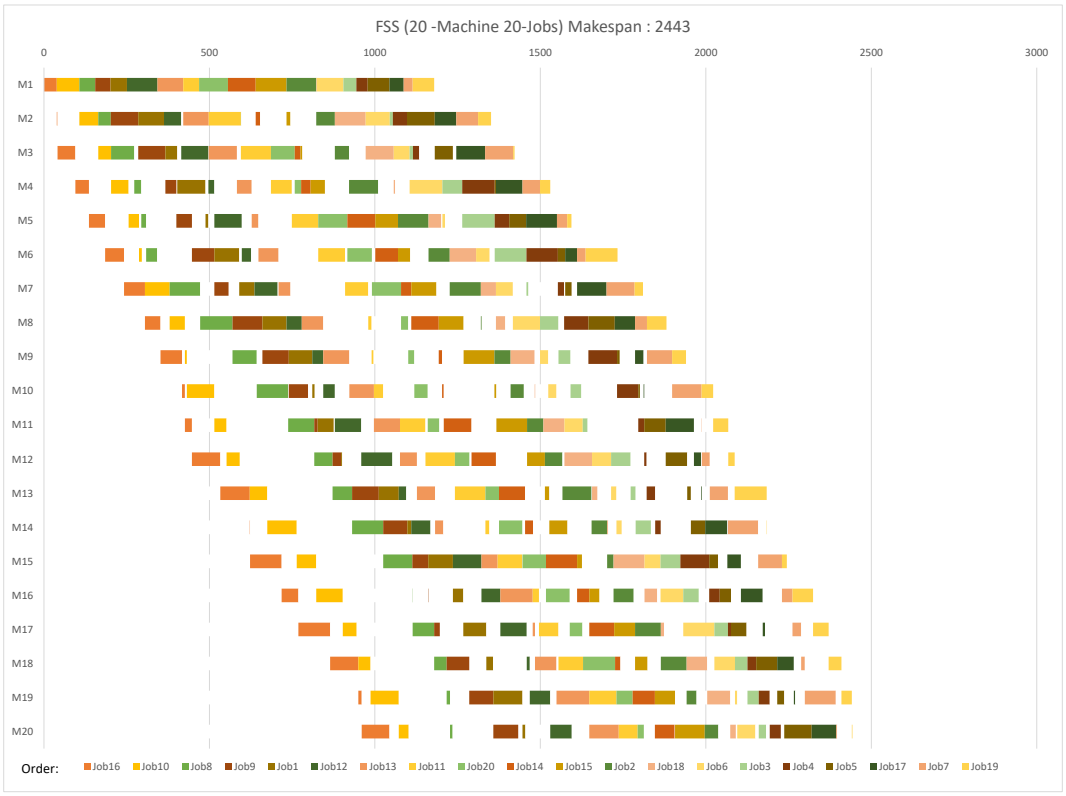

Figure 3.8: FSS with 20 machines and 20 jobs

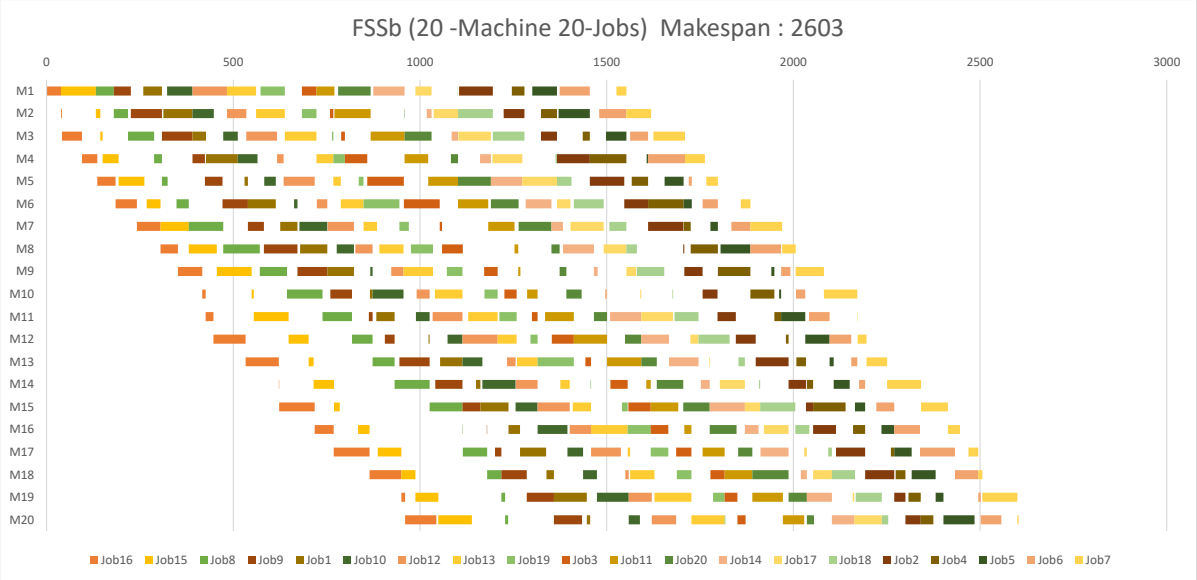

Figure 3.9: FSSB with 20 machines and 20 jobs

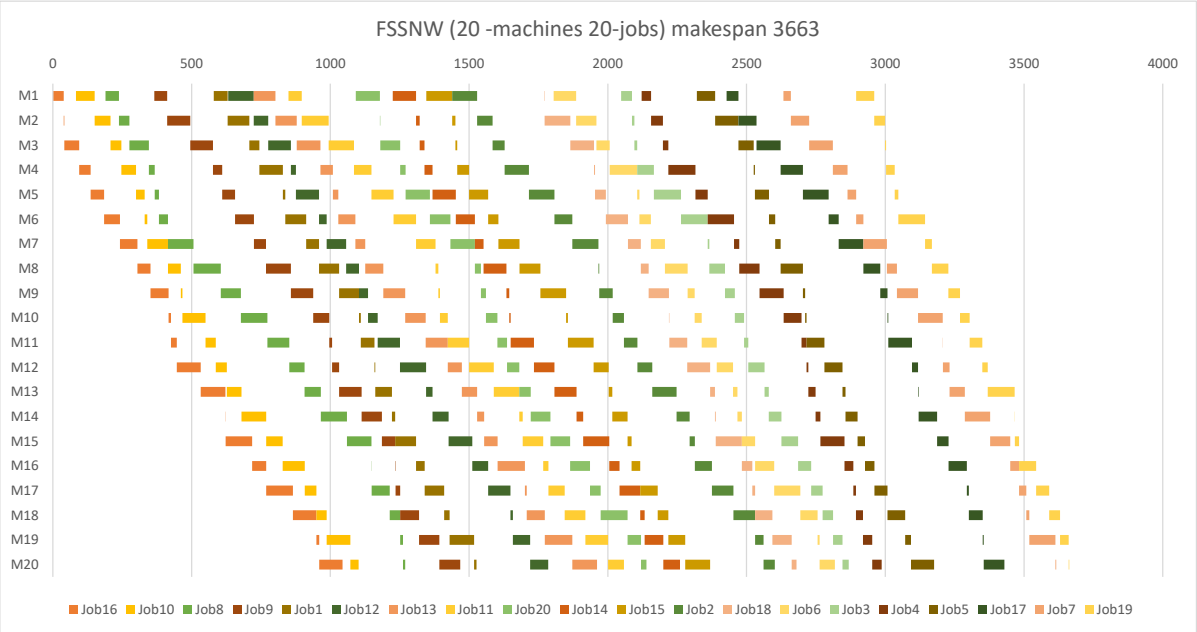

Figure 3.10: FSSNW with 20 machines and 20 jobs

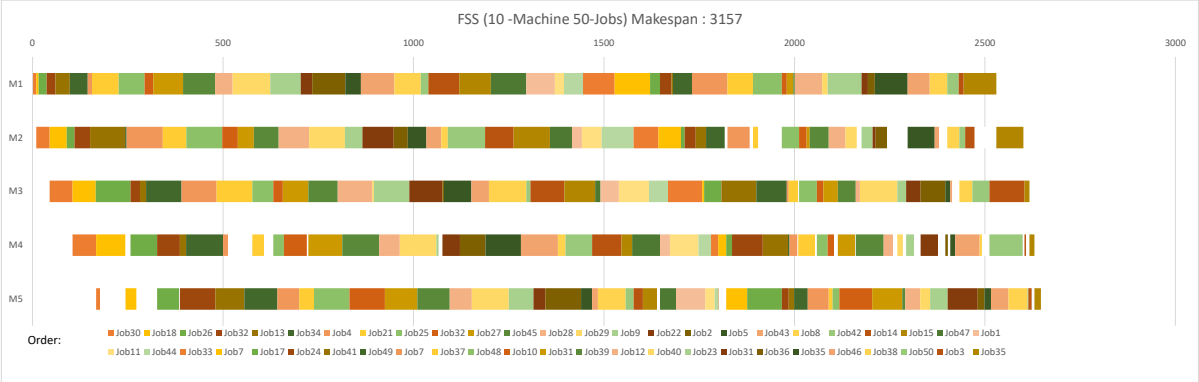

Figure 3.11: FSS with 5 machines and 50 jobs

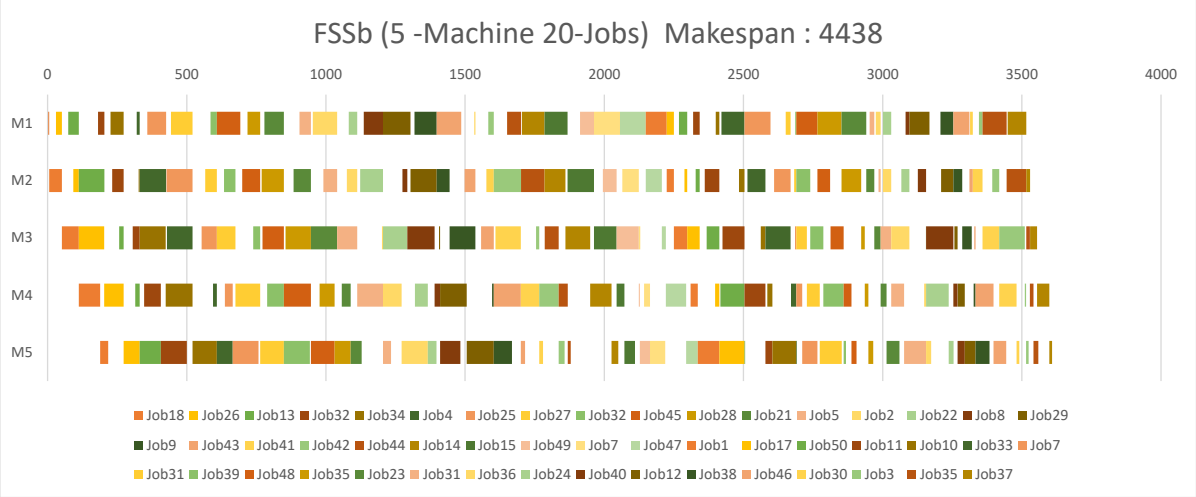

Figure 3.12: FSSB with 5 machines and 50 jobs

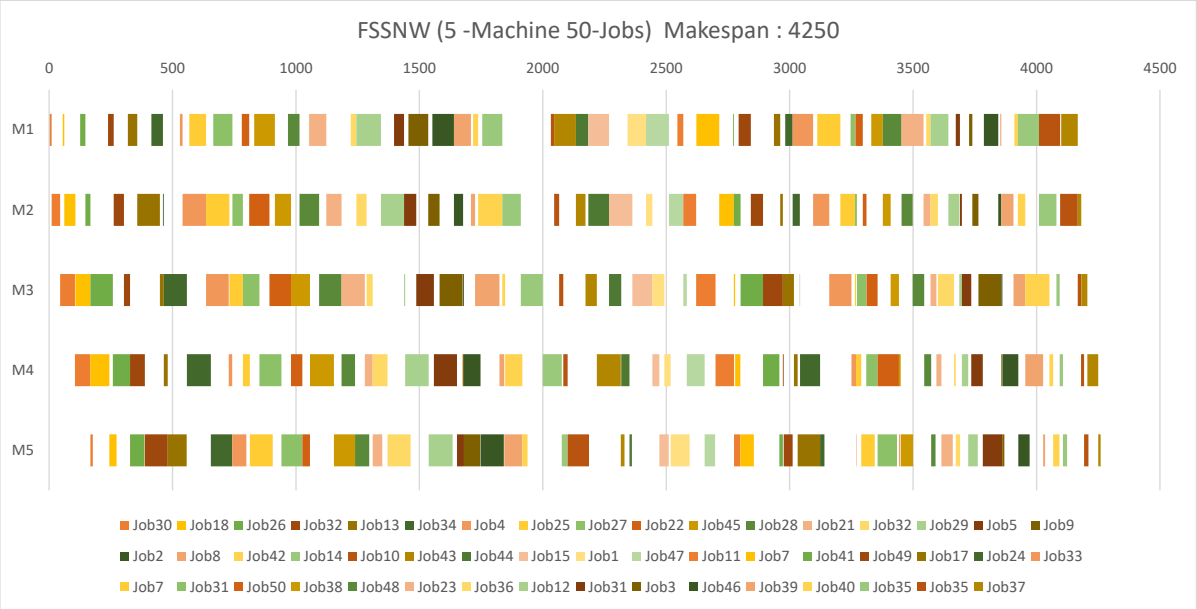

Figure 3.13: FSSNW with 5 machines and 50 jobs

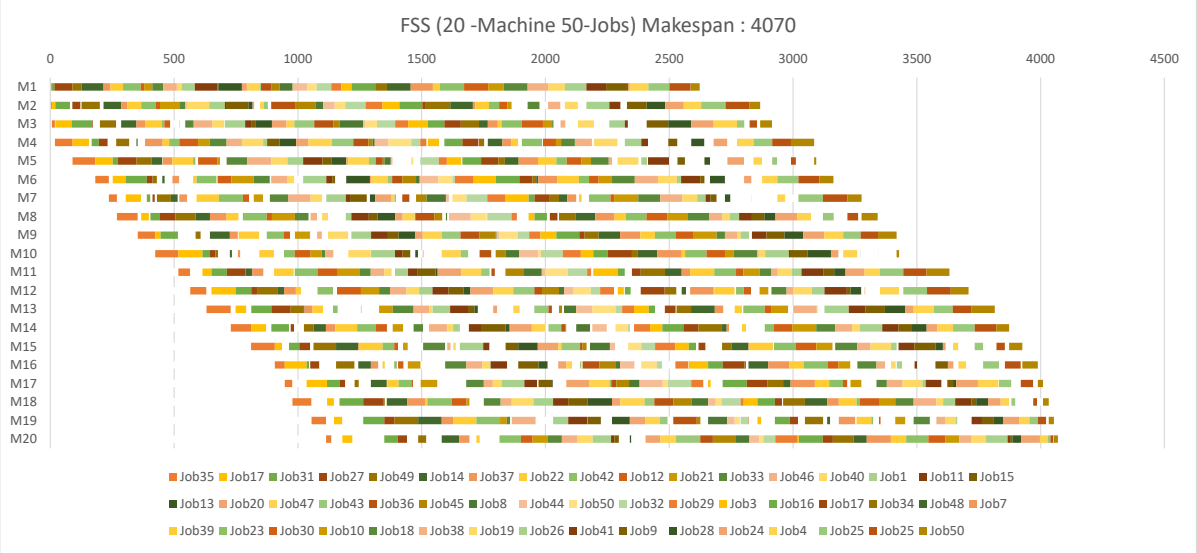

Figure 3.14: FSS with 20 machines and 50 jobs

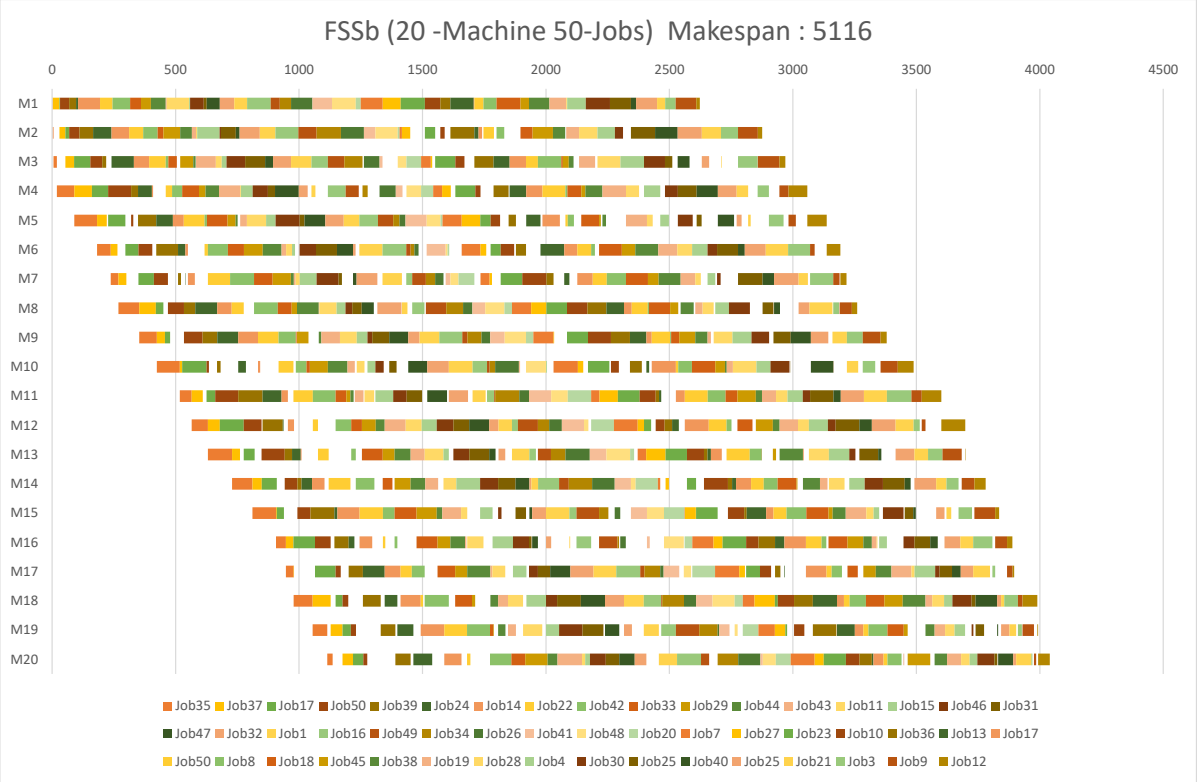

Figure 3.15: FSSB with 20 machines and 50 jobs

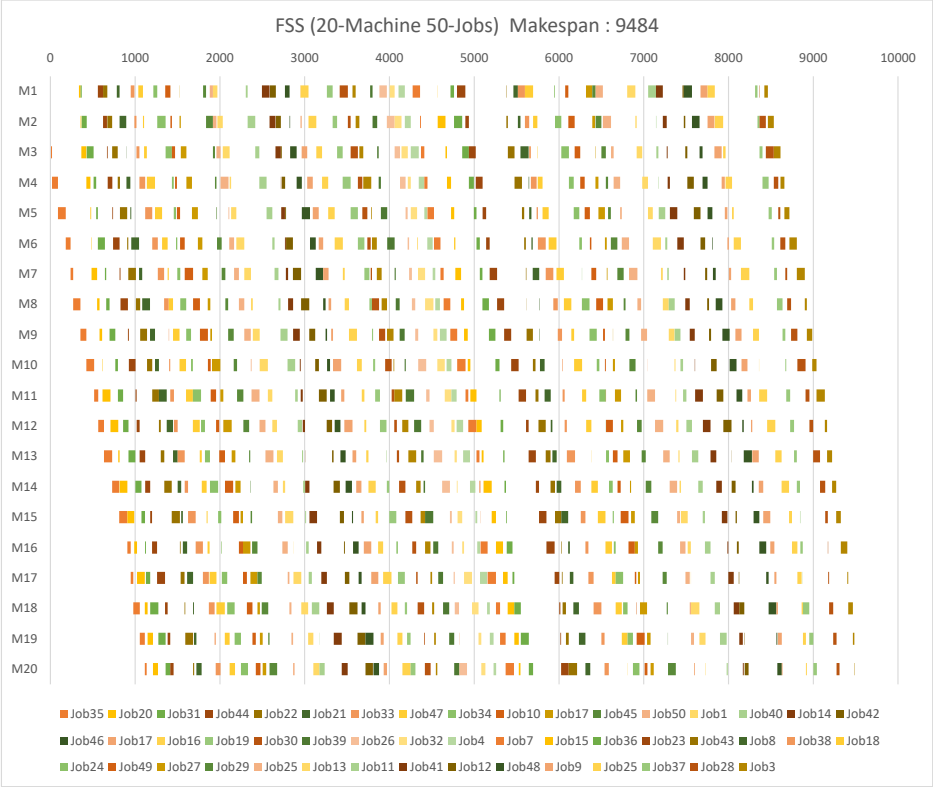

Figure 3.16: FSSNW with 20 machines and 50 jobs

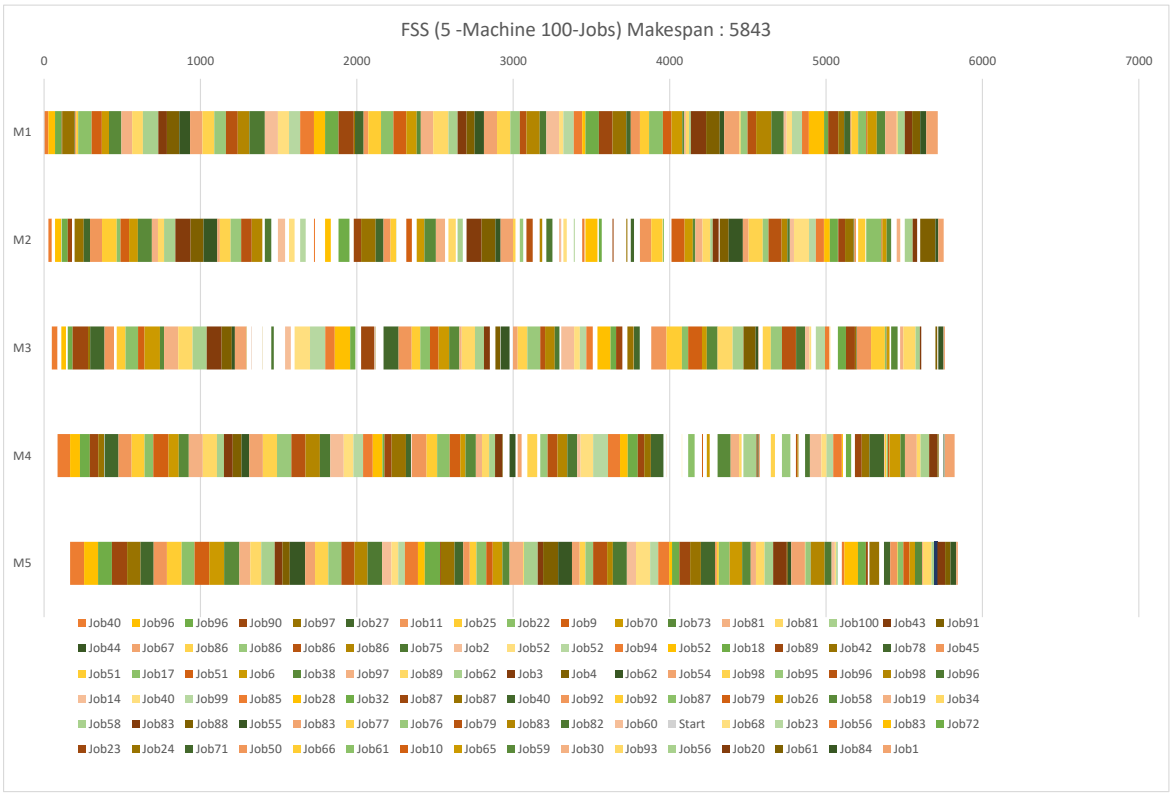

Figure 3.17: FSSB with 5 machines and 100 jobs

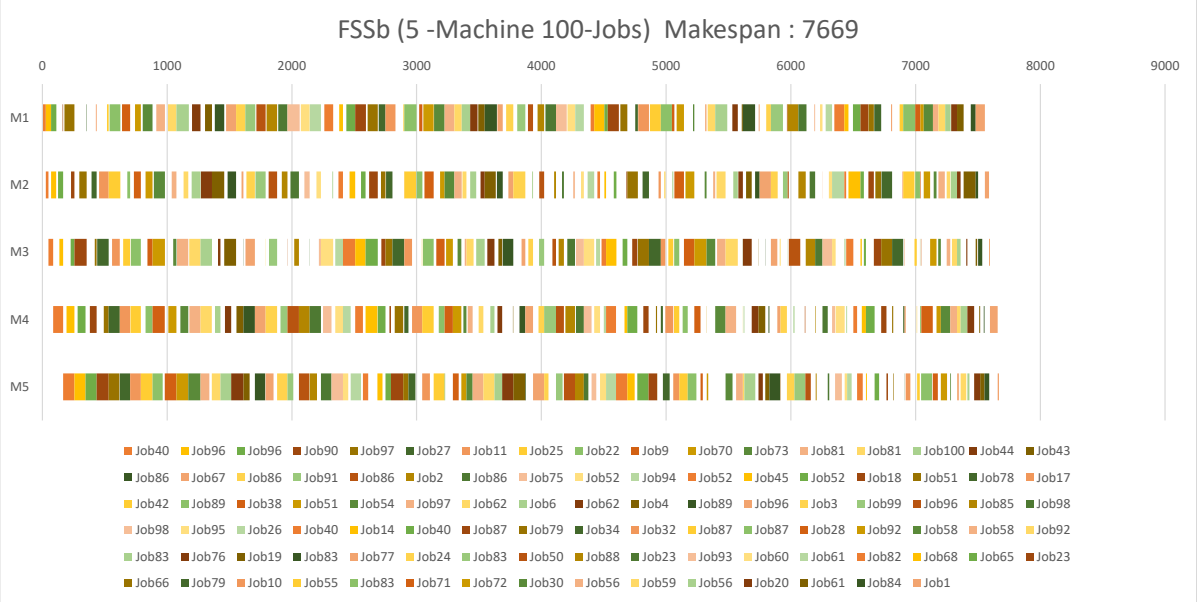

Figure 3.18: FSSB with 5 machines and 100 jobs

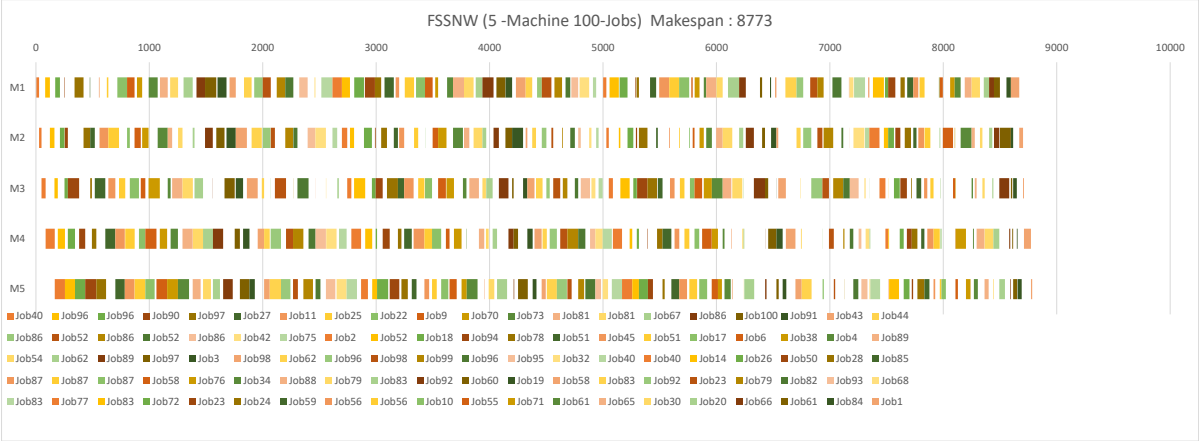

Figure 3.19: FSSNW with 5 machines and 100 jobs