Project 4 Analysis

Mathew Dela Cruz

6-6-2022

Contents

1 Introduction

This is an analytical report on the NEH algorithm performed on two variants of scheduling problems. The flowshop (FSS) and flowshop with blocking (FSSB) problems were used as constraints in this experimentation with 120 problem instances from the Tailard data sets. Both scheduling problems are methods of finding a makespan of a dataset that include a certain number of machines and job process times on each machine. The NEH algorithm as well as the scheduling problems we implemented in Java. The Tailard data sets can be viewed here. Included in this dataset is a list of 120 problem instances with differing sizes ranging from 5 machines and 20 jobs to 20 machines and 500 jobs.

2 The NEH Heuristic Algorithm

The NEH Heuristic algorithm is an Algorithm that seeks to find a schedule of jobs on machines that results in a minimum makespan. In this experimentation, FSS and FSSB are used as constraints when calculating the makespan of a given dataset. This algorithm was developed by Nawaz, Enscore, and Ham in 1983. This algorithm has also been deemed to be the best heuristic to solve the flow shop scheduling problem. First, the algorithm orders the job sequence in decreasing order based on the total process time of each job individually. Then the algorithm works recursively in order to find the optimal sequence of n jobs by finding the sequence with the minimal makespan of the first two jobs in a dataset. Once that sequence is determined, the order is retained and the algorithm adds the next job in the list and goes through each permutation to find a minimal makespan again. The minimal makespan order sequence is retained again and the next job is added up until n jobs. When the algorithm is finished, a job sequence is returned that results in the minimal makespan of process times. (Note: these job sequences were recorded in the "jobSchedule.csv file")

3 Results

In the following tables, experimentation results have been recorded. These results were compiled in Java, into a CSV file, and then recorded again in these tables. For the sake of fitting in to pages, the results of the 120 problem instances have been split up into three tables. For each instance, m denotes the number of machines and n the number of jobs. FSS results were found with the Flow Shop Scheduling constraints and FSSB with the Flow Shop Scheduling with Blocking constraints respectively. Runtimes in milliseconds for each instance was also recorded below.

Table 1: Results (1-40)

			FSS		FSSB	
Instance	m	n	Makespan	Runtime	Makespan	Runtime
1	5	20	1286	63	1435	7
2	5	20	1365	60	1477	7
3	5	20	1132	48	1353	7
4	5	20	1325	44	1535	8
5	5	20	1178	52	1348	8
6	5	20	1228	48	1464	7
7	5	20	1201	55	1424	6
8	5	20	1203	47	1447	7
9	5	20	1292	47	1457	5
10	5	20	1146	47	1341	5
11	10	20	1578	57	1748	10
12	10	20	1729	54	1903	10
13	10	20	1562	52	1719	15
14	10	20	1433	47	1622	8
15	10	20	1502	52	1726	9
16	10	20	1450	58	1656	12
17	10	20	1562	65	1738	9
18	10	20	1609	51	1814	9
19	10	20	1661	58	1779	9
20	10	20	1617	52	1839	7
21	20	20	2375	53	2510	17
22	20	20	2137	61	2278	15
23	20	20	2411	51	2564	13
24	20	20	2264	58	2399	11
25	20	20	2397	54	2538	11
26	20	20	2312	54	2428	14
27	20	20	2383	53	2498	11
28	20	20	2259	51	2402	13
29	20	20	2313	63	2421	13
30	20	20	2277	62	2425	12
31	5	50	2775	71	3387	33
32	5	50	2850	73	3475	27
33	5	50	2640	82	3244	26
34	5	50	2737	89	3359	25
35	5	50	2868	77	3431	24
36	5	50	2664	73	3345	29
37	5	50	2713	71	3206	29
38	5	50	2677	100	3279	27
39	5	50	2617	75	3118	31
40	5	50	2803	77	3445	29

Table 2: Results (41-80)

			FSS		FSSB	
Instance	m	n	Makespan	Runtime	Makespan	Runtime
41	10	50	3166	78	3908	37
42	10	50	2978	79	3732	39
43	10	50	2950	86	3741	45
44	10	50	3198	78	3938	36
45	10	50	3129	86	3948	45
46	10	50	3219	80	3857	40
47	10	50	3277	85	3928	61
48	10	50	3191	94	3775	55
49	10	50	3021	98	3817	36
50	10	50	3202	106	3943	54
51	20	50	4082	127	4881	60
52	20	50	3921	100	4597	107
53	20	50	3827	123	4526	76
54	20	50	3875	90	4541	59
55	20	50	3833	104	4470	58
56	20	50	3962	96	4545	54
57	20	50	3994	81	4614	54
58	20	50	3958	91	4521	82
59	20	50	3952	93	4575	49
60	20	50	4079	85	4802	82
61	5	100	5501	146	6690	78
62	5	100	5284	106	6494	63
63	5	100	5155	111	6359	94
64	5	100	5028	118	6185	73
65	5	100	4845	109	6322	62
66	5	100	5006	115	6430	67
67	5	100	5297	141	6501	86
68	5	100	5188	108	6327	74
69	5	100	5414	116	6593	95
70	5	100	5321	104	6616	72
71	10	100	5846	149	7607	130
72	10	100	5410	136	7287	92
73	10	100	5781	146	7401	92
74	10	100	5983	143	7576	116
75	10	100	5663	140	7322	89
76	10	100	5375	139	7343	88
77	10	100	5668	124	7294	89
78	10	100	5725	131	7356	142
79	10	100	5958	135	7616	151
80	10	100	5934	124	7534	84

Table 3: Results (81-120)

			FSS		FSSB	
Instance	m	n	Makespan	Runtime	Makespan	Runtime
81	20	100	6594	213	8284	139
82	20	100	6506	180	8305	133
83	20	100	6633	224	8371	208
84	20	100	6506	189	8311	202
85	20	100	6585	318	8282	249
86	20	100	6664	174	8409	160
87	20	100	6528	172	8469	231
88	20	100	6739	187	8568	141
89	20	100	6692	163	8307	195
90	20	100	6715	154	8456	139
91	10	200	10942	459	14423	460
92	10	200	10787	370	14396	548
93	10	200	10990	371	14382	536
94	10	200	11057	370	14427	472
95	10	200	10568	401	14427	505
96	10	200	10518	369	14234	528
97	10	200	10981	395	14569	494
98	10	200	10838	397	14534	491
99	10	200	10616	385	14276	471
100	10	200	10754	369	14382	504
101	20	200	11668	680	15575	749
102	20	200	11668	639	15793	732
103	20	200	11784	642	15783	727
104	20	200	11748	648	16001	681
105	20	200	11685	613	15795	769
106	20	200	11593	628	15834	719
107	20	200	11855	631	15898	740
108	20	200	11755	665	15724	696
109	20	200	11735	611	16015	736
110	20	200	11797	642	15870	684
111	20	500	26754	6102	38517	8591
112	20	500	27334	6591	38538	9534
113	20	500	26973	6259	37993	8591
114	20	500	27007	6203	38406	8565
115	20	500	26850	6174	38243	8575
116	20	500	27040	6430	38480	8819
117	20	500	26885	6214	38124	8566
118	20	500	27138	6325	38338	9005
119	20	500	26728	6170	38104	8437
120	20	500	26975	6349	38525	8674

4 Conclusion

The first clear observation through this experimentation is that the larger the instance size, the longer the runtime to solve the problem is ranging from 6ms to 9000+ms. Another notable observation about runtime however is that in most early instances where the scale of the data was lesser, the runtime of the FSS variant was longer than FSSB. As the scale increases, these runtimes become more similar with the medium scale size in this dataset with less deviation between each other. Then towards the end of experimentation the run time of the FSSB variant becomes longer than FSS. These time deviations is correlated to how each variant determines makespan time as the calculation method is what what sets these two variants apart. Another obsevation can be made with the calculated makespan of the two variants. Again, as the scale of the problem instance increases the deviation between the two resulting makespans of the optimal jobs sequence increases as well. The FSS variant proves to calculate a much lesser makespan of process times as the scale increases.