Operations Research – Project

VŠB-TUO, ZS 2014/2015

Implementation of Artificial Bee Colony Algorithm to Solve No-Wait Flow Shop Problem

Josef Raška (RAS0029)

Introduction

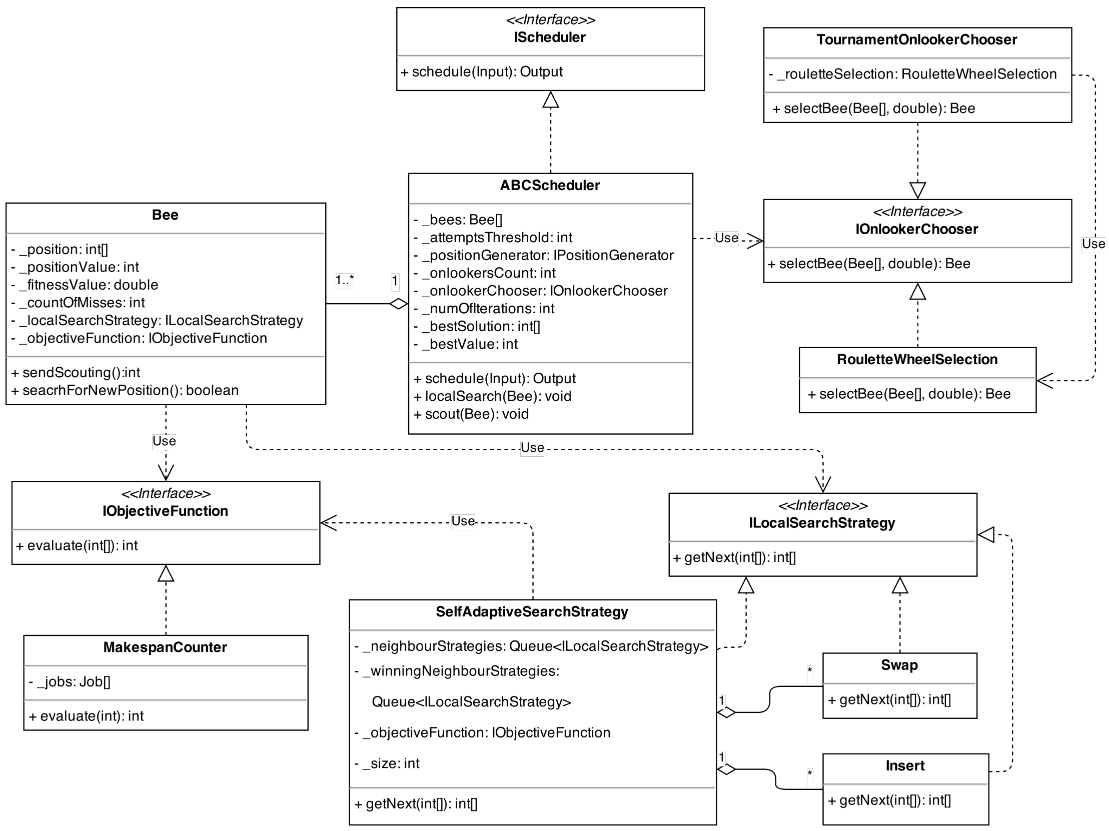
This document is intended as documentation and test results presentation for semester project of implementation Artificial Bee Colony in Java to solve No-Wait Flow Shop scheduling problem.

Implementation

You can find all the source codes in directory src, where the application source code is in directory main and unit tests source codes are in directory test. There are pretty much tests for the implementation so all the parts should be correct.

For our problem of No-Wait Flow Shop with given input, search space is the order of execution obtained job to minimize the makespan. Thus for *n* jobs, the possible solution is each permutation of numbers 0 to *n*-1, where the permutation defines the index of jobs.
In Java we can represent this as an array of integers.
The input will be represented with Java class Input, containing all jobs requested to schedule. The output is the best found permutation discovered by algorithm and some other statistics information like best/worst found value, standard deviation and average execution time for one execution of algorithm for problem instance.

All the parts of the algorithm are represented as objects and most of them covered by interface, to allow easily change parts of algorithm. Most important class is ABCScheduler, where is implemented the ABC algorithm itself. Now follows UML diagram of most important classes of implementation.



As we can see, ABCScheduler uses Bee objects to search feasible solutions space, hold their positions and IOnlookerChooser to choose bees for onlookers after bees searching. Other fields are intended as parameters for the algorithm, which the scheduler accepts in constructor. Objective function implementation is counter of makespan, which simply counts makespan for the given permutation of jobs, since it holds information about jobs as its field. As mentioned before, if we want to change the optimization for makespan for something different, we just need to implement the interface IObjectiveFunction.

It is similar for the local search strategy. Swap and Insert are simply and self-describing modification of jobs permutation. SelfAdaptiveSearchStrategy uses previous two and tries to adapt to given algorithm as described in [3].
Onlookers are selected with IOnlookerChooser implementation TournamentOnlookerChooser, which internally uses Roulette Wheel Selection algorithm to pick two bees and then selecting the one with better found solution also as described in [3].

Near these core classes, there are other classes holding input and output, performing reading of input and exporting the results to html for now.

Executable classes

There are four executable classes, which contains Java main method.

Class Program accepts the path to the input file as the only one parameter and exports the result to the standard output. This class is also set as main class after project build.
There are three other classes which can be made as executable.
AllInDirSchedule is there to apply algorithm on all text files named as numbers in directory taken as parameter, writing its output to file output.html in working directory, rewriting the old one if exists.
DebugSchedule, which behaves the same as Program, but writes all the steps of ABC algorithm to output in advance.

The last one is BruteForceComparison, which tries to solve the problem with brute force to compare the result of the ABC algorithm against best possible solution. ABC algorithm runs 30 times for each of the input instance.

After the build described below, you can run main program for example with this command:

```
java -jar build/libs/Flowshop-with-No-Wait-ABC-1.0.jar data/41.txt
```

Build

Gradle is used as the build system.

After cloning the repository, you can build the jar with the default Program executable on any machine with executing `gradlew build`, which will use Gradle Wrapper, download the Gradle to your computer if not present, builds the jar and also executes the tests. Jar can be found in path /build/libs.

Test Results

There were 30 runs of algorithm on every instance of the provided input data. The results were obtained with settings of scheduler like this:
50 bees, 50 Onlookers, 1000 iterations, 10 misses as the threshold for bee becoming scout, SelfAdaptiveSearchStrategy and TournamentOnlookerChooser used.

File: 41.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5256	5175	5319	5253	5183	5257	5207	5158	5284	5211	5286	5182	5197	5197	5219
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5190	5251	5200	5280	5111	5252	5303	5227	5277	5224	5261	5192	5252	5219	5285

Best result	5111
Worst result	5319
Standard deviation	46.733
Average time	0.543 s

File: 42.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5098	5079	5064	5036	5127	5128	5182	5071	5127	5061	5115	5099	5124	5121	5057
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5159	5114	5103	5092	5027	5119	5126	5103	5044	5030	5094	5108	5118	5119	5062

Best result	5027
Worst result	5182
Standard deviation	36.678
Average time	0.577 s

File: 43.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5254	5330	5313	5336	5332	5240	5359	5350	5291	5336	5296	5242	5160	5224	5276
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5253	5267	5207	5378	5335	5322	5319	5229	5276	5363	5374	5312	5335	5288	5299

Best result	5160
Worst result	5378
Standard deviation	52.317
Average time	0.595 s

File: 44.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5353	5330	5381	5384	5335	5411	5351	5353	5432	5317	5309	5291	5306	5304	5336
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5314	5294	5274	5321	5310	5372	5272	5336	5310	5288	5326	5339	5331	5412	5282

Best result	5272
Worst result	5432
Standard deviation	40.359
Average time	0.593 s

File: 45.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5453	5333	5535	5516	5373	5456	5491	5414	5502	5444	5363	5418	5488	5459	5458
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5530	5440	5398	5491	5417	5491	5481	5459	5392	5512	5422	5498	5419	5525	5502

Best result	5333
Worst result	5535
Standard deviation	51.751
Average time	0.566 s

File: 46.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5321	5226	5198	5294	5191	5283	5238	5298	5290	5266	5238	5271	5184	5250	5248
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5296	5164	5139	5261	5291	5278	5278	5222	5185	5250	5282	5257	5175	5261	5273

Best result	5139
Worst result	5321
Standard deviation	45.223
Average time	0.578 s

File: 47.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5493	5556	5577	5535	5609	5528	5539	5586	5584	5606	5484	5582	5538	5592	5634
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5574	5379	5577	5622	5530	5568	5542	5578	5582	5521	5507	5627	5507	5519	5553

Best result	5379
Worst result	5634
Standard deviation	50.815
Average time	0.595 s

File: 48.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5308	5224	5279	5236	5316	5311	5232	5285	5313	5241	5195	5277	5210	5248	5256
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5298	5334	5263	5243	5257	5259	5326	5297	5287	5275	5259	5266	5270	5184	5237

Best result	5184
Worst result	5334
Standard deviation	37.069
Average time	0.608 s

File: 49.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5253	5347	5328	5168	5320	5313	5220	5281	5354	5318	5318	5322	5301	5314	5304
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5378	5188	5292	5329	5287	5237	5283	5318	5340	5235	5253	5296	5273	5257	5219

Best result	5168
Worst result	5378
Standard deviation	48.824
Average time	0.595 s

File: 50.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	5396	5463	5368	5422	5396	5322	5473	5480	5374	5424	5386	5427	5431	5339	5394
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	5466	5467	5401	5297	5380	5366	5283	5316	5424	5450	5379	5402	5339	5365	5475

Best result	5283
Worst result	5480
Standard deviation	53.468
Average time	0.61 s

File: 51.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7562	7603	7556	7679	7689	7652	7510	7612	7658	7575	7590	7522	7460	7571	7634
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7710	7662	7530	7520	7434	7616	7600	7625	7514	7529	7528	7640	7553	7618	7593

Best result	7434
Worst result	7710
Standard deviation	66.043
Average time	0.973 s

File: 52.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7036	7237	7313	7141	7295	7228	7024	7228	7130	7139	7095	7171	7199	7281	7234
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7184	7280	7255	7157	7142	7222	7248	7236	7198	7252	7278	7106	7103	7210	7233

Best result	7024
Worst result	7313
Standard deviation	73.21
Average time	1.13 s

File: 53.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7457	7476	7395	7479	7504	7434	7315	7579	7530	7477	7567	7518	7457	7410	7559
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7477	7464	7550	7461	7520	7478	7526	7561	7431	7353	7193	7462	7450	7471	7384

Best result	7193
Worst result	7579
Standard deviation	79.748
Average time	1.175 s

File: 54.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7381	7274	7250	7231	7151	7181	7290	7319	7242	7373	7313	7359	7311	7413	7185
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7315	7269	7210	7308	7164	7205	7174	7331	7153	7376	7305	7355	7237	7288	7283

Best result	7151
Worst result	7413
Standard deviation	72.887
Average time	0.956 s

File: 55.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7582	7376	7358	7489	7593	7588	7345	7425	7494	7468	7338	7449	7355	7470	7574
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7537	7396	7223	7478	7506	7512	7317	7437	7420	7308	7422	7311	7541	7356	7496

Best result	7223
Worst result	7593
Standard deviation	94.5
Average time	0.954 s

File: 56.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7563	7387	7362	7373	7558	7327	7306	7445	7479	7505	7442	7474	7403	7454	7500
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7461	7454	7570	7387	7399	7503	7491	7510	7362	7352	7335	7472	7259	7489	7449

Best result	7259
Worst result	7570
Standard deviation	77.313
Average time	0.96 s

File: 57.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7260	7410	7366	7333	7294	7298	7376	7258	7347	7296	7300	7266	7237	7062	7299
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7258	7276	7177	7323	7069	7393	7360	7421	7389	7361	7334	7257	7267	7234	7300

Best result	7062
Worst result	7421
Standard deviation	83.265
Average time	0.948 s

File: 58.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7479	7376	7333	7367	7300	7299	7348	7453	7408	7206	7404	7421	7446	7392	7388
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7468	7357	7208	7390	7404	7449	7344	7458	7404	7364	7361	7439	7461	7334	7318

Best result	7206
Worst result	7479
Standard deviation	67.637
Average time	0.946 s

File: 59.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7283	7290	7241	7375	7205	7184	7368	7310	7413	7301	7323	7255	7404	7353	7221
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7357	7327	7353	7363	7387	7288	7266	7267	7460	7322	7394	7385	7172	7374	7418

Best result	7172
Worst result	7460
Standard deviation	72.124
Average time	0.96 s

File: 60.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7324	7388	7411	7376	7306	7353	7366	7329	7309	7218	7474	7449	7259	7428	7292
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7218	7310	7279	7328	7359	7269	7463	7442	7371	7284	7410	7386	7304	7388	7371

Best result	7218
Worst result	7474
Standard deviation	67.637
Average time	0.945 s

File: 61.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	8245	8223	8223	8142	8285	8145	8321	8214	8299	8227	8188	8112	8217	8166	8180
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	8256	8199	8161	8296	8287	8186	8194	8194	8216	8152	8298	8116	8156	8226	8266

Best result	8112
Worst result	8321
Standard deviation	56.162
Average time	0.735 s

File: 62.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7927	8130	8086	8056	8145	8098	8104	8130	7977	8073	8061	8163	8080	8179	8132
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	8141	8140	8138	8068	8184	8102	7974	8130	8097	8108	8091	8128	8161	8122	8115

Best result	7927
Worst result	8184
Standard deviation	57.683
Average time	0.749 s

File: 63.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7896	7880	7826	7843	7841	7871	7792	7790	7872	7789	7795	7858	7763	7858	7747
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7729	7931	7758	7822	7725	7818	7720	7691	7816	7634	7898	7792	7764	7896	7844

Best result	7634
Worst result	7931
Standard deviation	67.734
Average time	0.745 s

File: 64.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7876	7880	7821	7827	7796	7774	7733	7805	7793	7849	7816	7863	7869	7809	7831
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7901	7793	7749	7775	7861	7832	7862	7695	7837	7741	7867	7821	7828	7784	7736

Best result	7695
Worst result	7901
Standard deviation	49.344
Average time	0.74 s

File: 65.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	8042	8097	8052	8141	8011	8098	8141	7956	8094	8186	8087	8048	8147	8147	8209
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	8072	8111	8096	8059	8027	8088	8034	8079	7984	8113	8048	8172	8073	8073	8034

Best result	7956
Worst result	8209
Standard deviation	56.801
Average time	0.742 s

File: 66.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7986	8003	8026	8032	8084	8016	8075	8013	8031	7942	8064	8017	8108	7987	7985
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	7986	7850	7905	7967	8099	7872	7937	7869	7908	7984	8102	8002	8066	8024	7973

Best result	7850
Worst result	8108
Standard deviation	68.42
Average time	0.757 s

File: 67.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	8113	8079	8148	8077	8153	8116	8090	8165	8171	8090	8177	8123	8134	8155	7976
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	8158	8158	8214	8156	8220	8073	8170	8192	8174	8147	8116	8094	8179	8142	8011

Best result	7976
Worst result	8220
Standard deviation	53.484
Average time	0.737 s

File: 68.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	7974	8010	7978	7922	7934	7991	7977	7953	7944	7994	7856	7985	7879	8052	7926
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	8001	7989	8020	7868	7865	8004	7942	7914	7942	8024	7916	7894	7851	8028	7996

Best result	7851
Worst result	8052
Standard deviation	55.075
Average time	0.745 s

File: 69.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	8283	8279	8255	8406	8304	8319	8365	8336	8342	8201	8259	8366	8339	8296	8309
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	8208	8429	8275	8321	8282	8294	8395	8274	8200	8308	8383	8293	8356	8372	8249

Best result	8200
Worst result	8429
Standard deviation	57.823
Average time	0.742 s

File: 70.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	8413	8256	8423	8381	8438	8385	8324	8415	8323	8332	8337	8364	8352	8207	8282
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	8249	8376	8144	8294	8277	8312	8329	8230	8307	8336	8371	8346	8419	8269	8300

Best result	8144
Worst result	8438
Standard deviation	67.6
Average time	0.724 s

File: 71.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10809	10897	10973	10866	10825	10934	10891	10898	10549	10767	10729	10862	10781	10812	10723
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10798	10812	10794	10970	10872	10899	10832	10754	10757	10804	10875	10811	10920	10907	10791

Best result	10549
Worst result	10973
Standard deviation	84.197
Average time	1.176 s

File: 72.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10715	10644	10738	10656	10702	10737	10769	10518	10618	10756	10654	10706	10674	10753	10665
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10567	10690	10724	10710	10710	10752	10762	10641	10744	10594	10711	10651	10694	10760	10649

Best result	10518
Worst result	10769
Standard deviation	60.41
Average time	1.171 s

File: 73.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10607	10836	10900	10679	10839	10789	10800	10845	10855	10782	10651	10689	10707	10691	10770
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10768	10840	10784	10761	10846	10791	10782	10893	10747	10780	10833	10735	10899	10689	10746

Best result	10607
Worst result	10900
Standard deviation	73.144
Average time	1.174 s

File: 74.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10980	11095	11341	11247	11330	11089	11141	11263	11023	10991	11095	11113	11013	11247	11171
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	11249	11086	11096	11042	11321	11130	11094	11203	11114	11235	11093	11200	11223	11270	10970

Best result	10970
Worst result	11341
Standard deviation	105.14
Average time	1.233 s

File: 75.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10790	10714	10736	10744	10821	10749	10842	10328	10533	10830	10714	10705	10775	10780	10698
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10833	10717	10814	10812	10588	10805	10704	10733	10760	10772	10856	10461	10732	10737	10865

Best result	10328
Worst result	10865
Standard deviation	115.795
Average time	1.183 s

File: 76.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10610	10726	10629	10529	10783	10795	10694	10845	10712	10702	10678	10585	10707	10795	10762
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10529	10691	10614	10720	10752	10796	10757	10673	10644	10578	10603	10712	10703	10727	10752

Best result	10529
Worst result	10845
Standard deviation	79.487
Average time	1.169 s

File: 77.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10885	10992	10968	10855	10734	10813	10682	11010	11054	10900	10966	10854	10769	10867	10866
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10861	10849	10981	10923	10978	10801	10748	11052	10819	10831	10792	10814	10902	10709	11008

Best result	10682
Worst result	11054
Standard deviation	99.059
Average time	1.142 s

File: 78.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10544	10318	10472	10441	10517	10514	10561	10498	10439	10449	10452	10494	10271	10317	10439
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10450	10484	10361	10246	10413	10582	10447	10408	10442	10578	10486	10573	10435	10605	10523

Best result	10246
Worst result	10605
Standard deviation	88.209
Average time	1.155 s

File: 79.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	10919	10720	10907	10743	10902	10471	10676	10776	10764	10858	10768	10691	10673	10795	10709
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10836	10850	10648	10722	10774	10861	10904	10750	10781	10721	10653	10780	10752	10746	10877

Best result	10471
Worst result	10919
Standard deviation	95.001
Average time	1.135 s

File: 80.txt

Sequence no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Makespan	11042	10969	11027	11002	10832	10780	11000	10896	11027	11027	10895	11051	11019	11012	10901
Sequence no.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Makespan	10944	11007	10930	10926	10936	10954	11031	11005	10864	11076	10949	10993	10972	10932	11050

Best result	10780
Worst result	11076
Standard deviation	68.734
Average time	1.174 s

There were also made few experiments and comparisons with different parameters and various local search strategies. In compare with the direct scheduling of jobs in order as there were in input files, ABC algorithm resulted in significantly better results and since the computed makespans for all runs for one problem instance tend to be close to each other, we can evaluate the implementation as working. Due to high degree of modularity of algorithm, every part can be easily changed and a lot of experiments with search strategies, onlooker selecting, initial solutions etc. can be made.

Conclusion

We presented our implementation of the ABC algorithm in Java. ABC algorithm is powerful and easily customizable algorithm and this idea also tries to go through the implementation. The obtained result are promising and there is also a huge space for improvement as in case for new algorithms for certain parts, then for performance optimization since the easiest way to get better results is to increase the number of iterations.
Through this project, I became quite familiar with the ABC algorithm and it became next great part of my knowledge base, which I consider as the most important impact of the project.

References

- [1] Karabonga, D. , Artificial Bee Colony Algorithm Homepage http://mf.erciyes.edu.tr/abc/
- [2] Karaboga, D., *Artificial Bee Colony Algorithm,* Scholaropedia 2010, http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm
- [3] Quan-Ke Pan, M. Fatih Tasgetiren, P.N. Suganthan, T.J. Chua, *A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem,* Elsevier Inc. 2011 http://www.sciencedirect.com/science/article/pii/S0020025509005684