Benchmarks for MISTA 2007 paper

Daniel Godard
ILOG, 9 rue de Verdun, 94253 Gentilly, France, dgodard@ilog.fr
Philippe Laborie
ILOG, 9 rue de Verdun, 94253 Gentilly, France, plaborie@ilog.fr

---

This paper describes the set of benchmarks used in the companion MISTA-07 paper. Note that all the problems studied below are NP-complete.

Keywords: Constraint Logic Programming, Heuristic Search, Local Search.

---

1 Introduction

We give in this paper some references about the different benchmarks used in the computational experiments of our MISTA-07 paper.

2 Job-shop scheduling

BENCH NAME: JobShop

2.1 Benchmark description

- Fisher-Thompson (ft*): [16]
- Carlier (car*): [13]
- Lawrence (la*): [26]
- Adams-Balas-Zawack (abz*): [1]
- Yamada-Nakano (yam*): [42]
- Taillard (tail*): [38]
- Applegate-Cook (orb*): [2]
- Storer-Wu-Vaccari (swv*): [37]

3 Resource-constrained project scheduling

BENCH NAME: RCPSP

3.1 Benchmark description

- PSP Lib: [24]
- Baptiste-LePape: [5]
- Patterson: [30]
- Carlier-Neron: ?

3.2 State-of-the-art

[23]

4 Cumulative job-shop

BENCH NAME: CumulativeJobShop

4.1 Benchmark description

Use instances from [28]. We selected duplicated open instances.

Problem	Lower bound	Upper bound
la03d	593	593
la04d	572	576
la16d	892	925
la17d	754	755
la18d	803	811
la19d	756	795
la20d	849	859
la21d	1017	1034
la22d	913	926
la24d	885	903
la25d	907	952
la29d	1117	1131
la36d	1229	1250
la37d	1378	1397
la38d	1092	1175
la39d	1221	1226
la40d	1180	1205
ft10d	837	891

Table 1: Cumulative job-shop

4.2 State-of-the-art

[18]

5 Semiconductor testing

BENCH NAME: SemiconductorTesting

5.1 Benchmark description and state-of-the-art

We randomly selected 18 instances among the biggest ones described in [29]. The selected instances are described in the table below.

Problem	Upper bound
i305-306	2028
i305-307	2674
i305-317	2026
i315-302	4398
i315-313	2587
i315-320	2770
i325-304	3185
i325-307	2371
i325-319	1972
i605-306	2654
i605-312	4695
i605-316	5197
i615-308	1942
i615-309	4591
i615-317	2479
i625-301	2712
i625-310	1503
i625-311	3689

Table 2: Semiconductor testing

6 Flow-shop with earliness/tardiness costs

BENCH NAME: FlowShopEarliTardi

6.1 Benchmark description

Use instances from Morton and Pentico: 12 instances. Total running time is 8430 seconds. Room to apply 2 to 3 different seeds.

[27]

6.2 State-of-the-art

[41, 15]

7 Resource-constrained project scheduling with earliness/tardiness costs

BENCH NAME: RCPSPEarliTardi

7.1 Benchmark description and state-of-the-art

We take the 5400 instances of the paper.

We randomly select 30 instances for which the approach in [40] finds the optimal value within 30s whereas and our approach does not and 30 instances for which the approach in [40] did not prove optimality.

Problem	Size	Lower bound	Upper bound
jb1	10x3	0.191	0.191
jb2	10x3	0.137	0.137
jb4	10x5	0.568	0.568
jb9	15x3	0.333	0.333
jb11	15x5	0.213	0.213
jb12	15x5	0.190	0.190
ljb1	30x3	0.215	0.215
ljb2	30x3	0.230	0.508
ljb7	50x5	0.0.058	0.110
ljb9	50x5	0.124	0.739
ljb10	50x8	0.197	0.512
ljb12	50x8	0.053	0.399

Table 3: Flow-shop with earliness/tardiness costs

We then run these 60 instances during 300s both with the approach in the paper and with our approach and compare the results.

[40]

8 Air land scheduling

BENCH NAME: AirLand

8.1 Benchmark description and state-of-the-art

Take the first 8 instances (upon 13) from Beasley as we do not have results for the other ones to compare to.

[6]

9 Open-shop scheduling

BENCH NAME: OpenShop

9.1 Benchmark description

We use instances of:

- Brucker: [10]
- Taillard: [38]
- Gueret-Prins: [19]

For each type of instance, we select the biggest problems, thus, for Brucker: $j8*$, for Taillard: tai_20x20_*, for Gueret-Prins: $gp10 - *$.

It results in 28 instances.

9.2 state-of-the-art

We compare to the best results between the ones mentioned in [8] and the ones in [25]. We use a time-limit equal to half of the time limit used in [8] to cope with the fact Blum worked on a 1100 MHz PC.

10 Quality maximization resource-constrained project scheduling

BENCH NAME: QMRCPSP

10.1 Benchmark description and state-of-the-art

We use the 3600 instances described in [32] and compare with their results using a time-limit of 30s for each instance.

11 Trolley scheduling

BENCH NAME: Trolley

11.1 Benchmark description

We automatically generated some instances of the scheduling problem described in [39]. These instances are generated by extending some classical job-shop problems adding the trolley resource with some transition times for transporting the items from one machine to the other. They are listed on Table 4. We compare with OPL search results.

Original problem	Size of trolley instance
abz5	430
abz6	430
orb2	430
orb7	430
orb9	430
la04	230
la18	430
la19	430
la20	430
ft10	430
ft20	460
e0ddr2-1	230
car5	270
car6	312
car8	280

Table 4: Trolley instances

12 Job-shop scheduling with earliness/tardiness costs

BENCH NAME: JobShopEarliTardi

Reference from Philippe Baptiste, Marta Flamini and Francis Sourd in 2005. Selected instances with 15 jobs and 20 jobs as we don't have UB for instance with 10 jobs.

12.1 Benchmark description and state-of-the-art

[4]

13 Single-machine scheduling with earliness/tardiness costs

BENCH NAME: SingleMachineEarliTardi

13.1 Benchmark description

- BKY: [12]
- SKS: [35]

For the benchmark SKS, we take only the biggest instances (50 activities) and randomly select among these instances: 10 instances for which the B&B approach in [36] proved the optimality and 10 for which it could not prove the optimality, for these last instances, we compare with the results of the heuristic HEURn described in the same paper.

For the benchmark BKY, we randomly selected 10 instances of the biggest problems with 200 activities and 10 instances with 100 activities and compare with the heuristics HEURn of [36]. The selected instances are described on Table 5 together with the best known upper-bound and lower-bound when available.

13.2 state-of-the-art

[36]

14 Aircraft assembly scheduling

BENCH NAME: AircraftAssembly

14.1 Benchmark description

[17]

14.2 state-of-the-art

[14]

15 Air traffic flow management

BENCH NAME: AirTrafficFlowManagement

Only one instance. No reference. Compare with SetTimesForward.

[21]

16 Flow-shop scheduling

BENCH NAME: FlowShop

16.1 Benchmark description

- Heller: [20]
- Carlier: [13]
- Reeves: [33]
- Taillard: [38]

Use instances from Taillard only (no reference for other instances). 120 instances (20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20, 200x20, 500x20), some are open (50x20, 100x20, 200x20, 500x20). Choose randomly 2 instances of each group having less than 500 activities so total running time does not exceed 6 hours.

17 Single-machine scheduling with common due-date

BENCH NAME: CommonDueDate

17.1 Benchmark description

We randomly selected 10 instances of the problems with 100 activities and 10 instances with 200 activities in the benchmark described in [7]. The selected instances are described on Table 6 together with the best known upper-bound and lower-bound when available.

Instances with 100 activities are run for 600s, instances with 200 activities are run for 1200s.

18 Multi-stage hybrid flow-shop scheduling

BENCH NAME: MultiStageHybridFlowShop

18.1 Benchmark description and state-of-the-art

We randomly select 20 instances from the biggest ones described in [34]. 10 instances with 20 jobs and 10 machines (200 operations), 5 instances with 50 jobs and 10 machines (500 operations) and 5 instances with 100 jobs and 10 machines (1000 operations).

The selected instances are described on Table 7.

19 Single-processor scheduling with total tardiness cost

BENCH NAME: MultiProcessorTotalTardiness

19.1 Benchmark description

We randomly selected 20 instances among the ones described in [22]: 10 among the biggest ones (500 activities), 10 for problems with 200 activities. The selected instances are described on Table 8.

19.2 state-of-the-art

[31]

20 Shop scheduling with set-up times

BENCH NAME: UnaryAlternativeTransitionTime

20.1 Benchmark description

Use instance from Brucker and Thielle: 15 jobshop instances with sequence dependent setup times (5x10x5, 5x15x5, 5x20x10). Other instances of type open-shop and generalized-shop have not been used as we don't have reference. Total runnning time is less than 2 hours. There is room to try up to 3 different seeds.

[11]

20.2 state-of-the-art

[3]

21 Flow-shop with intermediate buffers

BENCH NAME: FlowShopBuffers

21.1 Benchmark description

We randomly selected 30 instances among the ones of size 20x5, 20x10, 20x20, 50x5, 50x10 and 100x5 from [38] and buffer sizes of 0, 1 and 2. The selected instances are described on Table 9. We use the classical time-limit depending on the size of the instance.

We compare to the results of [9].

References

[1] J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34(3):391–401, 1988.

[2] D. Applegate and W. Cook. A computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, 3(2):149–156, 1991.

[3] C. Artigues and D. Feillet. A branch and bound method for the job-shop problem with sequence-dependent setup times. *Annals of Operations Research*, 2006.

[4] P. Baptiste, M. Flamini, and F. Sourd. Lagrangean bounds and lagrangean heuristics for just in time job-shop scheduling. Technical report, Universita degli Studi di Roma Tre, 2005.

[5] P. Baptiste and C. Le Pape. Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems. In *Proc. CP97*, 1997.

[6] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, and D. Abramson. Scheduling aircraft landings - the static case. *Transportation Science*, 34:180–197, 2000.

[7] D. Biskup and M. Feldmann. Benchmarks for scheduling on a single machine against restrictive and unrestrictive common due dates. *Computers and Operations Research*, 28(8):787–801, 2001.

[8] C. Blum. Beam-ACO - hybridizing ant colony optimization with beam search: an application to open-shop scheduling. *Computers and Operations Research*, 32(6):1565–1591, 2005.

[9] P. Brucker, S. Heitmann, and J. L. Hurink. Flow-shop problems with intermediate buffers. *OR Spectrum*, 25(4):549–574, 2003.

[10] P. Brucker, J. Hurink, B. Jurisch, and B. Wöstmann. A branch & bound algorithm for the open-shop problem. *Discrete Applied Mathematics*, 76:43–59, 1997.

[11] P. Brucker and O. Thiele. A branch and bound method for the general shop problem with sequence dependent setup-times. *OR Spektrum*, 18:145–161, 1996.

[12] K. Bulbul, P. Kaminsky, and C. Yano. Preemption in single machine earliness/tardiness scheduling. Submitted for publication, 2001.

[13] J. Carlier. Ordonnancements à contraintes disjonctives. *RAIRO Recherche opérationnelle*, 12:333–351, 1978. In French.

[14] J. Crawford. An approach to resource constrained project scheduling. In *Proc. 1996 Artificial Intelligence and Manufacturing Research Planning Workshop*, 1996.

[15] E. Danna and L. Perron. Structured vs. unstructured large neighborhood search: A case study on job-shop scheduling problems with earliness and tardiness costs. In *Proc. CP 2003*, pages 817–821, 2003.

[16] H. Fisher and G.L. Thompson. Probabilistic learning combinations of local job-shop scheduling rules. In J.F. Muth and G.L. Thompson, editors, *Industrial Scheduling*. Prentice Hall, 1963.

[17] B. Fox and M. Ringer. Planning & scheduling benchmarks, 1995. URL: www.neosoft.com/ benchmrx/.

[18] D. Godard, P. Laborie, and W. Nuijten. Randomized Large Neighborhood Search for Cumulative Scheduling. In *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS 05)*, pages 81–89, 2005.

[19] C. Guéret and C. Prins. A new lower bound for the open-shop problem. *Annals of Operations Research*, 92:165–183, 1999.

[20] J. Heller. Some numerical experiments for an $m \times j$ flow shop and its decision-theoretical aspects. *Operations Research*, 8:178–184, 1960.

[21] ILOG. *ILOG OPL Studio 3.7 User's Manual and Reference Manual*. ILOG, S.A., Gentilly, France, 2003.

[22] B. Kara. SMTTP problem library, http://www.bilkent.edu.tr/ bkara/start.html, 2002.

[23] R. Kolisch and S. Hartmann. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem: An update. *European Journal of Operational Research*, 174:23–37, 2006.

[24] R. Kolisch and A. Sprecher. PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, 96:205–216, 1996.

[25] P. Laborie. Complete MCS-Based Search: Application to Resource Constrained Project Scheduling. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005.

[26] S. Lawrence. *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement)*. Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, 1984.

[27] T. Morton and D. Pentico. *Heuristic Scheduling Systems*. Wiley, 1993.

[28] W. Nuijten. *Time and Resource Constrained Scheduling: A Constraint Satisfaction Approach*. PhD thesis, Eindhoven University of Technology, 1994.

[29] I. M Ovacik and R. Uzsoy. Decomposition methods for scheduling semiconductor testing facilities. *International Journal of Flexible Manufacturing Systems*, 8:357–398, 1996.

[30] J.H. Patterson. A comparison of exact procedures for solving the multiple constrained resource project scheduling problem. *Management Science*, 30(7):854–867, 1984.

[31] M. Pavlin, H. H. Hoos, and T. Stützle. Stochastic local search for multiprocessor scheduling for minimimum total tardiness. In *Proc. of the 16th Canadian Conference on Artificial Intelligence (AI'2003)*, pages 96–113, 2003.

[32] N. Policella, X. Wang, S.F. Smith, and A. Oddi. Exploiting temporal flexibility to obtain high quality schedules. In *Proc. AAAI-2005*, 2005.

[33] C.R. Reeves. A genetic algorithm for flowshop sequencing. *Computers and Operations Research*, 22:5–13, 1995.

[34] F Sivrikaya-Serifolu and G. Ulusoy. Multiprocessor task scheduling in multistage hybrid flowshops: a genetic algorithm approach. *Journal of the Operational Research Society*, 55(5):504–512, 2004.

[35] F. Sourd and S. Kedad-Sidhoum. The one machine scheduling with earliness and tardiness penalties. *Journal of Scheduling*, 6:533–549, 2003.

[36] F. Sourd and S. Kedad-Sidhoum. An efficient algorithm for the earliness-tardiness scheduling problem. In *Optimization Online*, 2005.

[37] R.H. Storer, S.D. Wu, and R. Vaccari. New search spaces for sequencing problems with application to job shop scheduling. *Management Science*, 38(10):1495–1509, 1992.

[38] E. Taillard. Benchmarks for basic scheduling problems. *European Journal of Operations Research*, 64:278–285, 1993.

[39] P. van Hentenryck, L. Michel, P. Laborie, W. Nuijten, and J. Rogerie. Combinatorial optimization in OPL Studio. In *Proc. EPIA 1999*, pages 1–15, 1999.

[40] M. Vanhoucke, E. Demeulemeester, and W. Herroelen. An exact procedure for the resource-constrained weighted earlinesstardiness project scheduling problem. *Annals of Operations Research*, 102(1-4):179–196, 2001.

[41] M. Vázquez and L. D. Whitley. A comparison of genetic algorithms for the dynamic job shop scheduling problem. In *Proc. GECCO 2000*, 2000.

[42] T. Yamada and R. Nakano. A genetic algorithm applicable to large-scale job-shop problems. In R. Männer and B. Manderick, editors, *Proc. 2nd International Workshop on Parallel Problem Solving from Nature*, pages 281–290, 1992.

Problem	Lower bound	Upper bound
sks524l	77519	77519
sks524r	35160	35160
sks525e	44885	44885
sks527q	18706	18706
sks538l	12489	12489
sks544b	32682	32682
sks545s	26683	26683
sks555a	19781	19781
sks576l	14163	14163
sks587s	18588	18588
sks526u		38051
sks543k		30849
sks545z		27455
sks556u		17090
sks563a		27590
sks564g		31302
sks565a		21073
sks566f		18058
sks568x		11650
sks585x		22765
bky100_21	1215688.67	1241950
bky100_33	825013.58	836286
bky100_65	1630922.89	1659231
bky100_66	2720987.61	2738845
bky100_75	2703753.82	2737821
bky100_129	3333021.95	3407414
bky100_178	3633135.29	3651789
bky100_199	2932066.48	2974934
bky100_251	11094424	11131513
bky100_298	6913028.19	6962141
bky200_3	4863520.4	4971667
bky200_4	5317123.58	5378151
bky200_64	5879030.06	5957791
bky200_74	13485119.43	13716648
bky200_107	5520158.49	5585045
bky200_218	16166389.17	16238878
bky200_219	12390622.3	12526464
bky200_265	21171454.96	21345342
bky200_269	32178497.8	32470446
bky200_286	11757326.22	11870994

Table 5: Single-machine scheduling with earliness/tardiness cost instances

Problem	Upper bound
sch100-1, h=0.8	72019
sch100-3, h=0.2	137463
sch100-3, h=0.4	85363
sch100-3, h=0.6	68537
sch100-4, h=0.4	87730
sch100-4, h=0.6	69231
sch100-5, h=0.2	136761
sch100-8, h=0.4	95361
sch100-9, h=0.6	58771
sch100-10, h=0.2	124446
sch200-1, h=0.6	254268
sch200-5, h=0.2	547953
sch200-5, h=0.8	260455
sch200-7, h=0.2	479651
sch200-7, h=0.8	247555
sch200-8, h=0.8	225572
sch200-9, h=0.4	331107
sch200-9, h=0.8	255029
sch200-10, h=0.4	332808
sch200-10, h=0.6	269236

Table 6: Single-machine scheduling with common due-date

Problem	Lower bound	Upper bound
2010HA1	1210	1565
2010HA3	1468	1653
2010HA4	1500	1660
2010HA7	1295	1600
2010HA10	1254	1603
2010HB2	956	1437
2010HB4	1116	1608
2010HB6	1136	1538
2010HB8	1017	1403
2010HB10	912	1442
5010HA4	3073	3370
5010HA5	2493	3080
5010HA9	2646	3031
5010HB1	1967	2815
5010HB5	2209	3023
H10HA1	5327	5557
H10HA3	5809	6112
H10HB5	3321	4740
H10HB8	3674	5238
H10HB9	3645	4880

Table 7: Multi-stage hybrid flow-shop instances

Problem	Upper bound
inp200.14	69665
inp200.22	186170
inp200.43	6
inp200.44	24
inp200.50	12
inp200.86	0
inp200.119	376528
inp200.133	1683
inp200.168	0
inp200.199	403664
inp500.2	47170
inp500.3	48193
inp500.43	45
inp500.49	26
inp500.72	2421810
inp500.146	819767
inp500.177	0
inp500.180	0
inp500.191	2591064
inp500.198	2607595

Table 8: Single-processor scheduling with total tardiness cost

Problem	Buffer size	Problem size	Upper bound
ta002	0	100	1451
ta003	0	100	1353
ta015	0	200	1678
ta020	0	200	1806
ta037	0	250	3166
ta039	0	250	3045
ta021	0	400	2512
ta026	0	400	2414
ta043	0	500	3658
ta049	0	500	3771
ta001	1	100	1287
ta010	1	100	1134
ta013	1	200	1565
ta018	1	200	1582
ta034	1	250	2888
ta038	1	250	2769
ta021	1	400	2428
ta029	1	400	2310
ta067	1	500	5526
ta068	1	500	5393
ta007	2	100	1251
ta010	2	100	1115
ta017	2	200	1521
ta020	2	200	1642
ta034	2	250	2764
ta038	2	250	2697
ta024	2	400	2242
ta028	2	400	2249
ta047	2	500	3234
ta044	2	500	3129

Table 9: Flow-shop with intermediate buffers