Poznań University of Technology

Extended Performance Results of Algorithm Portfolios for 2D Strip Packing

Kamil Piechowiak, Maciej Drozdowski, Éric Sanlaville

Research Report RA-4/2021

2021

Institute of Computing Science, Piotrowo 2, 60-965 Poznań, Poland

Contents

Algorithm Portfolios for 2D Strip Packing, Extended Results

Kamil Piechowiak*, Maciej Drozdowski*, Éric Sanlaville‡,

*Institute of Computing Science, Poznań University of Technology, Poland
‡LITIS, Normandy University, UNIHAVRE, Le Havre, France

Abstract

In this report extended results are collected to complement paper "Framework of Algorithm Portfolios for Strip Packing Problem".

Keywords: Heuristics; quality-runtime trade-off; 2D packing; algorithm selection problem

1 Introduction

The 2D packing problem considered here consists in placing rectangles on a strip of the given width for minimum strip length. While solving combinatorial optimization problems longer runtimes increase chances of obtaining higher quality solutions. This means that solution quality vs runtime trade-off exists in solving problems such as rectangle strip packing. Given some limited runtime a method is needed to provide the best solution possible. Usually a single algorithm outperforming all other methods under all possible conditions does not exist. Therefore, algorithm portfolios can reliably provide high quality solutions in the limited runtime. We propose a method choosing algorithm portfolios on the basis of the algorithm performance on a set of training instances. The portfolios cover the instances with the best solutions which could be obtained in the given runtime, subject to the minimum overall computational cost of the selected algorithms. The portfolios were evaluated on a set of test instances. We demonstrate that our method is capable of porting solution quality from the training datasets to the testing datasets. In other words, our algorithm selection method can learn from the training instances. Performance of our portfolio selection method is compared against some other more straightforward approaches to the portfolio selection.

The goal of this report is to provide an extended set of results to complement paper [10] *"Framework of Algorithm Portfolios for Strip Packing*

Table 1: Literature dataset summary

subset name	AH	babu	beng	bwmv	C	cgcut	CX
size n	1000	50	20-200	20-100	16-197	16-62	50-15000
No. of inst.	360	1	10	500	21	3	7
Reference	[13]	[13]	[13]	[13]	[13]	[13]	[13]
subset name	gcut	IY	nB	NT	ngcut	Nice	nice36
size n	10-50	$2^4 - 2^{10}$	16-3152	17-197	7-22	25-1000	25-5000
No. of inst.	13	170	13	35	12	6	36
Reference	[13]	[4]	[2]	[13]	[13]	[13]	[13]
subset name	Path	path36	SpPk	T	TC		
size n	25-1000	25-5000	4-1011	17-199	25-188		
No. of inst.	6	36	32	35	10		
Reference	[13]	[13]	[8]	[13]	[9]		

Problem" by the same authors. The current report comprises different results than [11] because a new algorithms space including skyline packing is used [2, 12]. Further organization of the report is the following. In the next section the test datasets are described. In Section 4 example portfolios and their changes in time are shown. Section 5 presents results of performance evaluation of the portfolios constructed on one dataset and validated on a different dataset. In Section 6 our portfolios are compared against the portfolios selected by a simple ad hoc approach which remain the same for all runtime limits.

2 Test Datasets

The algorithms were evaluated on two datasets: the instances generated for this publication and the instances from the earlier literature.

The first set consists of 2000 randomly generated instances divided into dataset1 of 1000 *training instances* and dataset2 of 1000 *validation instances*. Both datasets are further divided into 10 subsets of sizes $n \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$. For each size n 100 instances were generated. Let (x, y, p) denote that a rectangle can be generated with probability p and its sizes are drawn from discrete uniform distribution in range $[1, x] \times [1, y]$. For each instance size n the following mix was generated: 70 instances with rectangles in $(W, W, 1)$, 5 instances in $(W/2, W/2, 1)$, 5 instances in $(W/2, W/2, 0.75)$, $(W, W, 0.25)$, 5 instances in $(W/10, 2W, 1)$, 5 instances in $(W, W, 0.5)$, $(W/10, 2W, 0.5)$, 10 instances in $(W, W/5, 0.33)$, $(W/5, W, 0.34)$, $(W, W, 0.33)$. These test instances are available from [11].

The second dataset is a collection of 1306 instances referred to in earlier publications and will be referred to as literature dataset. The names, sizes and origin of the literature instances are summarized in Tab. 1. For example, instances SpPk represent a practical problem of CSS-Sprite construction [8] and instances TC represent tag cloud construction problem [9].

3 Performance with various runtime limits

Results of algorithm performance under various runtime limits are collected in Tab.2. Five topmost positions in the rankings of the median of the relative distances from the best solutions found, number of wins, unique wins and five worst results in the number of solved instances (in 1000) are shown. Algorithm performance is presented in relation to the best solution found for the considered instance in the given runtime limit. Let us note that minimum time to solve all instances in dataset1 by any algorithm is $\approx$ 3.9ms, so in Tab.2 the smallest shown runtime limit is 4ms to have performance data for all instances. Conversely, the time required to solve each instance with each algorithm is $\approx$0.974s so the medians of the relative distances from the best solutions found are shown only for larger times. Consequently, also the number of unsolved instances is shown up to 0.1s runtime limit because above 0.974s all algorithms solved all instances. Note that SG, as a portfolio of greedy algorithms, cannot solve any instance uniquely. SG, skB, skISH, and other skyline-, ISH-packing algorithms dominate in the first three rankings. Still, these methods do not win in all cases. For example, there are shelf-packing-based shSA, shNFDH, shBFDHr that win uniquely for short runtimes. Furthermore, algorithms with the best relative distance are not able to solve all instances for short runtimes (cf the fourth ranking). For instance, skB solves 890, 996 instances in 1000 for runtimes 4ms and 10ms, respectively (not shown in Tab.2 these algorithms are on further positions).

Table 2: Initial experiment results for various runtime limits (1000 instances). Algorithms referred to in the text are written in boldface.

Algorithms with the lowest relative distance from the best solution found (medians).										
limit	1st		2nd		3rd		4th		5th	
1s	SG	1	skB	1.000211	skTS1	1.003827	skTS0	1.003908	skISH	1.00409
10s	SG	1.000287	skB	1.000611	skTS1	1.002239	skTS0	1.002531	skISH	1.002757
100s	SG	1.000518	skB	1.000837	skISH	1.001569	skTS1	1.002504	skTS0	1.002657
1000s	SG	1.000823	skISH	1.001054	skB	1.001231	skTS1	1.002653	skTS0	1.002832
Algorithms with the biggest number of wins										
limit	1st		2nd		3rd		4th		5th	
4ms	skB	546	skHCg	250	skISH	246	skTS0	171	skTS1	170
10ms	skB	612	skHCg	280	skISH	261	SG	199	skTS1	181
100ms	skB	533	SG	395	skISH	304	skTS1	279	skTS0	269
1s	SG	509	skB	497	skISH	314	skTS0	307	skTS1	299
10s	SG	489	skB	478	skISH	335	skTS0	320	skTS1	317
100s	SG	470	skB	459	skISH	371	skTS0	321	skTS1	305
1000s	SG	447	skB	436	skISH	406	blSA	329	skTS1	298
Algorithms with the biggest number of unique wins										
limit	1st		2nd		3rd		4th		5th	
4ms	skB	425	skISH	76	skHCg	75	**shSA**	41	**shNFDH**	31
10ms	skB	491	skHCg	94	skISH	75	**shSA**	45	**shBFDHr**	31
100ms	skB	415	skISH	85	**shSA**	53	skHCg	52	skTS1	51
1s	skB	380	skISH	75	skTS1	67	skTS0	66	**shSA**	31
10s	skB	361	skISH	78	skTS0	67	skTS1	64	**shSA**	29
100s	skB	341	skISH	101	skTS0	60	skTS1	44	blSA	38
1000s	skB	318	skISH	119	blSA	62	skHCs	58	skTS1	29
Algorithms with the smallest number of solved instances										
limit	1st		2nd		3rd		4th		5th	
4ms	SG	394	shHC	607	shSA	614	skISH	741	skHCs	748
10ms	SG	498	shHC	690	shSA	703	skISH	787	skHCs	830
100ms	SG	800	shHC	820	shSA	835	–	–	–	–

4 Example Portfolios Evolution in Time

Portfolios changes with runtime limits are presented in Figs 1, 2, 3. The portfolios were developed on training datasets of some given size (see figure captions) from dataset1. Along horizontal axis runtime limit T in nanoseconds is shown. The left sides of the figures present algorithm membership in the portfolio. Each colored strip is depicting interval(s) of certain algorithm membership in the portfolio. The right sides of the figures show the number of algorithms in the portfolios and the computational cost. The computational cost is the total execution time of the algorithms in the portfolio. In Figs 1 - 3 computational cost is shown in T units (i.e. $Cost/T$ is shown along vertical axis on the right figures). The number of algorithms and the computational costs are not equal if some algorithm stops before runtime limit T. This is the case of greedy deterministic algorithms, and it can be observed in Figs 2d and 3 (on the right). Observe how the relative positions of shelf-, bottom-left-, skyline- (sh, sh, bl) heuristics change with problem sizes (n).

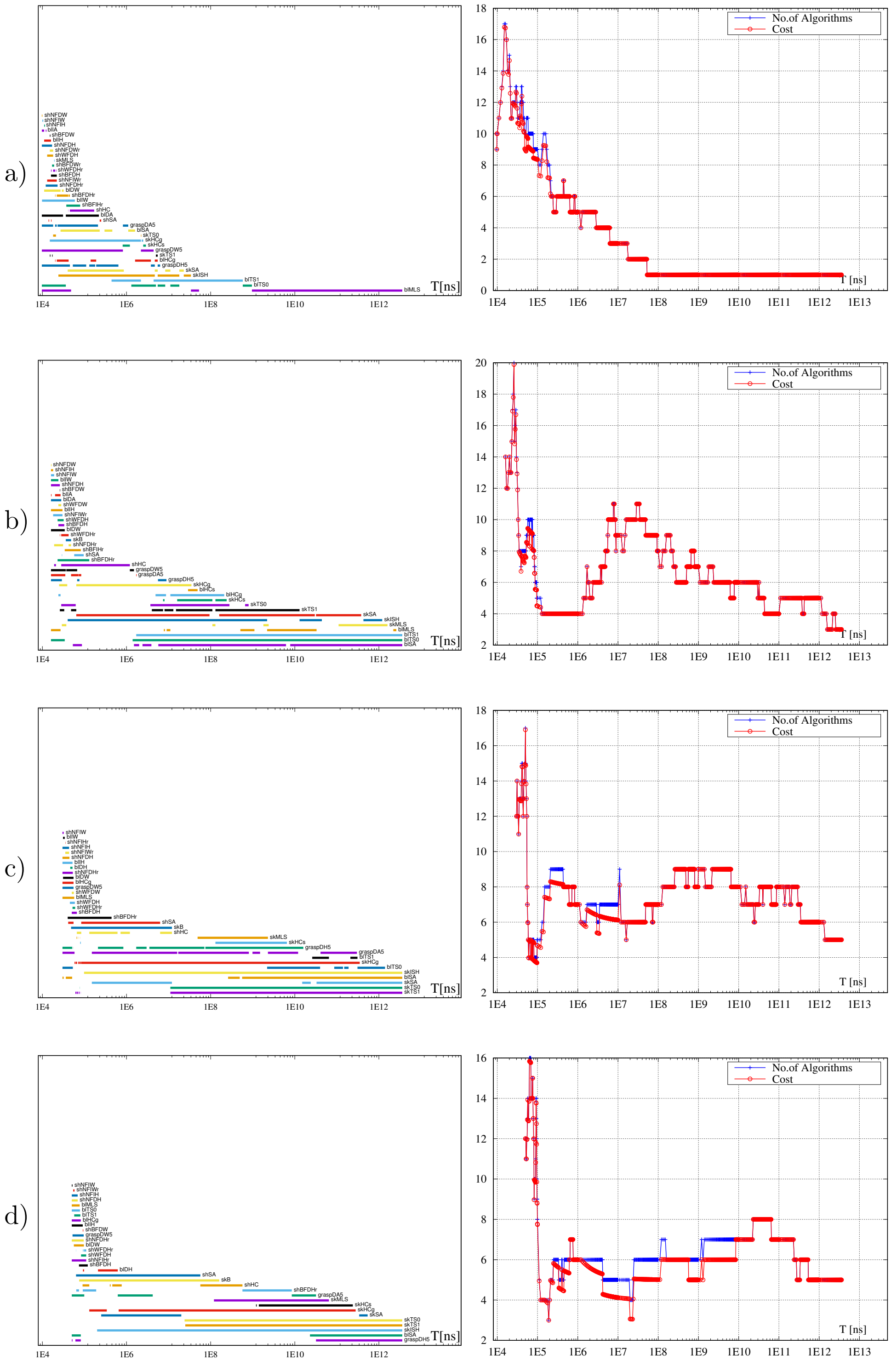

Figure 1: Examples of algorithm portfolios and their costs for instances of various sizes (left, right): a) $n = 10$, b) $n = 20$, c) $n = 50$, d) $n = 100$. Portfolios in time $PF(T)$ are shown on the left side, portfolio costs and number of algorithms on the right side.

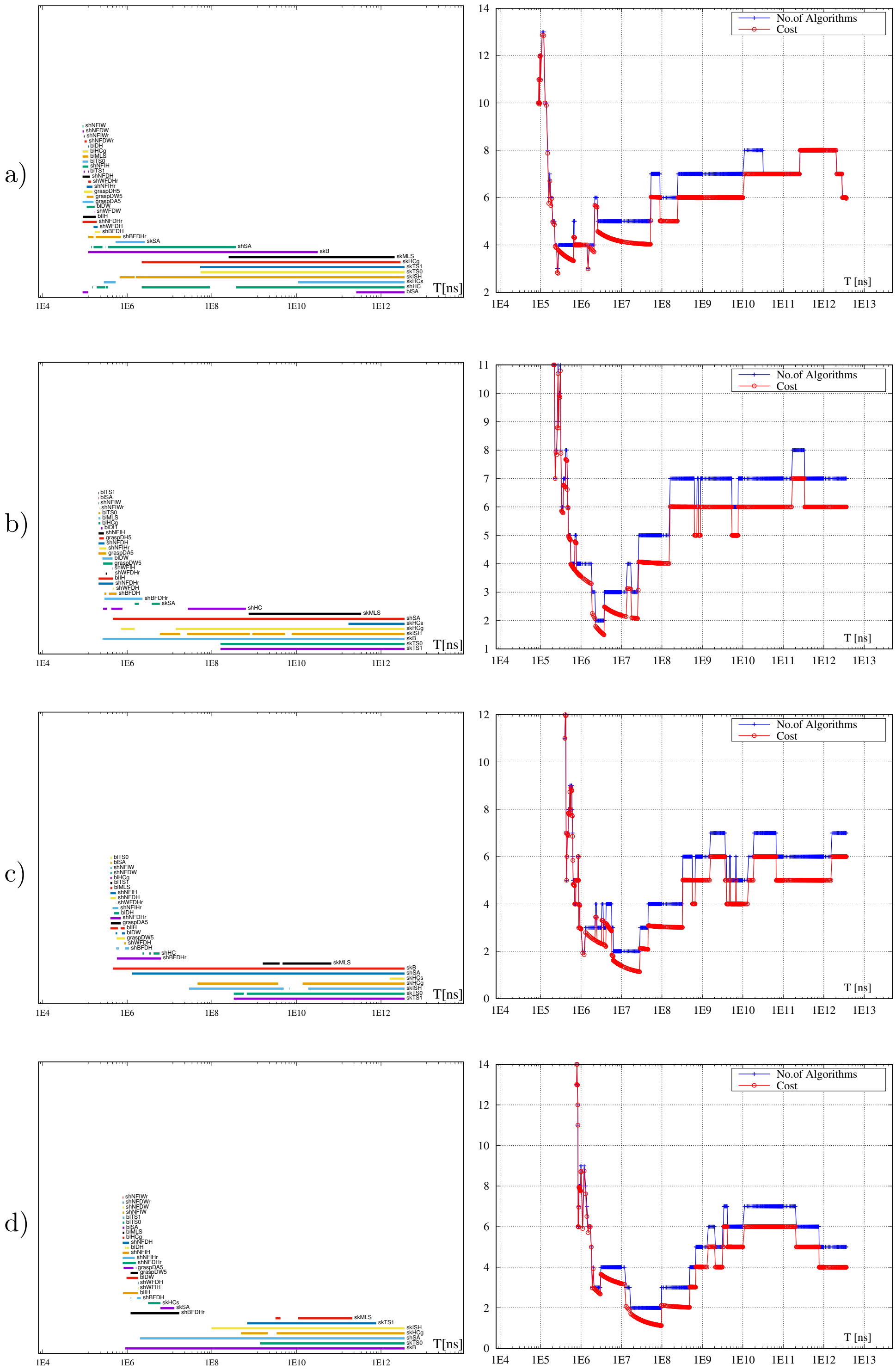

Figure 2: Examples of algorithm portfolios and their costs for instances of various sizes (left, right): a) $n = 200$, b) $n = 500$, c) $n = 1000$, d) $n = 2000$. Portfolios in time $PF(T)$ are shown on the left side, portfolio costs and number of algorithms on the right side.

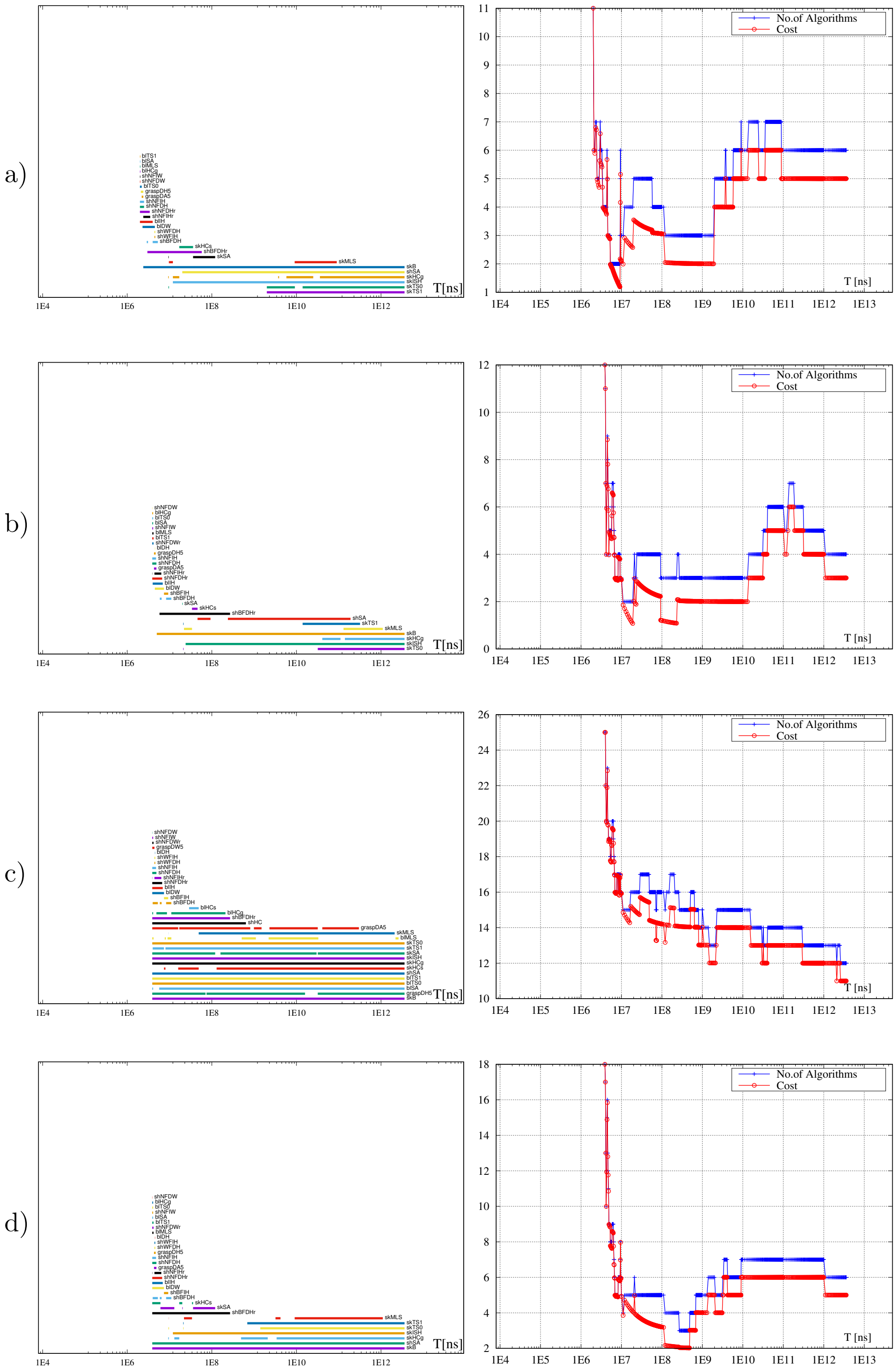

Figure 3: Examples of algorithm portfolios and their costs for instances of various sizes (left, right): a) $n = 5000$, b) $n = 10000$, c) $n = 10, \ldots, 10000$ (whole dataset1), d) $n \in \{2000, 5000, 10000\}$. Portfolios in time $PF(T)$ are shown on the left side, portfolio costs and number of algorithms on the right side.

5 Cross-validation

In this section we present results of performance evaluation of the algorithm portfolios. The outcomes are shown in Figs 4 - 19. The portfolios were developed for a set of training instances of some given size (see figure captions) from dataset1. The sizes of training and validation instances were varied to analyze the impact of the differences in training and validation instance sizes. The evaluation consisted in solving validation instances from dataset2 and real instances both by the portfolio algorithms and by all our algorithms. Quality of the solutions was measured as wins, and relative distance from the best solution found. A win means that one of the portfolio algorithms provided the best solution found. These quality scores are shown for a population of instances.

5.1 Wins

The number of wins in Fig.4 through Fig.6 is limited to the cardinality of the validation dataset. In particular, the maximum possible number of wins is 100 in Fig.4a-e and 1306 in Fig.4f. The line for whole dataset2 in Fig.4e is an exception and at most 1000 wins are possible. Note that wins here are from comparison of the portfolio with the best algorithm result in our algorithm space. These are not comparisons with the best results known in the literature for the literature dataset.

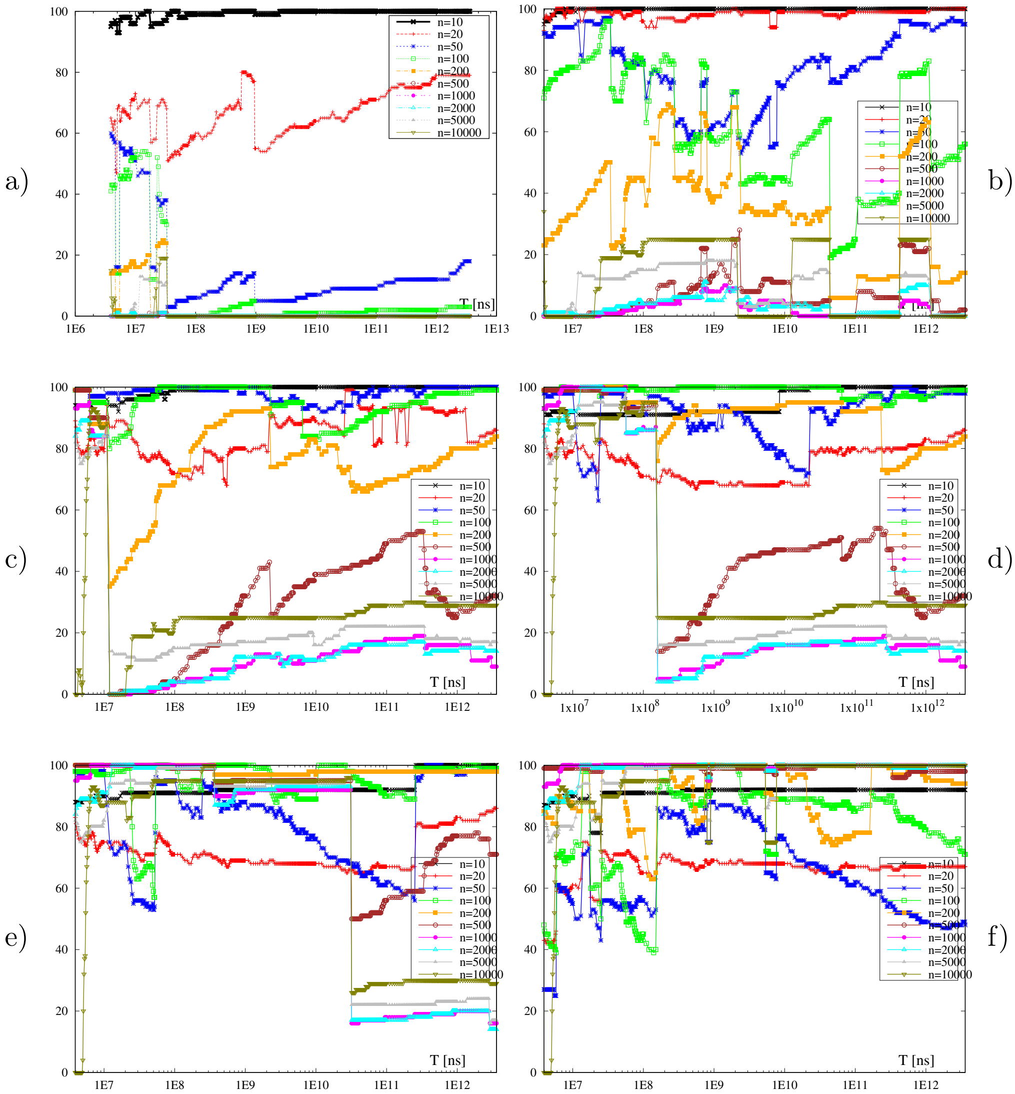

Figure 4: Cross-validation. Number of wins. Training: instances of a fixed size n from dataset1: a) $n = 10$, b) $n = 20$, c) $n = 50$, d) $n = 100$. e) $n = 200$. f) $n = 500$. Validation: instances of a fixed size n from dataset2. Each line in the pictures is for one size of validating instances. 100 is the maximum number of possible wins.

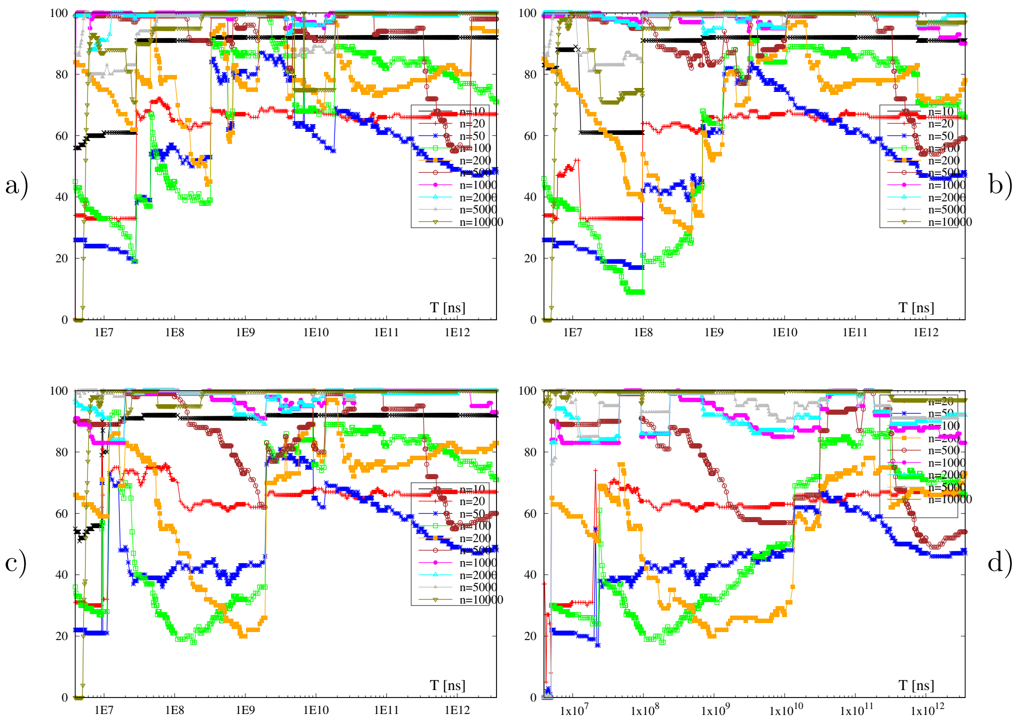

Figure 5: Cross-validation. Number of wins. Training: instances of a fixed size n from dataset1: a) $n = 1000$, b) $n = 2000$, c) $n = 5000$, d) $n = 10000$. Validation: instances of a fixed size n from dataset2. Each line in the pictures is for one size of validating instances. 100 is the maximum number of possible wins.

a) b) c)

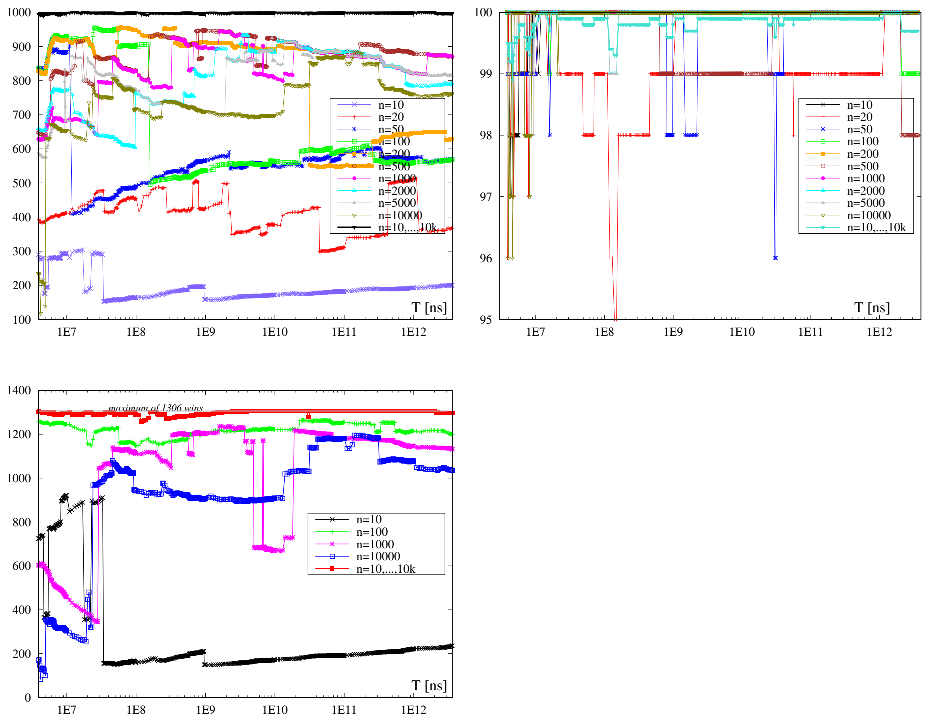

Figure 6: Cross-validation. Number of wins. a) Training: the subset of dataset1 instances of a given size, validation: whole datasets2 the maximum number of wins is 1000. b) Training: whole dataset1 ($n = 10, \dots, 10000$), validation: subset of dataset2 instances of a given size. The maximum number of possible wins is 100 except for the line denoted $n = 10, \dots, 10000$ where whole datasets2 was used for validation and the maximum number of wins is 1000 (in this case values are divided by 10 to be shown on the same Y-axis) c) Training: instances of the given sizes n in dataset1, validation: literature instances. The maximum number of wins is 1306. .

5.2 Median Relative distance from the best solution found

Medians of the relative distance from the best solution found are shown in Fig.7 through Fig.9. The biggest differences are shown in Fig.10 through Fig.12.

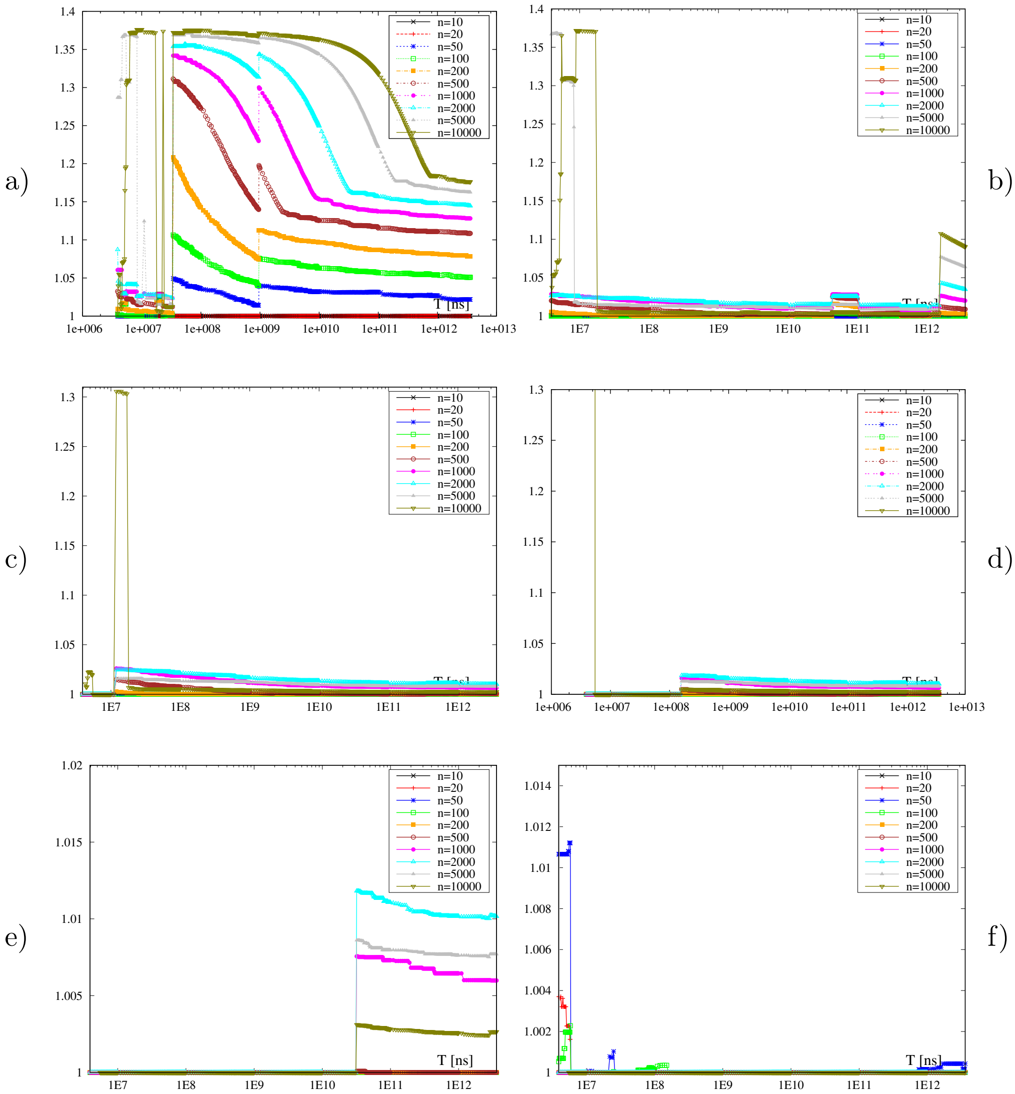

Figure 7: Cross-validation. Median of the relative distance from the best solution found. Training: instances of a fixed size n from dataset1: a) $n = 10$, b) $n = 20$, c) $n = 50$, d) $n = 100$. e) $n = 200$. f) $n = 500$. Validation: instances of a fixed size n from dataset2. Each line in the pictures is for one size of validating instances.

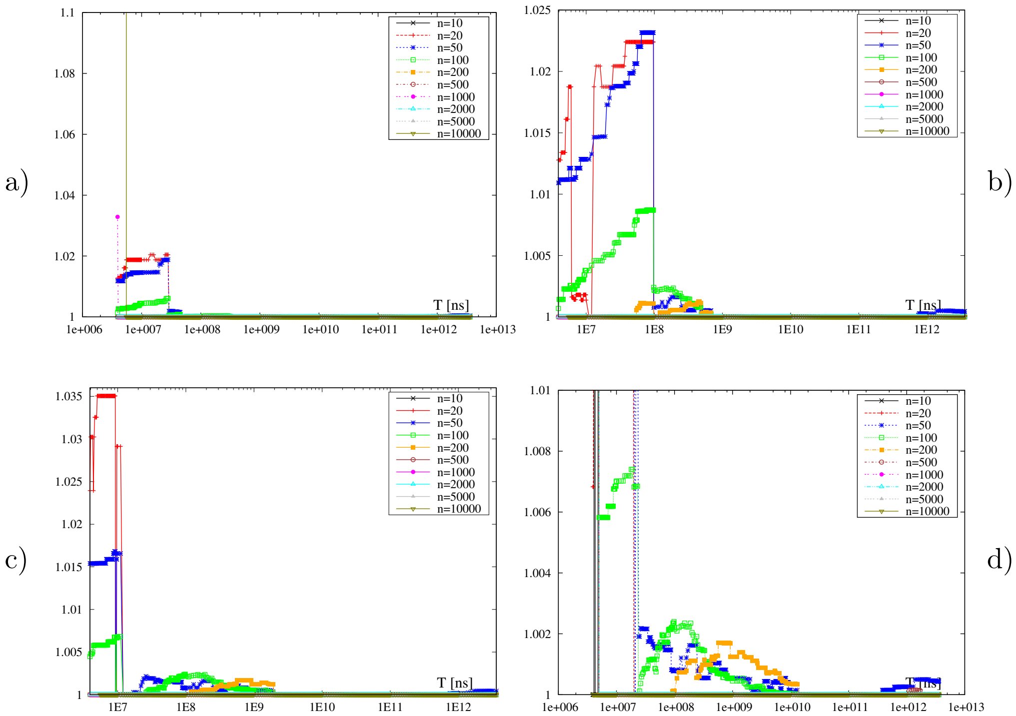

Figure 8: Cross-validation. Median of the relative distance from the best solution found. Training: instances of a fixed size n from dataset1: a) $n = 1000$, b) $n = 2000$, c) $n = 5000$, d) $n = 10000$. Validation: instances of a fixed size n from dataset2. Each line in the pictures is for one size of validating instances.

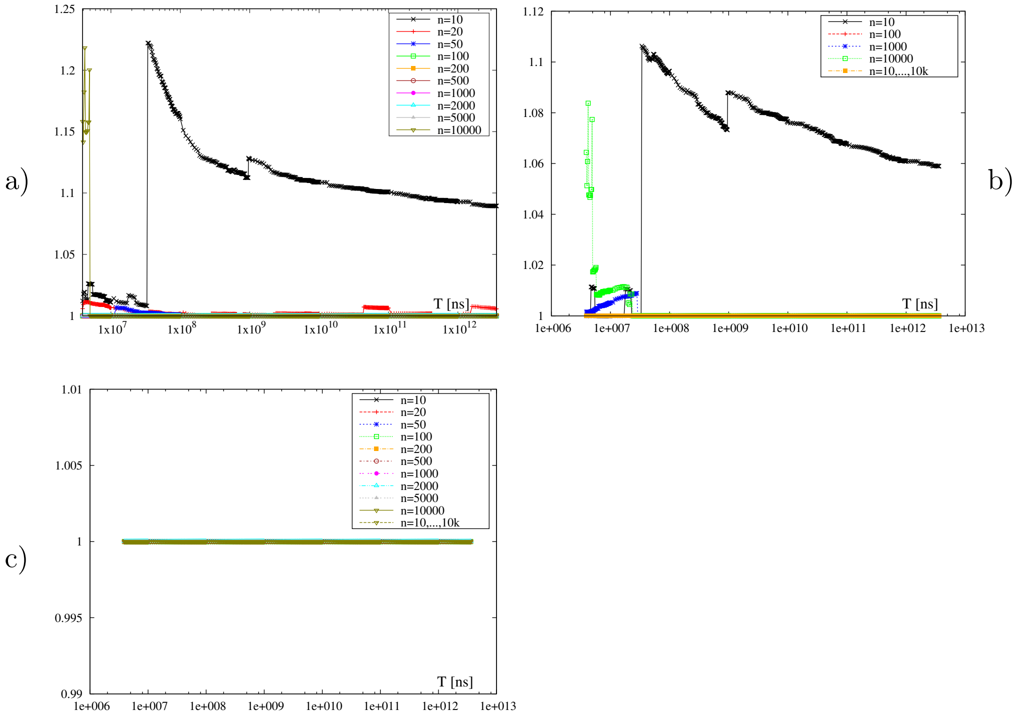

Figure 9: Cross-validation. Median of the relative distance from the best solution found. a) Training: the subset of dataset1 instances of a given size, validation: whole datasets2. b) Training: instances of the given sizes n in dataset1, validation: literature instances. c) Training: whole dataset1 ($n = 10, \ldots, 10000$), validation: subset of dataset2 instances of a given size.

5.3 Maximum Relative distance from the best solution found

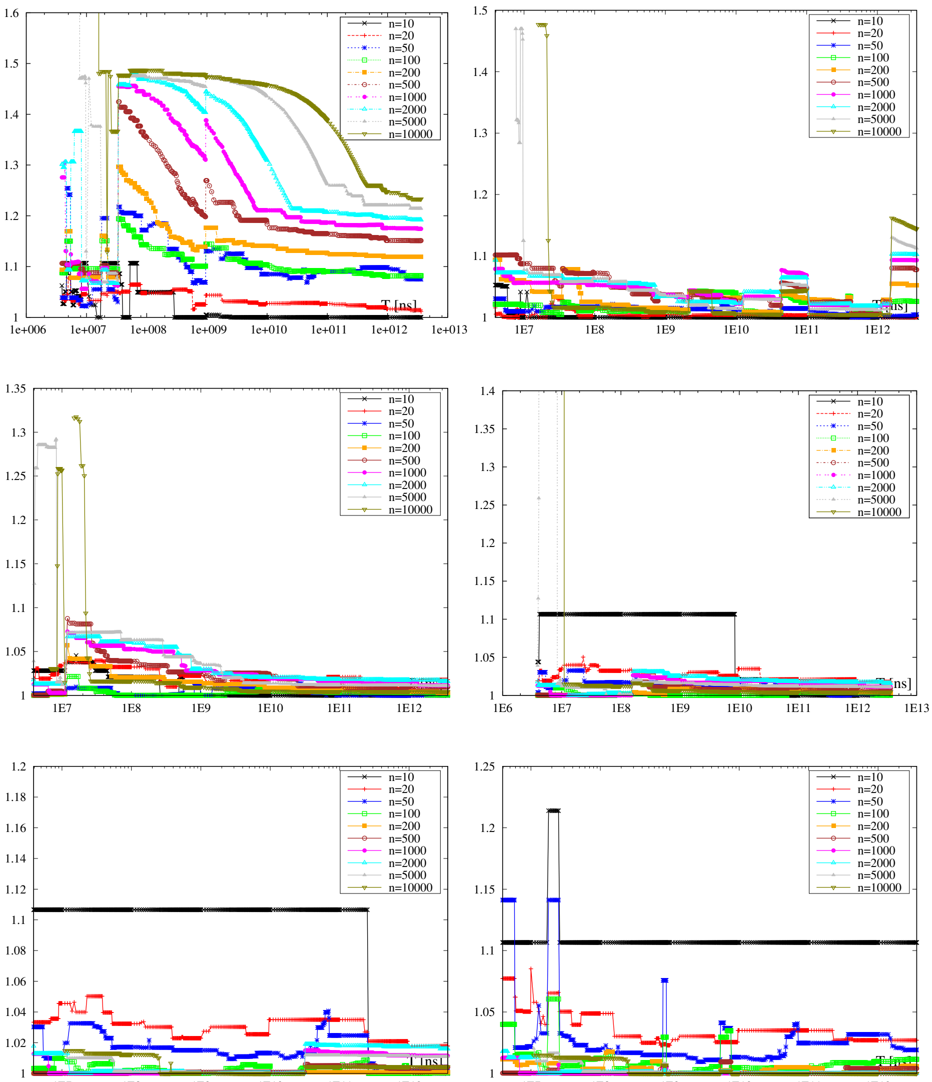

Figure 10: Cross-validation. The worst-case relative distance from the best solution found. Training: instances of a fixed size n from dataset1: a) $n = 10$, b) $n = 20$, c) $n = 50$, d) $n = 100$. e) $n = 200$. f) $n = 500$. Validation: instances of a fixed size n from dataset2. Each line in the pictures is for one size of validating instances.

a) b) c) d)

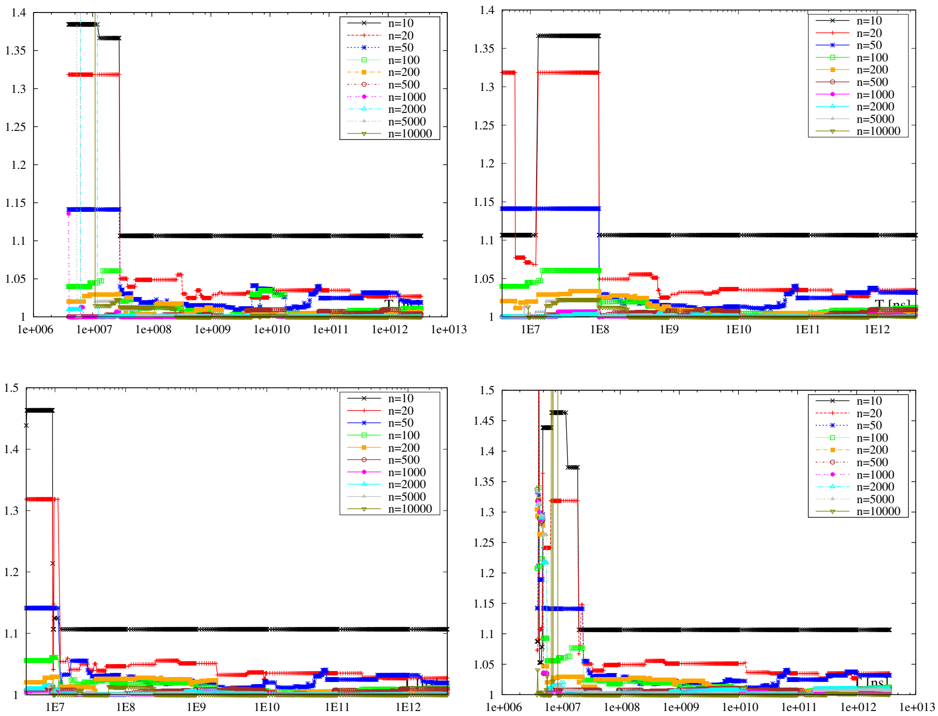

Figure 11: Cross-validation. The worst-case relative distance from the best solution found. Training: instances of a fixed size n from dataset1: a) $n = 1000$, b) $n = 2000$, c) $n = 5000$, d) $n = 10000$. Validation: instances of a fixed size n from dataset2. Each line in the pictures is for one size of validating instances.

a) b) c)

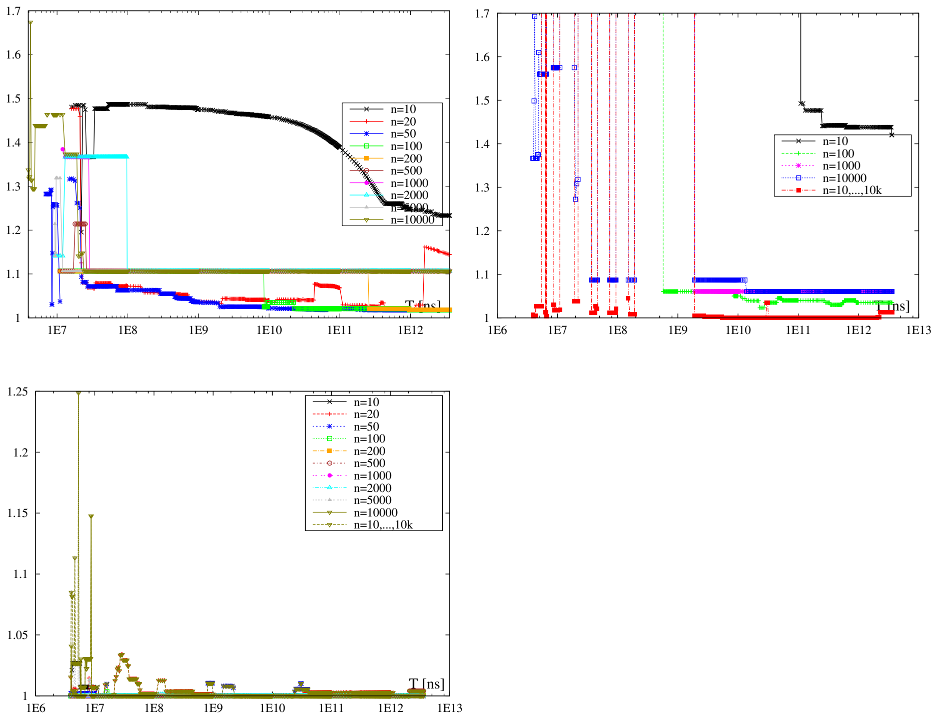

Figure 12: Cross-validation. The worst-case relative distance from the best solution found. a) Training: the subset of dataset1 instances of a given size, validation: whole datasets2. b) Training: instances of the given sizes n in dataset1, validation: literature instances. c) Training: whole dataset1 ($n = 10, \ldots, 10000$), validation: subset of dataset2 instances of a given size.

6 Comparison with Fixed Portfolios

Potentially, there are other ways of constructing algorithm portfolios. In this section results of comparison with 5 alternative portfolios are presented:

1. super greedy which is a portfolio itself (SG),
2. all versions of skyline metaheuristics (skMH),
3. all versions of bottom-left placement metaheuristics (blMH),
4. all versions of shelf placement metaheuristics (shMH),
5. all simulated annealing metaheuristics (SA).

Since these portfolios do not change with runtime limit T we will call them *fixed*. Performance of the individual fixed portfolios with respect to the number of wins, median and the worst-case relative distance from the best solution found are shown in Figs 13 - 17. Computational cost is shown for SG only because only in the SG portfolio greedy algorithms stopping before runtime limit T are present. Consequently, only for SG can the computational cost expressed in T units be different than the number of the portfolio algorithms. All other above portfolios comprise only metaheuristics which run to the runtime limit T. As a result, for all these portfolios computational cost in T units is constant and equal to the number of comprised algorithms. In Fig.18 performance of the fixed portfolios is compared with our portfolios evolving in runtime T.

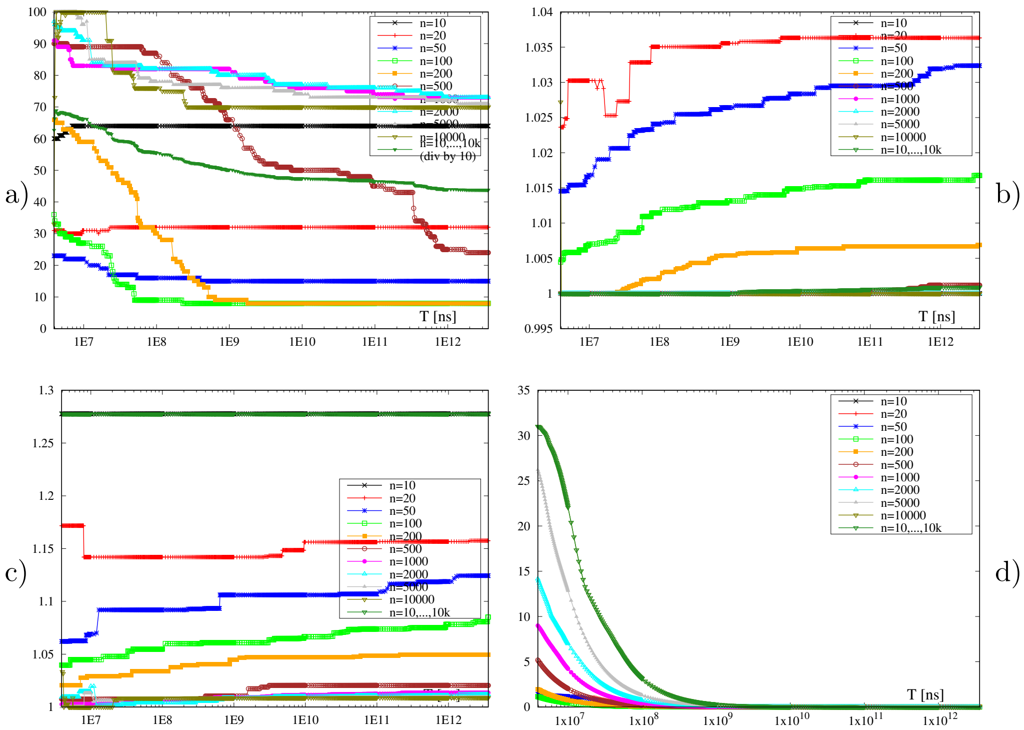

Figure 13: Fixed portfolio Super Greedy (SG). a) Number of wins, b) median solution quality, c) worst solution quality, d) computational cost in $Cost/T$ units.

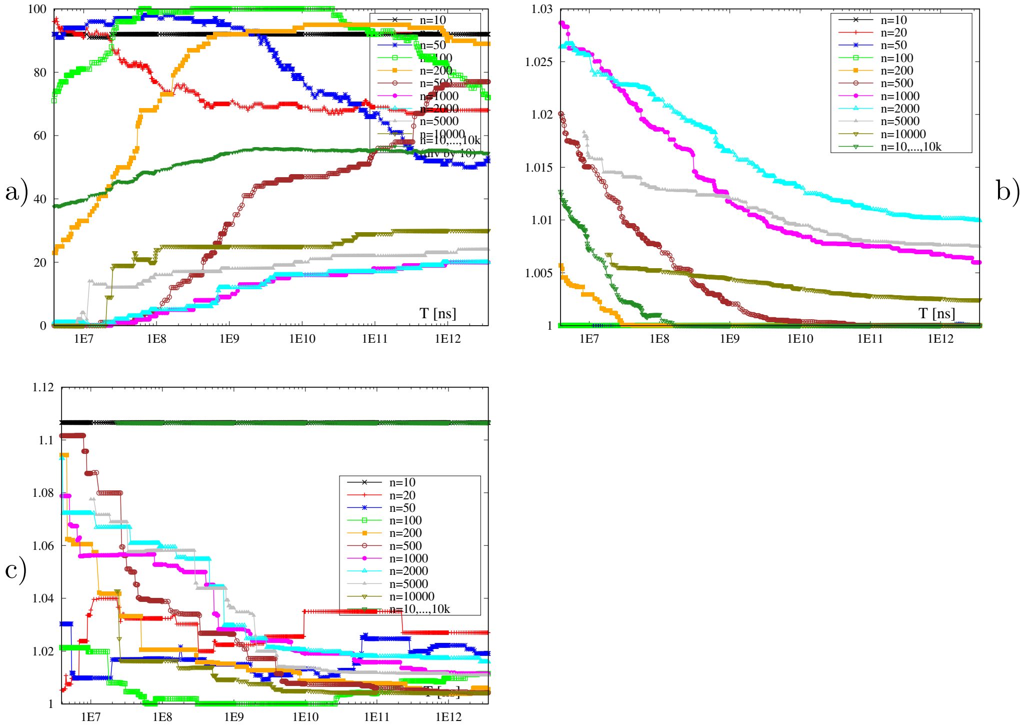

Figure 14: Fixed portfolio of all skyline metaheuristics. a) Number of wins, b) median solution quality, c) worst solution quality.

Note that for instances of sizes $n = 5000, 10000$ and dataset2 which also has this size instances, the relative distance from the best result found is undefined for $T \leq 2.3$E7ns because no solution is provided by any skyline metaheuristic. The cost of this portfolio in $Cost/T$ units is equal to 7 (the number of algorithms in the portfolio) because all these metaheuristics are suspended on runtime limit.

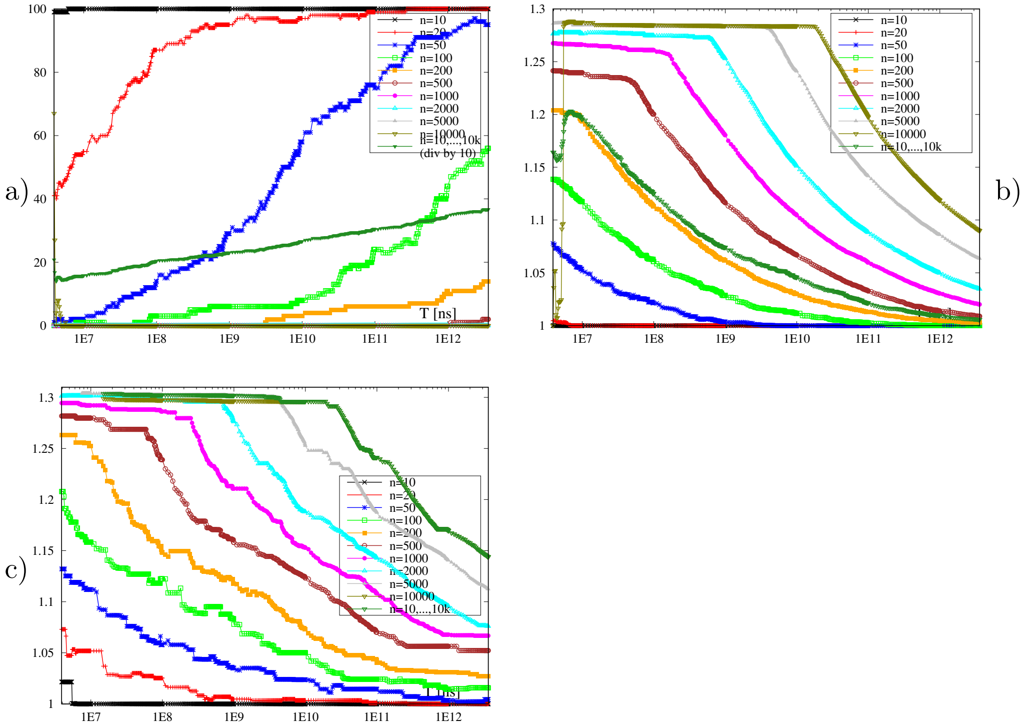

Figure 15: Fixed portfolio of all bottom-left metaheuristics. a) Number of wins, b) median solution quality, c) worst solution quality.

Note that for some instances of sizes $n = 5000, 10000$ and dataset2 which also has these size instances, the relative distance from the best result found is undefined for $T \leq 1.4$E7ns because no solution is provided by any bottom-left metaheuristic. The cost of this portfolio in $Cost/T$ units is equal to 9 (the number of algorithms in the portfolio) because all these metaheuristics are suspended on runtime limit.

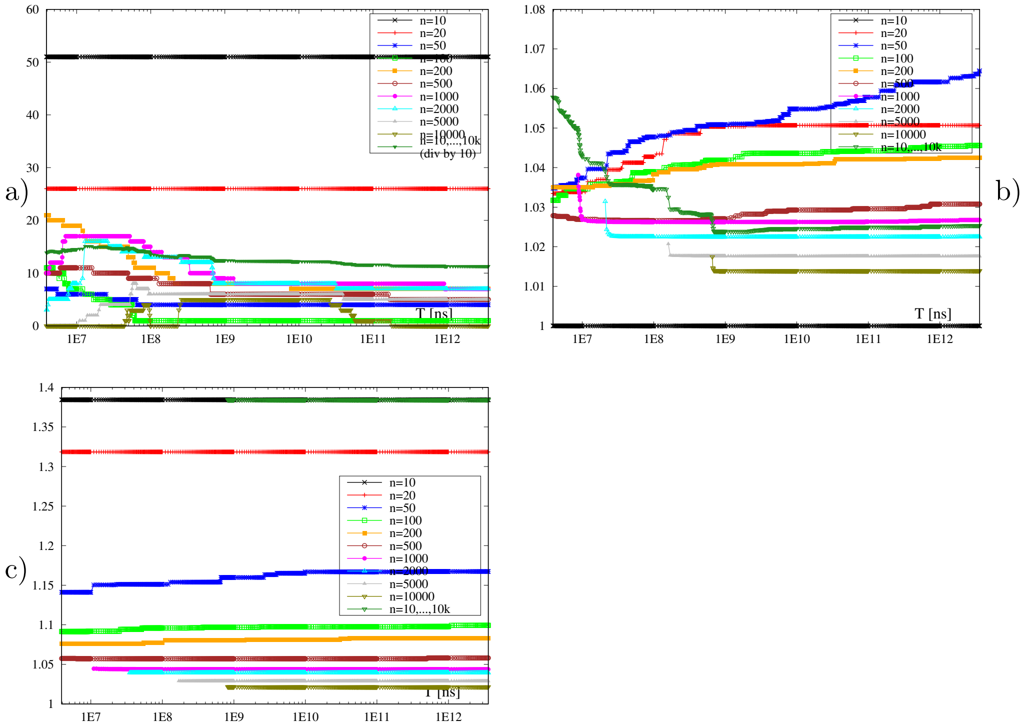

Figure 16: Fixed portfolio of all shelf-placement metaheuristics. a) Number of wins, b) median solution quality, c) worst solution quality.

Note that for some instances of sizes $n \geq 1000$ and dataset2 which also has these size instances, the relative distance from the best result found is undefined for $T \leq 7.9$E8ns because no solution is provided by any bottom-left metaheuristic. The cost of this portfolio in $Cost/T$ units is equal to 2 (the number of algorithms in the portfolio) because all these metaheuristics are suspended on runtime limit.

a) b) c)

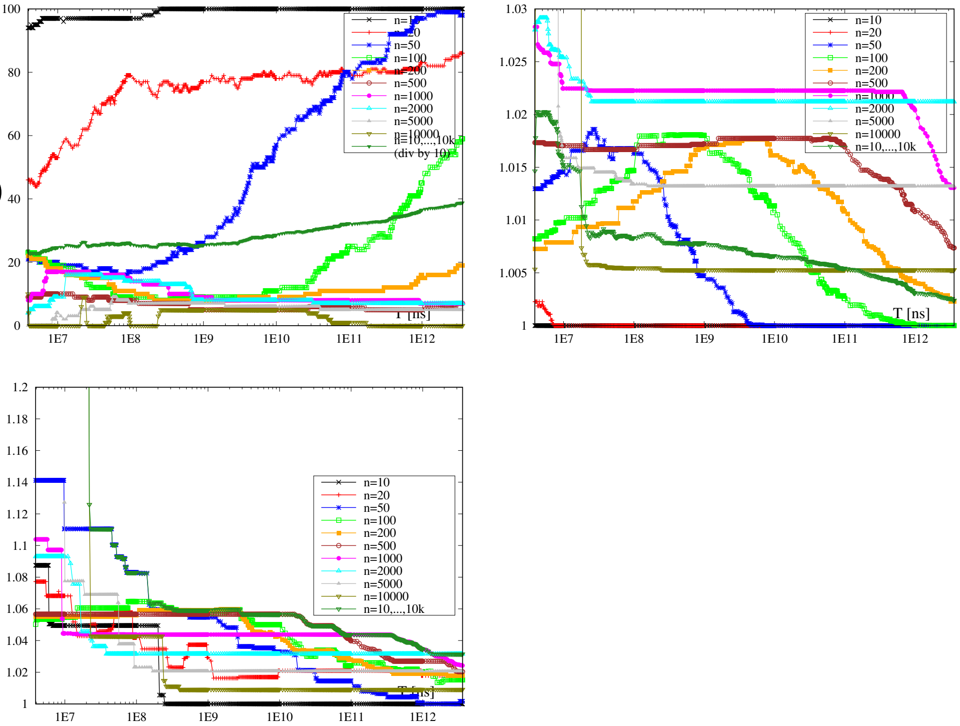

Figure 17: Fixed portfolio of all simulated annealing metaheuristics. a) Number of wins, b) median solution quality, c) worst solution quality.

Note that for some instances of sizes $n \geq 5000$ and dataset2 which also has these size instances, the relative distance from the best result found is undefined for $T \leq 9.8\text{E}8\text{ns}$ because no solution is provided by any bottom-left metaheuristic. The median relative distance for $n = 5000, 10000$ in range of $T \leq 1.7\text{E}7\text{ns}$ is ≈ 1.38 but the figure range of Y-axis values is reduced to [1,1.03] for better visibility. The cost of this portfolio in $Cost/T$ units is equal to 3 (the number of algorithms in the portfolio) because all these metaheuristics are suspended on runtime limit.

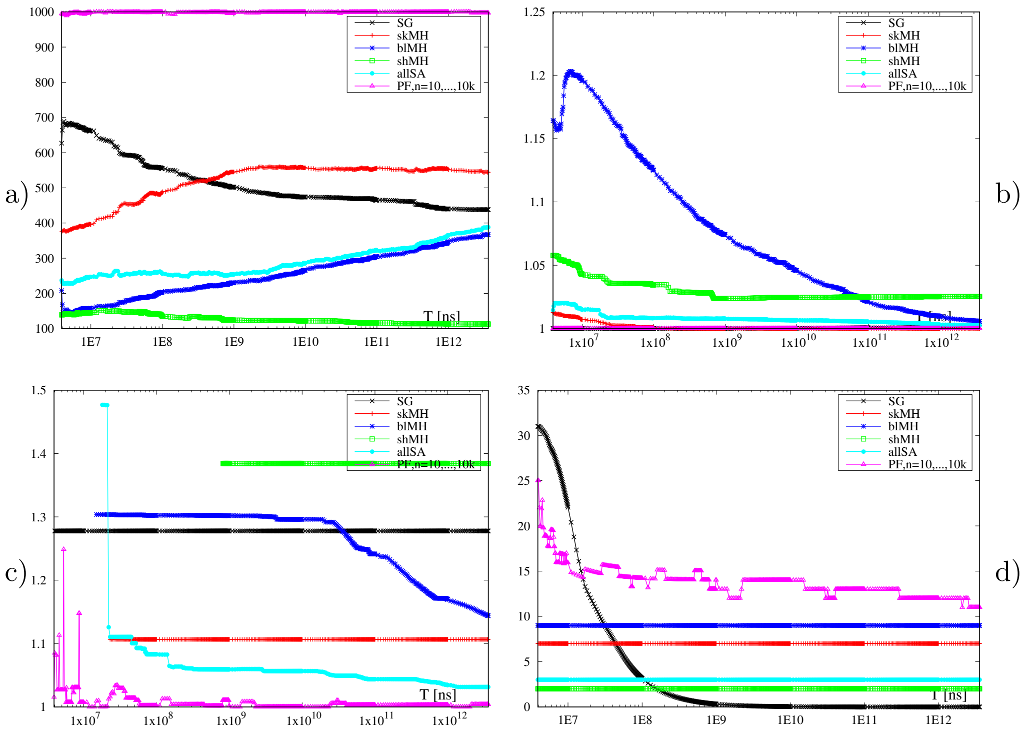

Figure 18: Our portfolio against fixed portfolios on whole dataset2. a) Number of wins, b) median solution quality, c) worst solution quality, d) computational cost in $Cost/T$ units. SG - super-greedy, skMH - skyline placement metaheuristics, blMH - bottom-left placement metaheuristics, shMH - shelf-placement metaheuristics, allSA - all simulated annealing metaheuristics.

Note that for some instances of large size, the relative distance from the best result found is undefined because no solution is provided by the considered portfolio metaheuristic. In particular this applies to bottom-left metaheuristics (blMH) and $T \leq 1.4$E7ns, all simulated annealing metaheuristics (allSA) and $T \leq 1.7$E7ns, skyline placement metaheuristics (skMH) and $T \leq 2.2$E7ns, shelf-placement metaheuristics (shMH) and $T \leq 8.1$E8ns.

7 Detailed Results for Literature Dataset Instances

Each line provides results in the format: our instance number, literature name, our result (height of the strip).

2166 100.txt 612
2167 1000.txt 600
2168 10000.txt 600
2169 15000.txt 600
2170 50.txt 600
2171 500.txt 600
2172 5000.txt 600
2173 AH1.txt 35378
2174 AH10.txt 38235
2175 AH100.txt 1265
2176 AH101.txt 2118
2177 AH102.txt 2048
2178 AH103.txt 3573
2179 AH104.txt 2374
2180 AH105.txt 1868
2181 AH106.txt 1366
2182 AH107.txt 2824
2183 AH108.txt 1734
2184 AH109.txt 2925
2185 AH11.txt 42723
2186 AH110.txt 2179
2187 AH111.txt 2785
2188 AH112.txt 1995
2189 AH113.txt 2272
2190 AH114.txt 2007
2191 AH115.txt 2678
2192 AH116.txt 1962
2193 AH117.txt 3135
2194 AH118.txt 2375
2195 AH119.txt 2563
2196 AH12.txt 29966
2197 AH120.txt 2391
2198 AH121.txt 41884
2199 AH122.txt 39056
2200 AH123.txt 39629
2201 AH124.txt 42083
2202 AH125.txt 40494
2203 AH126.txt 39888
2204 AH127.txt 40139
2205 AH128.txt 41626
2206 AH129.txt 39663
2207 AH13.txt 38551
2208 AH130.txt 40689
2209 AH131.txt 39488
2210 AH132.txt 39336
2211 AH133.txt 39773
2212 AH134.txt 41166
2213 AH135.txt 38304
2214 AH136.txt 39503
2215 AH137.txt 38981
2216 AH138.txt 39126
2217 AH139.txt 39270
2218 AH14.txt 47014
2219 AH140.txt 38484
2220 AH141.txt 40886
2221 AH142.txt 38968
2222 AH143.txt 40534
2223 AH144.txt 41517
2224 AH145.txt 41277
2225 AH146.txt 41743
2226 AH147.txt 42362
2227 AH148.txt 38000
2228 AH149.txt 39302
2229 AH15.txt 35176
2230 AH150.txt 40910
2231 AH151.txt 41426
2232 AH152.txt 42337
2233 AH153.txt 40982
2234 AH154.txt 36688
2235 AH155.txt 42747
2236 AH156.txt 37336
2237 AH157.txt 41724
2238 AH158.txt 40281
2239 AH159.txt 43910
2240 AH16.txt 38309
2241 AH160.txt 40473
2242 AH161.txt 40814
2243 AH162.txt 35444
2244 AH163.txt 42720

2245 AH164.txt 38733
2246 AH165.txt 42484
2247 AH166.txt 42411
2248 AH167.txt 38639
2249 AH168.txt 37136
2250 AH169.txt 36574
2251 AH17.txt 39990
2252 AH170.txt 36309
2253 AH171.txt 40723
2254 AH172.txt 36050
2255 AH173.txt 46596
2256 AH174.txt 41780
2257 AH175.txt 40587
2258 AH176.txt 42089
2259 AH177.txt 40222
2260 AH178.txt 41721
2261 AH179.txt 43518
2262 AH18.txt 49277
2263 AH180.txt 42899
2264 AH181.txt 2533
2265 AH182.txt 2652
2266 AH183.txt 2384
2267 AH184.txt 2573
2268 AH185.txt 2606
2269 AH186.txt 2572
2270 AH187.txt 2407
2271 AH188.txt 2752
2272 AH189.txt 2555
2273 AH19.txt 42349
2274 AH190.txt 2541
2275 AH191.txt 2534
2276 AH192.txt 2553
2277 AH193.txt 2599
2278 AH194.txt 2616
2279 AH195.txt 2549
2280 AH196.txt 2501
2281 AH197.txt 2376
2282 AH198.txt 2464
2283 AH199.txt 2494
2284 AH2.txt 35626
2285 AH20.txt 34039
2286 AH200.txt 2439
2287 AH201.txt 2486
2288 AH202.txt 2635
2289 AH203.txt 2660
2290 AH204.txt 2458
2291 AH205.txt 2547
2292 AH206.txt 2424
2293 AH207.txt 2417
2294 AH208.txt 2479
2295 AH209.txt 2300
2296 AH21.txt 46198
2297 AH210.txt 2511
2298 AH211.txt 2278
2299 AH212.txt 2322
2300 AH213.txt 2340
2301 AH214.txt 2458
2302 AH215.txt 2316
2303 AH216.txt 2155
2304 AH217.txt 2448
2305 AH218.txt 2164
2306 AH219.txt 2087
2307 AH22.txt 44689
2308 AH220.txt 2487
2309 AH221.txt 2081
2310 AH222.txt 2509
2311 AH223.txt 1970
2312 AH224.txt 2129
2313 AH225.txt 2124
2314 AH226.txt 2527
2315 AH227.txt 2316
2316 AH228.txt 2225
2317 AH229.txt 2470
2318 AH23.txt 41137
2319 AH230.txt 2282
2320 AH231.txt 2614
2321 AH232.txt 2427
2322 AH233.txt 2328
2323 AH234.txt 2447
2324 AH235.txt 2436
2325 AH236.txt 2438
2326 AH237.txt 2365
2327 AH238.txt 2558
2328 AH239.txt 2430
2329 AH24.txt 35892
2330 AH240.txt 1988
2331 AH241.txt 40320

2332 AH242.txt 40008
2333 AH243.txt 39696
2334 AH244.txt 40620
2335 AH245.txt 40961
2336 AH246.txt 40447
2337 AH247.txt 40553
2338 AH248.txt 40580
2339 AH249.txt 41411
2340 AH25.txt 37666
2341 AH250.txt 40495
2342 AH251.txt 40346
2343 AH252.txt 41144
2344 AH253.txt 41231
2345 AH254.txt 40630
2346 AH255.txt 40118
2347 AH256.txt 40654
2348 AH257.txt 40936
2349 AH258.txt 39559
2350 AH259.txt 41003
2351 AH26.txt 40912
2352 AH260.txt 40768
2353 AH261.txt 39403
2354 AH262.txt 40032
2355 AH263.txt 39778
2356 AH264.txt 41249
2357 AH265.txt 40709
2358 AH266.txt 40632
2359 AH267.txt 40526
2360 AH268.txt 40705
2361 AH269.txt 40166
2362 AH27.txt 44488
2363 AH270.txt 41190
2364 AH271.txt 39669
2365 AH272.txt 40012
2366 AH273.txt 39951
2367 AH274.txt 39850
2368 AH275.txt 39827
2369 AH276.txt 40869
2370 AH277.txt 40960
2371 AH278.txt 38768
2372 AH279.txt 38952
2373 AH28.txt 46035
2374 AH280.txt 40924
2375 AH281.txt 39318
2376 AH282.txt 39106
2377 AH283.txt 41560
2378 AH284.txt 41029
2379 AH285.txt 40858
2380 AH286.txt 41162
2381 AH287.txt 41203
2382 AH288.txt 41187
2383 AH289.txt 41144
2384 AH29.txt 36874
2385 AH290.txt 41120
2386 AH291.txt 39724
2387 AH292.txt 39307
2388 AH293.txt 40625
2389 AH294.txt 40739
2390 AH295.txt 39925
2391 AH296.txt 40431
2392 AH297.txt 39859
2393 AH298.txt 41138
2394 AH299.txt 39969
2395 AH3.txt 38398
2396 AH30.txt 36326
2397 AH300.txt 39263
2398 AH301.txt 2513
2399 AH302.txt 2524
2400 AH303.txt 2520
2401 AH304.txt 2534
2402 AH305.txt 2537
2403 AH306.txt 2526
2404 AH307.txt 2486
2405 AH308.txt 2522
2406 AH309.txt 2516
2407 AH31.txt 44110
2408 AH310.txt 2542
2409 AH311.txt 2506
2410 AH312.txt 2525
2411 AH313.txt 2538
2412 AH314.txt 2482
2413 AH315.txt 2501
2414 AH316.txt 2528
2415 AH317.txt 2521
2416 AH318.txt 2506
2417 AH319.txt 2547
2418 AH32.txt 37669

2419 AH320.txt 2548
2420 AH321.txt 2516
2421 AH322.txt 2525
2422 AH323.txt 2514
2423 AH324.txt 2482
2424 AH325.txt 2519
2425 AH326.txt 2504
2426 AH327.txt 2505
2427 AH328.txt 2531
2428 AH329.txt 2494
2429 AH33.txt 39502
2430 AH330.txt 2516
2431 AH331.txt 2350
2432 AH332.txt 2289
2433 AH333.txt 2357
2434 AH334.txt 2269
2435 AH335.txt 2246
2436 AH336.txt 2257
2437 AH337.txt 2359
2438 AH338.txt 2340
2439 AH339.txt 2339
2440 AH34.txt 34188
2441 AH340.txt 2286
2442 AH341.txt 2288
2443 AH342.txt 2254
2444 AH343.txt 2365
2445 AH344.txt 2259
2446 AH345.txt 2272
2447 AH346.txt 2313
2448 AH347.txt 2281
2449 AH348.txt 2236
2450 AH349.txt 2361
2451 AH35.txt 42029
2452 AH350.txt 2307
2453 AH351.txt 2252
2454 AH352.txt 2253
2455 AH353.txt 2297
2456 AH354.txt 2372
2457 AH355.txt 2352
2458 AH356.txt 2323
2459 AH357.txt 2330
2460 AH358.txt 2347
2461 AH359.txt 2382
2462 AH36.txt 56111
2463 AH360.txt 2310
2464 AH37.txt 52323
2465 AH38.txt 33488
2466 AH39.txt 49074
2467 AH4.txt 34228
2468 AH40.txt 35146
2469 AH41.txt 41680
2470 AH42.txt 36306
2471 AH43.txt 37996
2472 AH44.txt 29747
2473 AH45.txt 40247
2474 AH46.txt 35893
2475 AH47.txt 43164
2476 AH48.txt 21820
2477 AH49.txt 44244
2478 AH5.txt 38813
2479 AH50.txt 55434
2480 AH51.txt 29117
2481 AH52.txt 37150
2482 AH53.txt 52028
2483 AH54.txt 31960
2484 AH55.txt 57142
2485 AH56.txt 45993
2486 AH57.txt 41263
2487 AH58.txt 37768
2488 AH59.txt 44250
2489 AH6.txt 45112
2490 AH60.txt 40646
2491 AH61.txt 2813
2492 AH62.txt 2725
2493 AH63.txt 2940
2494 AH64.txt 2852
2495 AH65.txt 2735
2496 AH66.txt 2247
2497 AH67.txt 2845
2498 AH68.txt 2618
2499 AH69.txt 2380
2500 AH7.txt 39448
2501 AH70.txt 2190
2502 AH71.txt 2931
2503 AH72.txt 2135
2504 AH73.txt 2079
2505 AH74.txt 2489

2506 AH75.txt 2705
2507 AH76.txt 3070
2508 AH77.txt 2252
2509 AH78.txt 2148
2510 AH79.txt 2612
2511 AH8.txt 41698
2512 AH80.txt 2461
2513 AH81.txt 2670
2514 AH82.txt 2110
2515 AH83.txt 2833
2516 AH84.txt 2243
2517 AH85.txt 2730
2518 AH86.txt 1737
2519 AH87.txt 2441
2520 AH88.txt 2503
2521 AH89.txt 1989
2522 AH9.txt 40739
2523 AH90.txt 2593
2524 AH91.txt 3049
2525 AH92.txt 1681
2526 AH93.txt 2674
2527 AH94.txt 2685
2528 AH95.txt 2058
2529 AH96.txt 2026
2530 AH97.txt 2018
2531 AH98.txt 2760
2532 AH99.txt 1826
2568 babu.txt 375
2036 beng01.txt 30
2037 beng02.txt 57
2038 beng03.txt 84
2039 beng04.txt 107
2040 beng05.txt 134
2041 beng06.txt 36
2042 beng07.txt 67
2043 beng08.txt 101
2044 beng09.txt 126
2045 beng10.txt 156
2644 C_1_1.txt 70
2645 C_1_10.txt 69
2646 C_1_11.txt 91
2647 C_1_12.txt 108
2648 C_1_13.txt 145
2649 C_1_14.txt 129
2650 C_1_15.txt 140
2651 C_1_16.txt 122
2652 C_1_17.txt 109
2653 C_1_18.txt 169
2654 C_1_19.txt 102
2655 C_1_2.txt 44
2656 C_1_20.txt 103
2657 C_1_21.txt 217
2658 C_1_22.txt 181
2659 C_1_23.txt 194
2660 C_1_24.txt 209
2661 C_1_25.txt 165
2662 C_1_26.txt 168
2663 C_1_27.txt 150
2664 C_1_28.txt 196
2665 C_1_29.txt 175
2666 C_1_3.txt 73
2667 C_1_30.txt 230
2668 C_1_31.txt 240
2669 C_1_32.txt 254
2670 C_1_33.txt 265
2671 C_1_34.txt 261
2672 C_1_35.txt 245
2673 C_1_36.txt 263
2674 C_1_37.txt 289
2675 C_1_38.txt 282
2676 C_1_39.txt 282
2677 C_1_4.txt 48
2678 C_1_40.txt 245
2679 C_1_41.txt 274
2680 C_1_42.txt 306
2681 C_1_43.txt 267
2682 C_1_44.txt 293
2683 C_1_45.txt 309
2684 C_1_46.txt 355
2685 C_1_47.txt 276
2686 C_1_48.txt 315
2687 C_1_49.txt 300
2688 C_1_5.txt 54
2689 C_1_50.txt 353
2690 C_1_6.txt 78
2691 C_1_7.txt 55
2692 C_1_8.txt 52

2693 C_1_9.txt 69
2594 C_10_451.txt 547
2595 C_10_452.txt 257
2596 C_10_453.txt 385
2597 C_10_454.txt 376
2598 C_10_455.txt 331
2599 C_10_456.txt 317
2600 C_10_457.txt 417
2601 C_10_458.txt 198
2602 C_10_459.txt 463
2603 C_10_460.txt 203
2604 C_10_461.txt 742
2605 C_10_462.txt 679
2606 C_10_463.txt 837
2607 C_10_464.txt 561
2608 C_10_465.txt 519
2609 C_10_466.txt 505
2610 C_10_467.txt 615
2611 C_10_468.txt 645
2612 C_10_469.txt 735
2613 C_10_470.txt 749
2614 C_10_471.txt 1057
2615 C_10_472.txt 1148
2616 C_10_473.txt 1051
2617 C_10_474.txt 705
2618 C_10_475.txt 733
2619 C_10_476.txt 1200
2620 C_10_477.txt 912
2621 C_10_478.txt 925
2622 C_10_479.txt 805
2623 C_10_480.txt 746
2624 C_10_481.txt 1197
2625 C_10_482.txt 997
2626 C_10_483.txt 1025
2627 C_10_484.txt 1319
2628 C_10_485.txt 1319
2629 C_10_486.txt 1246
2630 C_10_487.txt 1363
2631 C_10_488.txt 951
2632 C_10_489.txt 1182
2633 C_10_490.txt 1386
2634 C_10_491.txt 1381
2635 C_10_492.txt 1484
2636 C_10_493.txt 1518
2637 C_10_494.txt 1673
2638 C_10_495.txt 1683
2639 C_10_496.txt 1238
2640 C_10_497.txt 1306
2641 C_10_498.txt 1741
2642 C_10_499.txt 1529
2643 C_10_500.txt 1445
2694 C_2_100.txt 114
2695 C_2_51.txt 22
2696 C_2_52.txt 15
2697 C_2_53.txt 22
2698 C_2_54.txt 16
2699 C_2_55.txt 18
2700 C_2_56.txt 25
2701 C_2_57.txt 18
2702 C_2_58.txt 17
2703 C_2_59.txt 21
2704 C_2_60.txt 23
2705 C_2_61.txt 30
2706 C_2_62.txt 36
2707 C_2_63.txt 47
2708 C_2_64.txt 42
2709 C_2_65.txt 46
2710 C_2_66.txt 36
2711 C_2_67.txt 36
2712 C_2_68.txt 49
2713 C_2_69.txt 34
2714 C_2_70.txt 35
2715 C_2_71.txt 67
2716 C_2_72.txt 59
2717 C_2_73.txt 61
2718 C_2_74.txt 61
2719 C_2_75.txt 55
2720 C_2_76.txt 56
2721 C_2_77.txt 50
2722 C_2_78.txt 63
2723 C_2_79.txt 58
2724 C_2_80.txt 71
2725 C_2_81.txt 76
2726 C_2_82.txt 81
2727 C_2_83.txt 77
2728 C_2_84.txt 82
2729 C_2_85.txt 79

2730 C_2_86.txt 85
2731 C_2_87.txt 89
2732 C_2_88.txt 89
2733 C_2_89.txt 93
2734 C_2_90.txt 81
2735 C_2_91.txt 92
2736 C_2_92.txt 102
2737 C_2_93.txt 89
2738 C_2_94.txt 97
2739 C_2_95.txt 101
2740 C_2_96.txt 113
2741 C_2_97.txt 92
2742 C_2_98.txt 105
2743 C_2_99.txt 100
2744 C_3_101.txt 186
2745 C_3_102.txt 114
2746 C_3_103.txt 191
2747 C_3_104.txt 131
2748 C_3_105.txt 143
2749 C_3_106.txt 196
2750 C_3_107.txt 142
2751 C_3_108.txt 139
2752 C_3_109.txt 176
2753 C_3_110.txt 206
2754 C_3_111.txt 229
2755 C_3_112.txt 299
2756 C_3_113.txt 379
2757 C_3_114.txt 352
2758 C_3_115.txt 403
2759 C_3_116.txt 347
2760 C_3_117.txt 308
2761 C_3_118.txt 464
2762 C_3_119.txt 273
2763 C_3_120.txt 273
2764 C_3_121.txt 590
2765 C_3_122.txt 464
2766 C_3_123.txt 503
2767 C_3_124.txt 524
2768 C_3_125.txt 438
2769 C_3_126.txt 459
2770 C_3_127.txt 415
2771 C_3_128.txt 526
2772 C_3_129.txt 477
2773 C_3_130.txt 648
2774 C_3_131.txt 667
2775 C_3_132.txt 695
2776 C_3_133.txt 658
2777 C_3_134.txt 701
2778 C_3_135.txt 629
2779 C_3_136.txt 735
2780 C_3_137.txt 759
2781 C_3_138.txt 797
2782 C_3_139.txt 788
2783 C_3_140.txt 657
2784 C_3_141.txt 723
2785 C_3_142.txt 854
2786 C_3_143.txt 710
2787 C_3_144.txt 768
2788 C_3_145.txt 860
2789 C_3_146.txt 981
2790 C_3_147.txt 774
2791 C_3_148.txt 845
2792 C_3_149.txt 811
2793 C_3_150.txt 1040
2794 C_4_151.txt 71
2795 C_4_152.txt 43
2796 C_4_153.txt 72
2797 C_4_154.txt 51
2798 C_4_155.txt 58
2799 C_4_156.txt 79
2800 C_4_157.txt 57
2801 C_4_158.txt 53
2802 C_4_159.txt 69
2803 C_4_160.txt 77
2804 C_4_161.txt 91
2805 C_4_162.txt 115
2806 C_4_163.txt 147
2807 C_4_164.txt 133
2808 C_4_165.txt 153
2809 C_4_166.txt 113
2810 C_4_167.txt 117
2811 C_4_168.txt 161
2812 C_4_169.txt 108
2813 C_4_170.txt 109
2814 C_4_171.txt 218
2815 C_4_172.txt 184
2816 C_4_173.txt 199

2817 C_4_174.txt 198
2818 C_4_175.txt 176
2819 C_4_176.txt 179
2820 C_4_177.txt 161
2821 C_4_178.txt 207
2822 C_4_179.txt 187
2823 C_4_180.txt 235
2824 C_4_181.txt 244
2825 C_4_182.txt 256
2826 C_4_183.txt 250
2827 C_4_184.txt 263
2828 C_4_185.txt 252
2829 C_4_186.txt 277
2830 C_4_187.txt 292
2831 C_4_188.txt 291
2832 C_4_189.txt 300
2833 C_4_190.txt 261
2834 C_4_191.txt 289
2835 C_4_192.txt 334
2836 C_4_193.txt 284
2837 C_4_194.txt 307
2838 C_4_195.txt 325
2839 C_4_196.txt 370
2840 C_4_197.txt 296
2841 C_4_198.txt 336
2842 C_4_199.txt 317
2843 C_4_200.txt 374
2844 C_5_201.txt 648
2845 C_5_202.txt 365
2846 C_5_203.txt 622
2847 C_5_204.txt 415
2848 C_5_205.txt 456
2849 C_5_206.txt 695
2850 C_5_207.txt 446
2851 C_5_208.txt 431
2852 C_5_209.txt 600
2853 C_5_210.txt 657
2854 C_5_211.txt 733
2855 C_5_212.txt 966
2856 C_5_213.txt 1317
2857 C_5_214.txt 1111
2858 C_5_215.txt 1233
2859 C_5_216.txt 1106
2860 C_5_217.txt 980
2861 C_5_218.txt 1497
2862 C_5_219.txt 882
2863 C_5_220.txt 902
2864 C_5_221.txt 1853
2865 C_5_222.txt 1551
2866 C_5_223.txt 1667
2867 C_5_224.txt 1844
2868 C_5_225.txt 1395
2869 C_5_226.txt 1487
2870 C_5_227.txt 1295
2871 C_5_228.txt 1692
2872 C_5_229.txt 1498
2873 C_5_230.txt 2132
2874 C_5_231.txt 2125
2875 C_5_232.txt 2200
2876 C_5_233.txt 2328
2877 C_5_234.txt 2280
2878 C_5_235.txt 2057
2879 C_5_236.txt 2332
2880 C_5_237.txt 2509
2881 C_5_238.txt 2474
2882 C_5_239.txt 2467
2883 C_5_240.txt 2096
2884 C_5_241.txt 2309
2885 C_5_242.txt 2748
2886 C_5_243.txt 2261
2887 C_5_244.txt 2484
2888 C_5_245.txt 2658
2889 C_5_246.txt 3132
2890 C_5_247.txt 2401
2891 C_5_248.txt 2708
2892 C_5_249.txt 2577
2893 C_5_250.txt 3150
2894 C_6_251.txt 185
2895 C_6_252.txt 115
2896 C_6_253.txt 191
2897 C_6_254.txt 135
2898 C_6_255.txt 152
2899 C_6_256.txt 208
2900 C_6_257.txt 153
2901 C_6_258.txt 141
2902 C_6_259.txt 181
2903 C_6_260.txt 202

2904 C_6_261.txt 237
2905 C_6_262.txt 305
2906 C_6_263.txt 394
2907 C_6_264.txt 355
2908 C_6_265.txt 405
2909 C_6_266.txt 299
2910 C_6_267.txt 308
2911 C_6_268.txt 425
2912 C_6_269.txt 288
2913 C_6_270.txt 290
2914 C_6_271.txt 580
2915 C_6_272.txt 493
2916 C_6_273.txt 527
2917 C_6_274.txt 528
2918 C_6_275.txt 468
2919 C_6_276.txt 474
2920 C_6_277.txt 424
2921 C_6_278.txt 549
2922 C_6_279.txt 496
2923 C_6_280.txt 625
2924 C_6_281.txt 648
2925 C_6_282.txt 679
2926 C_6_283.txt 665
2927 C_6_284.txt 694
2928 C_6_285.txt 669
2929 C_6_286.txt 733
2930 C_6_287.txt 775
2931 C_6_288.txt 780
2932 C_6_289.txt 800
2933 C_6_290.txt 695
2934 C_6_291.txt 768
2935 C_6_292.txt 889
2936 C_6_293.txt 754
2937 C_6_294.txt 815
2938 C_6_295.txt 865
2939 C_6_296.txt 987
2940 C_6_297.txt 783
2941 C_6_298.txt 889
2942 C_6_299.txt 845
2943 C_6_300.txt 992
2944 C_7_301.txt 478
2945 C_7_302.txt 459
2946 C_7_303.txt 437
2947 C_7_304.txt 648
2948 C_7_305.txt 544
2949 C_7_306.txt 503
2950 C_7_307.txt 372
2951 C_7_308.txt 605
2952 C_7_309.txt 591
2953 C_7_310.txt 382
2954 C_7_311.txt 948
2955 C_7_312.txt 1187
2956 C_7_313.txt 902
2957 C_7_314.txt 1366
2958 C_7_315.txt 909
2959 C_7_316.txt 1059
2960 C_7_317.txt 1104
2961 C_7_318.txt 1076
2962 C_7_319.txt 789
2963 C_7_320.txt 1250
2964 C_7_321.txt 1694
2965 C_7_322.txt 1341
2966 C_7_323.txt 1657
2967 C_7_324.txt 1423
2968 C_7_325.txt 1420
2969 C_7_326.txt 1459
2970 C_7_327.txt 1459
2971 C_7_328.txt 1668
2972 C_7_329.txt 1385
2973 C_7_330.txt 1790
2974 C_7_331.txt 1997
2975 C_7_332.txt 2439
2976 C_7_333.txt 1994
2977 C_7_334.txt 2091
2978 C_7_335.txt 2304
2979 C_7_336.txt 2201
2980 C_7_337.txt 2345
2981 C_7_338.txt 2202
2982 C_7_339.txt 2299
2983 C_7_340.txt 2349
2984 C_7_341.txt 2663
2985 C_7_342.txt 2629
2986 C_7_343.txt 2426
2987 C_7_344.txt 2603
2988 C_7_345.txt 2443
2989 C_7_346.txt 2759
2990 C_7_347.txt 2631

2991 C_7_348.txt 2814
2992 C_7_349.txt 2448
2993 C_7_350.txt 3023
2994 C_8_351.txt 497
2995 C_8_352.txt 507
2996 C_8_353.txt 461
2997 C_8_354.txt 519
2998 C_8_355.txt 426
2999 C_8_356.txt 486
3000 C_8_357.txt 359
3001 C_8_358.txt 433
3002 C_8_359.txt 525
3003 C_8_360.txt 319
3004 C_8_361.txt 969
3005 C_8_362.txt 1084
3006 C_8_363.txt 904
3007 C_8_364.txt 1013
3008 C_8_365.txt 741
3009 C_8_366.txt 967
3010 C_8_367.txt 902
3011 C_8_368.txt 943
3012 C_8_369.txt 787
3013 C_8_370.txt 1074
3014 C_8_371.txt 1487
3015 C_8_372.txt 1455
3016 C_8_373.txt 1428
3017 C_8_374.txt 1260
3018 C_8_375.txt 1240
3019 C_8_376.txt 1311
3020 C_8_377.txt 1238
3021 C_8_378.txt 1499
3022 C_8_379.txt 1392
3023 C_8_380.txt 1516
3024 C_8_381.txt 1868
3025 C_8_382.txt 1970
3026 C_8_383.txt 1757
3027 C_8_384.txt 1752
3028 C_8_385.txt 2224
3029 C_8_386.txt 1907
3030 C_8_387.txt 1927
3031 C_8_388.txt 1976
3032 C_8_389.txt 1917
3033 C_8_390.txt 2123
3034 C_8_391.txt 2303
3035 C_8_392.txt 2327
3036 C_8_393.txt 2101
3037 C_8_394.txt 2579
3038 C_8_395.txt 2450
3039 C_8_396.txt 2299
3040 C_8_397.txt 2302
3041 C_8_398.txt 2494
3042 C_8_399.txt 2382
3043 C_8_400.txt 2791
3044 C_9_401.txt 1423
3045 C_9_402.txt 946
3046 C_9_403.txt 1076
3047 C_9_404.txt 1180
3048 C_9_405.txt 1269
3049 C_9_406.txt 1156
3050 C_9_407.txt 730
3051 C_9_408.txt 1068
3052 C_9_409.txt 1250
3053 C_9_410.txt 970
3054 C_9_411.txt 1910
3055 C_9_412.txt 2406
3056 C_9_413.txt 2257
3057 C_9_414.txt 2419
3058 C_9_415.txt 2083
3059 C_9_416.txt 2330
3060 C_9_417.txt 1896
3061 C_9_418.txt 2199
3062 C_9_419.txt 1734
3063 C_9_420.txt 2672
3064 C_9_421.txt 3622
3065 C_9_422.txt 3550
3066 C_9_423.txt 3719
3067 C_9_424.txt 3440
3068 C_9_425.txt 3146
3069 C_9_426.txt 2902
3070 C_9_427.txt 3217
3071 C_9_428.txt 3626
3072 C_9_429.txt 3296
3073 C_9_430.txt 3586
3074 C_9_431.txt 4481
3075 C_9_432.txt 4555
3076 C_9_433.txt 4491
3077 C_9_434.txt 4409

3078 C_9_435.txt 4792
3079 C_9_436.txt 4800
3080 C_9_437.txt 4828
3081 C_9_438.txt 4673
3082 C_9_439.txt 4110
3083 C_9_440.txt 4742
3084 C_9_441.txt 5613
3085 C_9_442.txt 5066
3086 C_9_443.txt 5217
3087 C_9_444.txt 5858
3088 C_9_445.txt 4956
3089 C_9_446.txt 5754
3090 C_9_447.txt 5179
3091 C_9_448.txt 5821
3092 C_9_449.txt 5148
3093 C_9_450.txt 5737
2130 cgcut1.txt 23
2131 cgcut2.txt 64
2132 cgcut3.txt 660
2569 gcut01.txt 1016
2570 gcut02.txt 1187
2571 gcut03.txt 1803
2572 gcut04.txt 3004
2573 gcut05.txt 1273
2574 gcut06.txt 2622
2575 gcut07.txt 4693
2576 gcut08.txt 5877
2577 gcut09.txt 2241
2578 gcut10.txt 6393
2579 gcut11.txt 7736
2580 gcut12.txt 13172
2581 gcut13.txt 4994
3154 i10-1 164513
3163 i10-10 164827
3155 i10-2 165235
3156 i10-3 165031
3157 i10-4 164608
3158 i10-5 164893
3159 i10-6 164617
3160 i10-7 164661
3161 i10-8 164779
3162 i10-9 164489
3164 i11-1 231591
3173 i11-10 231882
3165 i11-2 232087
3166 i11-3 231952
3167 i11-4 231713
3168 i11-5 231881
3169 i11-6 231649
3170 i11-7 231773
3171 i11-8 231444
3172 i11-9 231947
3174 i12-1 326637
3183 i12-10 326705
3175 i12-2 326764
3176 i12-3 326818
3177 i12-4 326514
3178 i12-5 326697
3179 i12-6 326754
3180 i12-7 326453
3181 i12-8 326424
3182 i12-9 326603
3184 i13-1 460879
3193 i13-10 460945
3185 i13-2 461553
3186 i13-3 461418
3187 i13-4 460840
3188 i13-5 460730
3189 i13-6 461371
3190 i13-7 460802
3191 i13-8 460562
3192 i13-9 460721
3194 i14-1 650744
3203 i14-10 650556
3195 i14-2 651398
3196 i14-3 651267
3197 i14-4 650414
3198 i14-5 650495
3199 i14-6 651086
3200 i14-7 650502
3201 i14-8 650187
3202 i14-9 650419
3204 i15-1 918279
3213 i15-10 918320
3205 i15-2 918749
3206 i15-3 918332
3207 i15-4 917771

3208 i15-5 917839
3209 i15-6 918315
3210 i15-7 918027
3211 i15-8 917692
3212 i15-9 918411
3214 i16-1 1296100
3223 i16-10 1295430
3215 i16-2 1295710
3216 i16-3 1295710
3217 i16-4 1295960
3218 i16-5 1296010
3219 i16-6 1295460
3220 i16-7 1295380
3221 i16-8 1295990
3222 i16-9 1295370
3224 i17-1 1830550
3233 i17-10 1829580
3225 i17-2 1829700
3226 i17-3 1829620
3227 i17-4 1830590
3228 i17-5 1829560
3229 i17-6 1829690
3230 i17-7 1829510
3231 i17-8 1830410
3232 i17-9 1829520
3234 i18-1 2586310
3243 i18-10 2586190
3235 i18-2 2586330
3236 i18-3 2586270
3237 i18-4 2586220
3238 i18-5 2586130
3239 i18-6 2586340
3240 i18-7 2586180
3241 i18-8 2587080
3242 i18-9 2586190
3244 i19-1 3656990
3253 i19-10 3656910
3245 i19-2 3656780
3246 i19-3 3656980
3247 i19-4 3656940
3248 i19-5 3656900
3249 i19-6 3657000
3250 i19-7 3656900
3251 i19-8 3656890
3252 i19-9 3656910
3254 i20-1 5171480
3263 i20-10 5171270
3255 i20-2 5171320
3256 i20-3 5171310
3257 i20-4 5171290
3258 i20-5 5171270
3259 i20-6 5171490
3260 i20-7 5171280
3261 i20-8 5171410
3262 i20-9 5171430
3094 i4-1 20200
3103 i4-10 20200
3095 i4-2 20200
3096 i4-3 20200
3097 i4-4 20200
3098 i4-5 20200
3099 i4-6 20200
3100 i4-7 20200
3101 i4-8 20200
3102 i4-9 20200
3104 i5-1 28567
3113 i5-10 28567
3105 i5-2 28567
3106 i5-3 28567
3107 i5-4 28567
3108 i5-5 28567
3109 i5-6 28567
3110 i5-7 28567
3111 i5-8 28567
3112 i5-9 28567
3114 i6-1 40400
3123 i6-10 40400
3115 i6-2 41389
3116 i6-3 40400
3117 i6-4 41386
3118 i6-5 40905
3119 i6-6 40400
3120 i6-7 41299
3121 i6-8 40400
3122 i6-9 40400
3124 i7-1 58521
3133 i7-10 58530

3125 i7-2 59209
3126 i7-3 58772
3127 i7-4 58642
3128 i7-5 58204
3129 i7-6 58493
3130 i7-7 58354
3131 i7-8 58399
3132 i7-9 58546
3134 i8-1 82653
3143 i8-10 83252
3135 i8-2 83126
3136 i8-3 83174
3137 i8-4 82870
3138 i8-5 82855
3139 i8-6 82276
3140 i8-7 83213
3141 i8-8 82769
3142 i8-9 82697
3144 i9-1 116714
3153 i9-10 117369
3145 i9-2 117611
3146 i9-3 116945
3147 i9-4 116764
3148 i9-5 116627
3149 i9-6 116647
3150 i9-7 117042
3151 i9-8 116667
3152 i9-9 116807
2145 lw161.txt 20
2146 lw163.txt 20
2147 lw172.txt 20
2148 lw1961.txt 241
2149 lw1963.txt 241
2150 lw1972.txt 241
2151 lw251.txt 15
2152 lw252.txt 15
2153 lw253.txt 15
2154 lw281.txt 30
2155 lw283.txt 30
2156 lw292.txt 30
2157 lw491.txt 60
2158 lw492.txt 61
2159 lw493.txt 60
2160 lw731.txt 90
2161 lw732.txt 90
2162 lw733.txt 90
2163 lw971.txt 121
2164 lw972.txt 120
2165 lw973.txt 121
2082 n1.txt 40
2083 n10.txt 150
2084 n11.txt 150
2085 n12.txt 300
2086 n13.txt 961
2095 N1a.txt 200
2096 N1b.txt 200
2097 N1c.txt 200
2098 N1d.txt 200
2099 N1e.txt 200
2087 n2.txt 50
2100 N2a.txt 200
2101 N2b.txt 200
2102 N2c.txt 200
2103 N2d.txt 200
2104 N2e.txt 200
2088 n3.txt 50
2105 N3a.txt 200
2106 N3b.txt 200
2107 N3c.txt 204
2108 N3d.txt 200
2109 N3e.txt 200
2089 n4.txt 80
2110 N4a.txt 204
2111 N4b.txt 205
2112 N4c.txt 204
2113 N4d.txt 203
2114 N4e.txt 204
2090 n5.txt 100
2115 N5a.txt 203
2116 N5b.txt 203
2117 N5c.txt 203
2118 N5d.txt 203
2119 N5e.txt 203
2091 n6.txt 100
2120 N6a.txt 202
2121 N6b.txt 202
2122 N6c.txt 202

2123 N6d.txt 203
2124 N6e.txt 202
2092 n7.txt 100
2125 N7a.txt 201
2126 N7b.txt 201
2127 N7c.txt 201
2128 N7d.txt 201
2129 N7e.txt 201
2093 n8.txt 81
2094 n9.txt 150
2582 ngcut01.txt 23
2583 ngcut02.txt 30
2584 ngcut03.txt 28
2585 ngcut04.txt 20
2586 ngcut05.txt 36
2587 ngcut06.txt 31
2588 ngcut07.txt 20
2589 ngcut08.txt 33
2590 ngcut09.txt 50
2591 ngcut10.txt 80
2592 ngcut11.txt 52
2593 ngcut12.txt 87
2139 nice1.txt 1032
2000 nice1.txt.txt 1032
2001 nice101t.txt 1005
2002 nice102t.txt 1003
2003 nice105t.txt 1001
2004 nice11t.txt 1005
2005 nice12t.txt 1002
2006 nice15t.txt 1001
2140 nice2.txt 1020
2007 nice2.txt 1023
2008 nice21t.txt 1004
2009 nice22t.txt 1003
2010 nice25t.txt 1001
2141 nice3.txt 1027
2011 nice3.txt 1030
2012 nice31t.txt 1004
2013 nice32t.txt 1003
2014 nice35t.txt 1002
2142 nice4.txt 1018
2015 nice4.txt 1022
2016 nice41t.txt 1003
2017 nice42t.txt 1001
2018 nice45t.txt 1001
2143 nice5.txt 1009
2019 nice5.txt 1009
2020 nice51t.txt 1004
2021 nice52t.txt 1002
2022 nice55t.txt 1002
2144 nice6.txt 1004
2023 nice6.txt 1003
2024 nice61t.txt 1003
2025 nice62t.txt 1002
2026 nice65t.txt 1001
2027 nice71t.txt 1004
2028 nice72t.txt 1003
2029 nice75t.txt 1002
2030 nice81t.txt 1004
2031 nice82t.txt 1002
2032 nice85t.txt 1001
2033 nice91t.txt 1004
2034 nice92t.txt 1003
2035 nice95t.txt 1001
2133 path1.txt 1028
2046 path1.txt 1028
2047 path101t.txt 1004
2048 path102t.txt 1003
2049 path105t.txt 1002
2050 path11t.txt 1003
2051 path12t.txt 1002
2052 path15t.txt 1000
2134 path2.txt 1001
2053 path2.txt 1001
2054 path21t.txt 1004
2055 path22t.txt 1003
2056 path25t.txt 999
2135 path3.txt 1018
2057 path3.txt 1019
2058 path31t.txt 1005
2059 path32t.txt 1002
2060 path35t.txt 1000
2136 path4.txt 1015
2061 path4.txt 1016
2062 path41t.txt 1003
2063 path42t.txt 999
2064 path45t.txt 995

2137 path5.txt 1011
2065 path5.txt 1012
2066 path51t.txt 1008
2067 path52t.txt 1004
2068 path55t.txt 1004
2138 path6.txt 1007
2069 path6.txt 1008
2070 path61t.txt 1004
2071 path62t.txt 1003
2072 path65t.txt 1000
2073 path71t.txt 1002
2074 path72t.txt 999
2075 path75t.txt 998
2076 path81t.txt 1004
2077 path82t.txt 1001
2078 path85t.txt 997
2079 path91t.txt 1003
2080 path92t.txt 1003
2081 path95t.txt 997
3264 spritepack-2000.in 1195
3265 spritepack-2001.in 300
3266 spritepack-2002.in 1600
3267 spritepack-2003.in 296
3268 spritepack-2004.in 185
3269 spritepack-2005.in 904
3270 spritepack-2006.in 969
3271 spritepack-2007.in 1988
3272 spritepack-2008.in 3389
3273 spritepack-2009.in 1242
3274 spritepack-2010.in 233
3275 spritepack-2011.in 537
3276 spritepack-2012.in 225
3277 spritepack-2013.in 2004
3278 spritepack-2014.in 7912
3279 spritepack-2015.in 1578
3280 spritepack-2016.in 140
3281 spritepack-2017.in 439
3282 spritepack-2018.in 1222
3283 spritepack-2019.in 678
3284 spritepack-2020.in 252
3285 spritepack-2021.in 1142
3286 spritepack-2022.in 132
3287 spritepack-2023.in 210
3288 spritepack-2024.in 332
3289 spritepack-2025.in 2167
3290 spritepack-2026.in 2118
3291 spritepack-2027.in 518
3292 spritepack-2028.in 137
3293 spritepack-2029.in 1175
3294 spritepack-2030.in 660
3295 spritepack-2031.in 178
2533 T1a.txt 200
2534 T1b.txt 200
2535 T1c.txt 200
2536 T1d.txt 200
2537 T1e.txt 200
2538 T2a.txt 200
2539 T2b.txt 200
2540 T2c.txt 200
2541 T2d.txt 200
2542 T2e.txt 200
2543 T3a.txt 200
2544 T3b.txt 200
2545 T3c.txt 200
2546 T3d.txt 200
2547 T3e.txt 200
2548 T4a.txt 203
2549 T4b.txt 203
2550 T4c.txt 203
2551 T4d.txt 203
2552 T4e.txt 204
2553 T5a.txt 203
2554 T5b.txt 202
2555 T5c.txt 203
2556 T5d.txt 203
2557 T5e.txt 203
2558 T6a.txt 202
2559 T6b.txt 202
2560 T6c.txt 202
2561 T6d.txt 202
2562 T6e.txt 202
2563 T7a.txt 201
2564 T7b.txt 201
2565 T7c.txt 201
2566 T7d.txt 201
2567 T7e.txt 201
3296 tag-clouds-2032.in 1046

3297 tag-clouds-2033.in 853
3298 tag-clouds-2034.in 178
3299 tag-clouds-2035.in 592
3300 tag-clouds-2036.in 1613
3301 tag-clouds-2037.in 346
3302 tag-clouds-2038.in 2676
3303 tag-clouds-2039.in 783
3304 tag-clouds-2040.in 369
3305 tag-clouds-2041.in 225

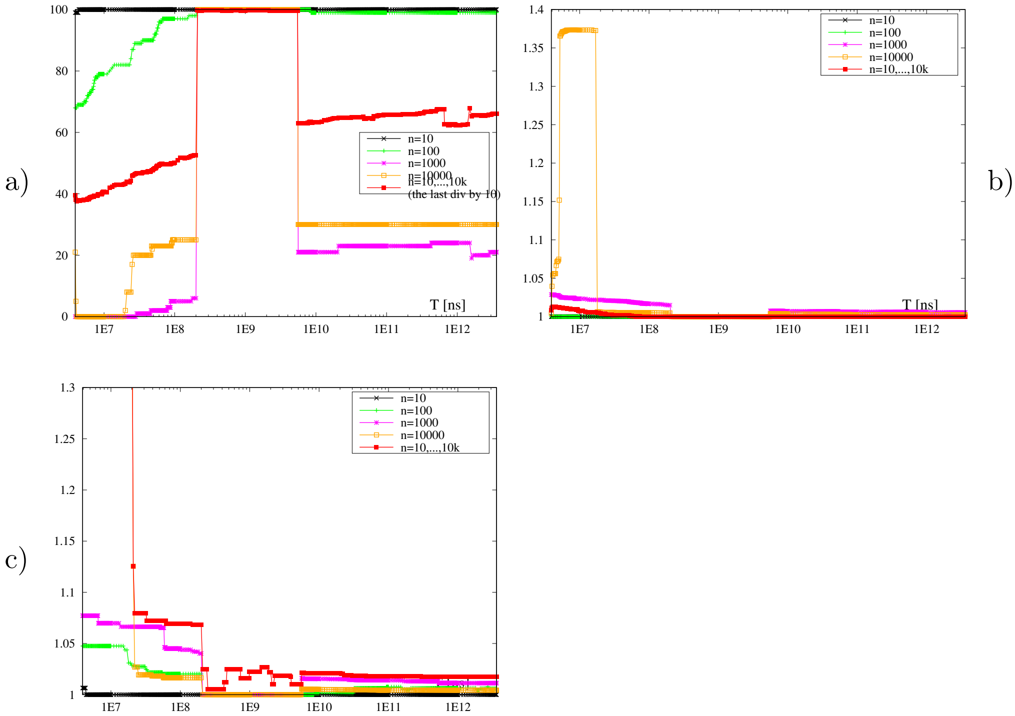

Figure 19: Curiosity – what if the literature instances were the training dataset and random dataset1 were a validation dataset? a) Number of wins, b) median relative distance from the best solution found, c) worst relative distance from the best solution found.

8 Literature Instances as Training Dataset

The lines going out of the Y-range in Fig.19c show that there are instances for which no solutions a re found by $T = 1.6\text{E}7$ns. In Fig.19a and in $T \in$[2.1E8,5.5E9]ns algorithms shBFDHr and skB join and leave the portfolio built on the literature dataset. Compare the results of literature dataset as validation instances when dataset1 was a training set in Fig.6c, Fig.9b, Fig.12b (in these figures training dataset1 was denoted $n = 10, \dots, 1000$). The number of wins is bigger when dataset1 (random) is training and literature dataset is validating (Fig.6c). An exception are instances with size $n = 10$ which are well solved by a portfolio trained on literature instances. It can be understood that small size instances are well represented in the literature dataset (probably for historical heritage reasons). When trained on whole dataset1 and tested on literature dataset, the median relative distance from the best solution found is smaller than in the opposite training-validation setting (Fig.9b vs Fig.19b). For $T \leq 1.9\text{E}9$ns the raining on the literature dataset is better (than on random dataset1) with respect to the worst case relative distance because more instances had any solution under limited runtime (Fig.19c vs Fig.12b). It seems that in the literature dataset there is a subset of instances (considering the known number of wins, fewer than 40 in 1306 instances) which are not well represented in the random dataset1 and the resulting portfolio. Then, for these few instances the changes of the portfolio membership with the runtime limit is not well coordinated with the algorithms which really solve such instances within the allowed runtime. Hence, many holes in the picture in Fig.12b.

It can be concluded that literature instances are mixed blessing as training dataset. On the one hand, for majority vote measures (wins, median relative distance from the best solution found) the literature dataset is not better for training a portfolio than the random dataset1. On the other hand, the literature dataset has some instances on which the random-instance-trained portfolio performs weakly. This is visible in maximum relative distance from the best solution found (a minority vote quality measure). The literature instances were often built as pathological instances for certain algorithms, and hence, these instances have certain set of biases which are hardly represented in random datasets. A solution to this situation is to start with random training portfolios and then include literature datasets and any odd instances emerging over time.

9 Acknowledgments

This research has been partially supported by a grant of French National Research Ministry and PHC *Polonium* project for years 2017-2018. Computational experiments have been conducted in Poznań Supercomputing and Networking Center. We are grateful to Shinji Imahori for providing IY instances.

References

[1] J. E. Beasley, An exact two-dimensional non-guillotine cutting tree search procedure. Operational Research, vol. 33, pp. 49 - 64, 1985. see also `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/binpacktwo.xls`

[2] Edmund K. Burke, Graham Kendall, Glenn Whitwell, A New Placement Heuristic for the Orthogonal Stock-Cutting Problem, Operations Research, Vol. 52, No. 4, 2004, pp. 655–671

[3] E. Hopper, B.C.H. Turton, An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem, European Journal of Operational Research 128 (2001) 34–57

[4] Imahori,S., Yagiura, M., 2010. The best-fit heuristic for the rectangular strip packing problem: An efficient implementation and the worst-case approximation ratio, Comp. & Oper. Res. 37, 325–333.

[5] Richard E. Korf, Michael D. Moffitt, Martha E. Pollack, Optimal rectangle packing, Ann Oper Res (2010) 179: 261–295, DOI 10.1007/s10479-008-0463-6

[6] Silvano Martello, Michele Monaci, Daniele Vigo, An Exact Approach to the Strip-Packing Problem, INFORMS Journal on Computing, Vol. 15, No. 3, Summer 2003, pp. 310–319

[7] Silvano Martello, Michele Monaci, Models and algorithms for packing rectangles into the smallest square, Computers & Operations Research 63(2015)161–171

[8] Jakub Marszałkowski, Dariusz Mokwa, Maciej Drozdowski, Analysis and Solution of CSS-Sprite Packing Problem, ACM Transactions on the Web, Vol. 10, No. 1, Article 1, 2015, http://dx.doi.org/10.1145/2818377

[9] Jakub Marszałkowski, Dariusz Mokwa, Maciej Drozdowski, Łukasz Rusiecki, Hubert Narożny, Fast algorithms for online construction of web tag clouds, Engineering Applications of Artificial Intelligence 64 (2017) 378–390 http://dx.doi.org/10.1016/j.engappai.2017.06.023

[10] Kamil Piechowiak, Maciej Drozdowski, Éric Sanlaville, Framework of Algorithm Portfolios for Strip Packing Problem, 2021, submitted.

[11] Kamil Piechowiak, Maciej Drozdowski, Éric Sanlaville, Algorithm Portfolios for 2D Strip Packing – Extended Results. Institute of Computing Science, Poznań University of Technology, Research Report RA-1/2020, 2020 http://www.cs.put.poznan.pl/mdrozdowski/2sp-asp/TechRept-1-20.pdf

[12] Wei, L., Hu, Q., Leung, S.C.H., Zhang, N., 2017. An improved skyline based heuristic for the 2D strip packing problem and its efficient implementation, Comp. & Oper. Res. 80, 12–24.

[13] Wei, L., Zhu, W., Skyline Heuristic for the 2D Rectangular Packing and Strip Packing Problems, https://www.computational-logistics.org/orlib/topic/2D%20Strip%20Packing/Data/Data.zip [accessed: 30.04.2021]