Numerical values and statistical results for the experimental results

January 11, 2007

The absolute values for all the results in the form of tables. In these tables, we present the numerical values of the final solution quality and the corresponding computational time for each probabiltiy range. Moreover, in order to test that the observed difference between the cost of the solutions reached by the considered algorithms are significant in a statistical sense, we use a non-parametric test known as Wilcoxon test. We also tabulate the p-values obtained from the Wilcoxon test. Please note that all statistics shown in the table are given for 100 instances at each probability level.

Table 1: Experimental results for `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on clustered instance of size 100. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are obtained on 100 instances for each probability level.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	`2.5-opt-EEs-100`	1132343	68046	0.067	0.017
	`2.5-opt-ACs`	1101016	61787	0.218	0.044
	`1-shift`	1098886	63110	0.340	0.095
	`2-p-opt`	1126890	65891	0.640	0.206
$p = 0.2$	`2.5-opt-EEs-100`	1670833	97442	0.031	0.007
	`2.5-opt-ACs`	1671529	99000	0.121	0.031
	`1-shift`	1662024	102018	0.305	0.097
	`2-p-opt`	1741020	117661	0.404	0.146
$p = 0.3$	`2.5-opt-EEs-100`	2087844	128515	0.020	0.005
	`2.5-opt-ACs`	2085521	127850	0.080	0.018
	`1-shift`	2088694	125446	0.260	0.087
	`2-p-opt`	2174071	142414	0.290	0.092
$p = 0.4$	`2.5-opt-EEs-100`	2420165	163908	0.015	0.004
	`2.5-opt-ACs`	2415510	151857	0.060	0.013
	`1-shift`	2441960	149124	0.216	0.084
	`2-p-opt`	2501604	159722	0.226	0.071
$p = 0.5$	`2.5-opt-EEs-100`	2688278	184701	0.012	0.003
	`2.5-opt-ACs`	2690344	171318	0.050	0.012
	`1-shift`	2746319	169654	0.177	0.058
	`2-p-opt`	2766786	176589	0.196	0.065
$p = 0.6$	`2.5-opt-EEs-100`	2924229	205232	0.011	0.003
	`2.5-opt-ACs`	2919191	197583	0.044	0.010
	`1-shift`	2996130	186335	0.156	0.053
	`2-p-opt`	2989179	198395	0.162	0.054
$p = 0.7$	`2.5-opt-EEs-100`	3125656	221498	0.010	0.002
	`2.5-opt-ACs`	3114247	219530	0.040	0.011
	`1-shift`	3225168	214666	0.141	0.044
	`2-p-opt`	3188901	219545	0.140	0.044
$p = 0.8$	`2.5-opt-EEs-100`	3309884	241626	0.009	0.002
	`2.5-opt-ACs`	3308512	229808	0.038	0.010
	`1-shift`	3412359	236564	0.136	0.050
	`2-p-opt`	3362156	235897	0.135	0.045
$p = 0.9$	`2.5-opt-EEs-100`	3468105	256429	0.008	0.002
	`2.5-opt-ACs`	3469370	234731	0.036	0.009
	`1-shift`	3582461	247875	0.127	0.044
	`2-p-opt`	3508904	254379	0.130	0.045

Table 2: The p-values of the comparison of `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on clustered instance of size 100. Each algorithm is allowed to run until it reaches a local optimum. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in the row performs significantly better than the algorithm in the column, while values in italic mean that the algorithm in the column performs significantly better than the algorithm in the row.

		p-values			
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	*0.000*	*0.000*	*0.004*
$p = 0.1$	`2.5-opt-ACs`	**0.000**	-	**0.000**	**0.000**
	`1-shift`	**0.000**	*0.000*	-	**0.000**
	`2-p-opt`	**0.004**	*0.000*	*0.000*	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	0.963	0.182	**0.000**
$p = 0.2$	`2.5-opt-ACs`	0.963	-	0.197	**0.000**
	`1-shift`	0.182	0.197	-	**0.000**
	`2-p-opt`	*0.000*	*0.000*	*0.000*	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	1.000	1.000	**0.000**
$p = 0.3$	`2.5-opt-ACs`	1.000	-	1.000	**0.000**
	`1-shift`	1.000	1.000	-	**0.000**
	`2-p-opt`	*0.000*	*0.000*	*0.000*	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	0.630	*0.018*	**0.000**
$p = 0.4$	`2.5-opt-ACs`	0.630	-	*0.014*	**0.000**
	`1-shift`	**0.018**	**0.014**	-	**0.000**
	`2-p-opt`	*0.000*	*0.000*	*0.000*	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	0.302	**0.000**	**0.000**
$p = 0.5$	`2.5-opt-ACs`	0.302	-	**0.000**	**0.000**
	`1-shift`	*0.000*	*0.000*	-	**0.036**
	`2-p-opt`	*0.000*	*0.000*	*0.036*	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	0.934	**0.000**	**0.000**
$p = 0.6$	`2.5-opt-ACs`	0.934	-	**0.000**	**0.000**
	`1-shift`	*0.000*	*0.000*	-	0.934
	`2-p-opt`	*0.000*	*0.000*	0.934	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	0.292	**0.000**	**0.000**
$p = 0.7$	`2.5-opt-ACs`	0.292	-	**0.000**	**0.000**
	`1-shift`	*0.000*	*0.000*	-	*0.005*
	`2-p-opt`	*0.000*	*0.000*	**0.005**	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	0.790	**0.000**	**0.000**
$p = 0.8$	`2.5-opt-ACs`	0.790	-	**0.000**	**0.000**
	`1-shift`	*0.000*	*0.000*	-	*0.000*
	`2-p-opt`	*0.000*	*0.000*	**0.000**	-
		`2.5-opt-EEs-100`	`2.5-opt-ACs`	`1-shift`	`2-p-opt`
	`2.5-opt-EEs-100`	-	0.814	**0.000**	**0.000**
$p = 0.9$	`2.5-opt-ACs`	0.814	-	**0.000**	**0.000**
	`1-shift`	*0.000*	*0.000*	-	*0.000*
	`2-p-opt`	*0.000*	*0.000*	**0.000**	-

Table 3: Experimental results for `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on clustered instance of size 200. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are obtained on 100 instances for each probability level.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	`2.5-opt-EEs-100`	2140467	484519	0.176	0.032
	`2.5-opt-ACs`	2100804	478778	1.687	0.251
	`1-shift`	2105953	476183	3.005	0.699
	`2-p-opt`	2184084	470597	6.165	1.716
$p = 0.2$	`2.5-opt-EEs-100`	2824227	495026	0.085	0.014
	`2.5-opt-ACs`	2814249	500834	0.915	0.179
	`1-shift`	2816343	497281	2.819	0.752
	`2-p-opt`	2933186	497643	3.834	1.047
$p = 0.3$	`2.5-opt-EEs-100`	3353023	524583	0.058	0.011
	`2.5-opt-ACs`	3348289	525096	0.612	0.110
	`1-shift`	3397536	512711	2.279	0.562
	`2-p-opt`	3487135	518725	2.718	0.719
$p = 0.4$	`2.5-opt-EEs-100`	3786619	555142	0.047	0.008
	`2.5-opt-ACs`	3787052	551377	0.472	0.093
	`1-shift`	3857909	553435	2.031	0.561
	`2-p-opt`	3921061	547841	2.070	0.568
$p = 0.5$	`2.5-opt-EEs-100`	4141461	577366	0.040	0.007
	`2.5-opt-ACs`	4140953	578252	0.383	0.069
	`1-shift`	4254132	572026	1.608	0.471
	`2-p-opt`	4262524	571931	1.730	0.417
$p = 0.6$	`2.5-opt-EEs-100`	4450012	586268	0.035	0.006
	`2.5-opt-ACs`	4457370	600716	0.332	0.065
	`1-shift`	4605097	594791	1.447	0.361
	`2-p-opt`	4568950	587538	1.463	0.372
$p = 0.7$	`2.5-opt-EEs-100`	4724429	620592	0.032	0.005
	`2.5-opt-ACs`	4724352	618312	0.306	0.068
	`1-shift`	4891234	612112	1.326	0.317
	`2-p-opt`	4817318	602476	1.329	0.310
$p = 0.8$	`2.5-opt-EEs-100`	4953206	627118	0.028	0.004
	`2.5-opt-ACs`	4953204	618092	0.283	0.056
	`1-shift`	5150361	645942	1.223	0.289
	`2-p-opt`	5042943	618123	1.212	0.282
$p = 0.9$	`2.5-opt-EEs-100`	5171842	650956	0.026	0.004
	`2.5-opt-ACs`	5184706	641782	0.277	0.057
	`1-shift`	5389853	670442	1.133	0.281
	`2-p-opt`	5234034	630773	1.146	0.275

Table 4: The p-values of the comparison of `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on clustered instance of size 200. Each algorithm is allowed to run until it reaches a local optimum. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in the row performs significantly better than the algorithm in the column, while values in italic mean that the algorithm in the column performs significantly better than the algorithm in the row.

		p-values			
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	*0.000*	*0.000*	**0.000**
$p = 0.1$	2.5-opt-ACs	**0.000**	-	0.484	**0.000**
	1-shift	**0.000**	0.484	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.282	0.787	**0.000**
$p = 0.2$	2.5-opt-ACs	0.282	-	0.854	**0.000**
	1-shift	0.787	0.854	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.865	**0.000**	**0.000**
$p = 0.3$	2.5-opt-ACs	0.865	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.763	**0.000**	**0.000**
$p = 0.4$	2.5-opt-ACs	0.763	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.877	**0.000**	**0.000**
$p = 0.5$	2.5-opt-ACs	0.877	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	0.693
	2-p-opt	*0.000*	*0.000*	0.693	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.330	**0.000**	**0.000**
$p = 0.6$	2.5-opt-ACs	0.330	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.033*
	2-p-opt	*0.000*	*0.000*	**0.033**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.999	**0.000**	**0.000**
$p = 0.7$	2.5-opt-ACs	0.999	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.814	**0.000**	**0.000**
$p = 0.8$	2.5-opt-ACs	0.814	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
	2.5-opt-EEs-100	-	0.227	**0.000**	**0.000**
$p = 0.9$	2.5-opt-ACs	0.227	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-

Table 5: Experimental results for `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on clustered instance of size 300. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are obtained on 100 instances for each probability level.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	`2.5-opt-EEs-100`	2776865	456487	0.120	0.014
	`2.5-opt-ACs`	2730221	454321	6.453	1.067
	`1-shift`	2738026	450970	11.771	2.404
	`2-p-opt`	2870013	462383	22.440	5.831
$p = 0.2$	`2.5-opt-EEs-100`	3595283	467721	0.086	0.011
	`2.5-opt-ACs`	3585254	471967	3.413	0.540
	`1-shift`	3606878	467069	10.103	1.773
	`2-p-opt`	3775106	474269	13.848	3.254
$p = 0.3$	`2.5-opt-EEs-100`	4239788	499001	0.064	0.008
	`2.5-opt-ACs`	4259032	501810	2.214	0.399
	`1-shift`	4286461	481061	8.478	1.856
	`2-p-opt`	4429328	497857	9.842	2.462
$p = 0.4$	`2.5-opt-EEs-100`	4764400	517350	0.052	0.006
	`2.5-opt-ACs`	4769245	519437	1.672	0.303
	`1-shift`	4861195	518759	7.154	1.452
	`2-p-opt`	4936775	511292	7.389	1.867
$p = 0.5$	`2.5-opt-EEs-100`	5190835	537186	0.046	0.005
	`2.5-opt-ACs`	5201221	557895	1.421	0.230
	`1-shift`	5336979	544328	5.933	1.355
	`2-p-opt`	5352456	553028	5.978	1.616
$p = 0.6$	`2.5-opt-EEs-100`	5552434	564439	0.042	0.004
	`2.5-opt-ACs`	5568872	586369	1.231	0.204
	`1-shift`	5744938	577165	5.358	1.215
	`2-p-opt`	5689353	567193	5.315	1.204
$p = 0.7$	`2.5-opt-EEs-100`	5874044	579475	0.038	0.004
	`2.5-opt-ACs`	5875100	579015	1.127	0.209
	`1-shift`	6087249	597356	4.851	1.056
	`2-p-opt`	5993481	589339	4.776	1.146
$p = 0.8$	`2.5-opt-EEs-100`	6154030	586937	0.037	0.004
	`2.5-opt-ACs`	6170881	593782	1.042	0.190
	`1-shift`	6401881	611155	4.399	0.972
	`2-p-opt`	6259008	598772	4.366	1.094
$p = 0.9$	`2.5-opt-EEs-100`	6412335	602907	0.036	0.004
	`2.5-opt-ACs`	6428845	602878	1.020	0.220
	`1-shift`	6683735	628833	4.112	0.937
	`2-p-opt`	6491451	604388	4.093	0.954

Table 6: The p-values of the comparison of `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on clustered instance of size 300. Each algorithm is allowed to run until it reaches a local optimum. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in the row performs significantly better than the algorithm in the column, while values in italic mean that the algorithm in the column performs significantly better than the algorithm in the row.

		p-values			
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.1$	2.5-opt-EEs-100	-	*0.000*	*0.000*	**0.000**
	2.5-opt-ACs	**0.000**	-	0.402	**0.000**
	1-shift	**0.000**	0.402	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.2$	2.5-opt-EEs-100	-	0.153	0.542	**0.000**
	2.5-opt-ACs	0.153	-	0.153	**0.000**
	1-shift	0.542	0.153	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.3$	2.5-opt-EEs-100	-	0.094	**0.000**	**0.000**
	2.5-opt-ACs	0.094	-	**0.017**	**0.000**
	1-shift	*0.000*	*0.017*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.4$	2.5-opt-EEs-100	-	0.432	**0.000**	**0.000**
	2.5-opt-ACs	0.432	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.5$	2.5-opt-EEs-100	-	0.679	**0.000**	**0.000**
	2.5-opt-ACs	0.679	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	0.217
	2-p-opt	*0.000*	*0.000*	0.217	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.6$	2.5-opt-EEs-100	-	0.153	**0.000**	**0.000**
	2.5-opt-ACs	0.153	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.7$	2.5-opt-EEs-100	-	0.892	**0.000**	**0.000**
	2.5-opt-ACs	0.892	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.8$	2.5-opt-EEs-100	-	0.517	**0.000**	**0.000**
	2.5-opt-ACs	0.517	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.9$	2.5-opt-EEs-100	-	0.160	**0.000**	**0.000**
	2.5-opt-ACs	0.160	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-

Table 7: Experimental results for `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on uniform instance of size 100. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are obtained on 100 instances for each probability level.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	`2.5-opt-EEs-100`	3004617	80822	0.042	0.013
	`2.5-opt-ACs`	2965191	91964	0.154	0.038
	`1-shift`	2954116	106033	0.278	0.096
	`2-p-opt`	3026572	118881	0.575	0.238
$p = 0.2$	`2.5-opt-EEs-100`	4186071	138265	0.022	0.007
	`2.5-opt-ACs`	4180097	150929	0.091	0.024
	`1-shift`	4232072	252595	0.264	0.084
	`2-p-opt`	4357901	205513	0.395	0.162
$p = 0.3$	`2.5-opt-EEs-100`	5087051	177199	0.015	0.005
	`2.5-opt-ACs`	5108405	191495	0.064	0.018
	`1-shift`	5180119	256342	0.230	0.079
	`2-p-opt`	5338532	216215	0.261	0.115
$p = 0.4$	`2.5-opt-EEs-100`	5769740	193997	0.012	0.003
	`2.5-opt-ACs`	5762558	185992	0.050	0.013
	`1-shift`	5883667	280138	0.200	0.068
	`2-p-opt`	6003318	208905	0.209	0.093
$p = 0.5$	`2.5-opt-EEs-100`	6332259	209153	0.010	0.002
	`2.5-opt-ACs`	6317676	199147	0.041	0.010
	`1-shift`	6522967	285099	0.173	0.063
	`2-p-opt`	6568786	257262	0.172	0.067
$p = 0.6$	`2.5-opt-EEs-100`	6820376	218170	0.009	0.002
	`2.5-opt-ACs`	6800091	212596	0.037	0.010
	`1-shift`	7058396	299514	0.155	0.060
	`2-p-opt`	7019421	276668	0.150	0.058
$p = 0.7$	`2.5-opt-EEs-100`	7242897	232260	0.008	0.002
	`2.5-opt-ACs`	7206654	214336	0.033	0.008
	`1-shift`	7522742	326437	0.141	0.051
	`2-p-opt`	7396710	259831	0.138	0.055
$p = 0.8$	`2.5-opt-EEs-100`	7605806	239102	0.007	0.002
	`2.5-opt-ACs`	7579628	257114	0.033	0.008
	`1-shift`	7933117	375477	0.137	0.049
	`2-p-opt`	7743227	314326	0.124	0.047
$p = 0.9$	`2.5-opt-EEs-100`	7947072	259587	0.006	0.001
	`2.5-opt-ACs`	7875565	256763	0.031	0.008
	`1-shift`	8280679	405128	0.126	0.046
	`2-p-opt`	8012462	311871	0.121	0.049

Table 8: The p-values of the comparison of `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on uniform instance of size 100. Each algorithm is allowed to run until it reaches a local optimum. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in the row performs significantly better than the algorithm in the column, while values in italic mean that the algorithm in the column performs significantly better than the algorithm in the row.

		p-values			
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.1$	2.5-opt-EEs-100	-	*0.000*	*0.000*	0.473
	2.5-opt-ACs	**0.000**	-	*0.000*	**0.000**
	1-shift	**0.000**	**0.000**	-	**0.000**
	2-p-opt	0.473	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.2$	2.5-opt-EEs-100	-	0.805	1.000	**0.000**
	2.5-opt-ACs	0.805	-	1.000	**0.000**
	1-shift	1.000	1.000	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.3$	2.5-opt-EEs-100	-	0.281	**0.015**	**0.000**
	2.5-opt-ACs	0.281	-	**0.028**	**0.000**
	1-shift	*0.015*	*0.028*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.4$	2.5-opt-EEs-100	-	0.413	**0.002**	**0.000**
	2.5-opt-ACs	0.413	-	**0.001**	**0.000**
	1-shift	*0.002*	*0.001*	-	**0.001**
	2-p-opt	*0.000*	*0.000*	*0.001*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.5$	2.5-opt-EEs-100	-	0.323	**0.000**	**0.000**
	2.5-opt-ACs	0.323	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	0.112
	2-p-opt	*0.000*	*0.000*	0.112	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.6$	2.5-opt-EEs-100	-	0.544	**0.000**	**0.000**
	2.5-opt-ACs	0.544	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	0.544
	2-p-opt	*0.000*	*0.000*	0.544	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.7$	2.5-opt-EEs-100	-	0.096	**0.000**	**0.000**
	2.5-opt-ACs	0.096	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.006*
	2-p-opt	*0.000*	*0.000*	**0.006**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.8$	2.5-opt-EEs-100	-	0.180	**0.000**	**0.000**
	2.5-opt-ACs	0.180	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.9$	2.5-opt-EEs-100	-	*0.002*	**0.000**	**0.038**
	2.5-opt-ACs	**0.002**	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.038*	*0.000*	**0.000**	-

Table 9: Experimental results for `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on uniform instance of size 200. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are obtained on 100 instances for each probability level.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	`2.5-opt-EEs-100`	4269851	92873	0.130	0.029
	`2.5-opt-ACs`	4187333	102432	1.234	0.222
	`1-shift`	4273859	229387	2.840	0.797
	`2-p-opt`	4483576	203827	5.480	2.112
$p = 0.2$	`2.5-opt-EEs-100`	5934729	159285	0.061	0.012
	`2.5-opt-ACs`	5934236	158997	0.670	0.135
	`1-shift`	6055657	259170	2.280	0.641
	`2-p-opt`	6367788	236029	3.122	1.117
$p = 0.3$	`2.5-opt-EEs-100`	7143750	196688	0.042	0.010
	`2.5-opt-ACs`	7149590	204459	0.460	0.096
	`1-shift`	7281495	281335	1.987	0.593
	`2-p-opt`	7572106	227783	2.192	0.760
$p = 0.4$	`2.5-opt-EEs-100`	8088249	209149	0.033	0.005
	`2.5-opt-ACs`	8080543	210513	0.360	0.071
	`1-shift`	8343138	311581	1.643	0.534
	`2-p-opt`	8487661	257061	1.672	0.560
$p = 0.5$	`2.5-opt-EEs-100`	8847997	235166	0.030	0.004
	`2.5-opt-ACs`	8822829	233329	0.298	0.061
	`1-shift`	9240847	360545	1.343	0.379
	`2-p-opt`	9212220	260377	1.381	0.411
$p = 0.6$	`2.5-opt-EEs-100`	9481826	236587	0.026	0.004
	`2.5-opt-ACs`	9470390	235284	0.263	0.057
	`1-shift`	9947679	399932	1.276	0.361
	`2-p-opt`	9834284	249469	1.228	0.357
$p = 0.7$	`2.5-opt-EEs-100`	10068790	278156	0.024	0.003
	`2.5-opt-ACs`	10037203	237516	0.241	0.046
	`1-shift`	10586830	401200	1.153	0.310
	`2-p-opt`	10331056	266726	1.108	0.303
$p = 0.8$	`2.5-opt-EEs-100`	10562786	273859	0.022	0.003
	`2.5-opt-ACs`	10558013	286791	0.226	0.045
	`1-shift`	11114808	390724	1.042	0.287
	`2-p-opt`	10759240	278585	1.031	0.276
$p = 0.9$	`2.5-opt-EEs-100`	11019271	298569	0.021	0.003
	`2.5-opt-ACs`	10956829	266835	0.221	0.048
	`1-shift`	11567428	393799	1.007	0.257
	`2-p-opt`	11175595	305391	0.971	0.263

Table 10: The p-values of the comparison of `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on uniform instance of size 200. Each algorithm is allowed to run until it reaches a local optimum. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in the row performs significantly better than the algorithm in the column, while values in italic mean that the algorithm in the column performs significantly better than the algorithm in the row.

		p-values			
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.1$	2.5-opt-EEs-100	-	*0.000*	0.761	**0.000**
	2.5-opt-ACs	**0.000**	-	0.271	**0.000**
	1-shift	0.761	0.271	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.2$	2.5-opt-EEs-100	-	0.774	**0.000**	**0.000**
	2.5-opt-ACs	0.774	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.3$	2.5-opt-EEs-100	-	0.901	**0.000**	**0.000**
	2.5-opt-ACs	0.901	-	**0.001**	**0.000**
	1-shift	*0.000*	*0.001*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.4$	2.5-opt-EEs-100	-	0.332	**0.000**	**0.000**
	2.5-opt-ACs	0.332	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.5$	2.5-opt-EEs-100	-	0.327	**0.000**	**0.000**
	2.5-opt-ACs	0.327	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	0.552
	2-p-opt	*0.000*	*0.000*	0.552	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.6$	2.5-opt-EEs-100	-	0.297	**0.000**	**0.000**
	2.5-opt-ACs	0.297	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.011*
	2-p-opt	*0.000*	*0.000*	**0.011**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.7$	2.5-opt-EEs-100	-	0.090	**0.000**	**0.000**
	2.5-opt-ACs	0.090	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.8$	2.5-opt-EEs-100	-	0.865	**0.000**	**0.000**
	2.5-opt-ACs	0.865	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.9$	2.5-opt-EEs-100	-	*0.001*	**0.000**	**0.000**
	2.5-opt-ACs	**0.001**	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-

Table 11: Experimental results for `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on uniform instance of size 300. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are obtained on 100 instances for each probability level.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	`2.5-opt-EEs-100`	5229780	113745	0.233	0.040
	`2.5-opt-ACs`	5137062	127732	5.370	0.813
	`1-shift`	5215083	217290	10.977	2.586
	`2-p-opt`	5649818	202396	20.501	6.046
$p = 0.2$	`2.5-opt-EEs-100`	7252967	174695	0.114	0.019
	`2.5-opt-ACs`	7236755	175205	2.706	0.510
	`1-shift`	7394412	261680	8.961	2.027
	`2-p-opt`	7780474	240973	12.080	3.615
$p = 0.3$	`2.5-opt-EEs-100`	8708749	196082	0.078	0.012
	`2.5-opt-ACs`	8724669	172038	1.780	0.281
	`1-shift`	8990908	306602	7.226	1.838
	`2-p-opt`	9258841	252958	8.209	2.597
$p = 0.4$	`2.5-opt-EEs-100`	9829218	213338	0.065	0.010
	`2.5-opt-ACs`	9831544	210342	1.421	0.287
	`1-shift`	10222090	338275	5.981	1.589
	`2-p-opt`	10358501	251618	6.262	1.702
$p = 0.5$	`2.5-opt-EEs-100`	10715877	215604	0.056	0.007
	`2.5-opt-ACs`	10757582	224458	1.176	0.205
	`1-shift`	11211640	341825	5.286	1.355
	`2-p-opt`	11217449	278671	5.189	1.161
$p = 0.6$	`2.5-opt-EEs-100`	11474117	213549	0.050	0.005
	`2.5-opt-ACs`	11511946	233220	1.035	0.186
	`1-shift`	12072398	377126	4.617	1.075
	`2-p-opt`	11921609	245596	4.621	1.208
$p = 0.7$	`2.5-opt-EEs-100`	12169080	242735	0.046	0.005
	`2.5-opt-ACs`	12154391	239848	0.965	0.154
	`1-shift`	12818014	408812	4.132	0.938
	`2-p-opt`	12543492	269934	4.098	1.007
$p = 0.8$	`2.5-opt-EEs-100`	12761274	271566	0.043	0.004
	`2.5-opt-ACs`	12751274	252100	0.898	0.153
	`1-shift`	13509104	418653	3.937	0.911
	`2-p-opt`	13080404	260668	3.846	0.898
$p = 0.9$	`2.5-opt-EEs-100`	13297593	295394	0.041	0.003
	`2.5-opt-ACs`	13295392	276115	0.871	0.147
	`1-shift`	14103919	441445	3.666	0.807
	`2-p-opt`	13592119	284643	3.597	0.793

Table 12: The p-values of the comparison of `2.5-opt-EEs-100`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on uniform instance of size 300. Each algorithm is allowed to run until it reaches a local optimum. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in the row performs significantly better than the algorithm in the column, while values in italic mean that the algorithm in the column performs significantly better than the algorithm in the row.

		p-values			
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.1$	2.5-opt-EEs-100	-	*0.000*	0.422	**0.000**
	2.5-opt-ACs	**0.000**	-	**0.005**	**0.000**
	1-shift	0.422	*0.005*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.2$	2.5-opt-EEs-100	-	0.506	**0.000**	**0.000**
	2.5-opt-ACs	0.506	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.3$	2.5-opt-EEs-100	-	0.171	**0.000**	**0.000**
	2.5-opt-ACs	0.171	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.4$	2.5-opt-EEs-100	-	0.881	**0.000**	**0.000**
	2.5-opt-ACs	0.881	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	**0.000**
	2-p-opt	*0.000*	*0.000*	*0.000*	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.5$	2.5-opt-EEs-100	-	**0.041**	**0.000**	**0.000**
	2.5-opt-ACs	*0.041*	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	0.960
	2-p-opt	*0.000*	*0.000*	0.960	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.6$	2.5-opt-EEs-100	-	0.051	**0.000**	**0.000**
	2.5-opt-ACs	0.051	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.7$	2.5-opt-EEs-100	-	0.512	**0.000**	**0.000**
	2.5-opt-ACs	0.512	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.8$	2.5-opt-EEs-100	-	0.591	**0.000**	**0.000**
	2.5-opt-ACs	0.591	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-
		2.5-opt-EEs-100	2.5-opt-ACs	1-shift	2-p-opt
$p = 0.9$	2.5-opt-EEs-100	-	0.985	**0.000**	**0.000**
	2.5-opt-ACs	0.985	-	**0.000**	**0.000**
	1-shift	*0.000*	*0.000*	-	*0.000*
	2-p-opt	*0.000*	*0.000*	**0.000**	-

Table 13: Experimental results for `2.5-opt-EEs-10`, `2.5-opt-EEs-100`, `2.5-opt-EEs-1000`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on clustered instances of size 1000. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are given for 100 instances at each probability level. The symbol × indicates that the algorithms do not produce meaningful results due to the numerical problems.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	`2.5-opt-EEs-10`	5909938	461029	0.462	0.030
	`2.5-opt-EEs-100`	5158215	448385	1.839	0.190
	`2.5-opt-EEs-1000`	5069560	424448	9.487	1.170
	`2.5-opt-ACs`	5068223	450709	443.952	70.934
	`1-shift`	5178144	469977	635.757	84.010
	`2-p-opt`	5365486	449318	1464.535	341.993
$p = 0.2$	`2.5-opt-EEs-10`	7364518	513937	0.524	0.032
	`2.5-opt-EEs-100`	6692459	486598	1.024	0.092
	`2.5-opt-EEs-1000`	6681179	475423	3.981	0.447
	`2.5-opt-ACs`	6697814	480609	229.288	33.165
	`1-shift`	6744906	494658	547.263	71.878
	`2-p-opt`	6978843	477590	859.276	159.102
$p = 0.3$	`2.5-opt-EEs-10`	8263425	554699	0.507	0.030
	`2.5-opt-EEs-100`	7894854	547385	0.722	0.051
	`2.5-opt-EEs-1000`	7875735	511413	2.658	0.306
	`2.5-opt-ACs`	7901717	524412	149.881	22.702
	`1-shift`	7982498	531787	451.773	58.575
	`2-p-opt`	8175022	547812	552.554	95.447
$p = 0.4$	`2.5-opt-EEs-10`	9025641	585240	0.459	0.031
	`2.5-opt-EEs-100`	8830141	596439	0.600	0.041
	`2.5-opt-EEs-1000`	8846373	594006	2.026	0.198
	`2.5-opt-ACs`	8848198	549139	106.665	17.015
	`1-shift`	8995824	567472	374.877	50.938
	`2-p-opt`	9060142	551377	422.211	76.872
$p = 0.5$	`2.5-opt-EEs-10`	9693061	630069	0.422	0.020
	`2.5-opt-EEs-100`	9592605	623310	0.526	0.038
	`2.5-opt-EEs-1000`	9591076	635788	1.689	0.149
	`2.5-opt-ACs`	9597432	599270	89.272	14.155
	`1-shift`	9856073	579796	316.049	44.883
	`2-p-opt`	9799426	594452	338.203	63.679
$p = 0.6$	`2.5-opt-EEs-10`	10282713	609678	0.391	0.018
	`2.5-opt-EEs-100`	10238620	648265	0.481	0.027
	`2.5-opt-EEs-1000`	10264821	637652	1.444	0.129
	`2.5-opt-ACs`	×	×	×	×
	`1-shift`	×	×	×	×
	`2-p-opt`	×	×	×	×
$p = 0.7$	`2.5-opt-EEs-10`	10849469	680716	0.370	0.017
	`2.5-opt-EEs-100`	10809915	627073	0.446	0.024
	`2.5-opt-EEs-1000`	10782974	669036	1.280	0.108
	`2.5-opt-ACs`	×	×	×	×
	`1-shift`	×	×	×	×
	`2-p-opt`	×	×	×	×
$p = 0.8$	`2.5-opt-EEs-10`	11323598	678658	0.355	0.015
	`2.5-opt-EEs-100`	11325374	667811	0.424	0.020
	`2.5-opt-EEs-1000`	11326928	665397	1.126	0.086
	`2.5-opt-ACs`	×	×	×	×
	`1-shift`	×	×	×	×
	`2-p-opt`	×	×	×	×
$p = 0.9$	`2.5-opt-EEs-10`	11754423	697164	0.346	0.013
	`2.5-opt-EEs-100`	11764130	697751	0.406	0.019
	`2.5-opt-EEs-1000`	11764290	705736	1.016	0.073
	`2.5-opt-ACs`	×	×	×	×
	`1-shift`	×	×	×	×
	`2-p-opt`	×	×	×	×

Table 14: Experimental results for `2.5-opt-EEs-10`, `2.5-opt-EEs-100`, `2.5-opt-EEs-1000`, `2.5-opt-ACs`, `2-p-opt`, and `1-shift` on uniform instances of size 1000. Each algorithm is allowed to run until it reaches a local optimum. The table gives mean and standard deviation (s.d.) of final solution cost and computation time in seconds. The results are given for 100 instances at each probability level. The symbol × indicates that the algorithms do not produce meaningful results due to the numerical problems.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
$p = 0.1$	2.5-opt-EEs-10	11447234	334497	0.425	0.025
	2.5-opt-EEs-100	9462476	89942	1.545	0.146
	2.5-opt-EEs-1000	9269830	120191	7.632	0.843
	2.5-opt-ACs	9262667	130967	408.822	48.205
	1-shift	9607745	279474	654.269	102.922
	2-p-opt	10463072	208580	1300.300	208.656
$p = 0.2$	2.5-opt-EEs-10	14538697	373616	0.467	0.029
	2.5-opt-EEs-100	13115296	177193	0.833	0.063
	2.5-opt-EEs-1000	13087980	211982	2.895	0.304
	2.5-opt-ACs	13118704	193246	191.297	21.445
	1-shift	13531134	282877	494.232	76.083
	2-p-opt	14290298	243493	675.138	106.666
$p = 0.3$	2.5-opt-EEs-10	16377103	270987	0.449	0.026
	2.5-opt-EEs-100	15735995	224942	0.605	0.042
	2.5-opt-EEs-1000	15745743	214509	1.882	0.201
	2.5-opt-ACs	15764653	208216	123.161	15.819
	1-shift	16356877	314947	388.584	54.338
	2-p-opt	16826114	229383	443.682	60.790
$p = 0.4$	2.5-opt-EEs-10	18107524	265441	0.413	0.023
	2.5-opt-EEs-100	17765575	232469	0.506	0.033
	2.5-opt-EEs-1000	17709005	239328	1.480	0.152
	2.5-opt-ACs	17768576	229315	95.476	11.463
	1-shift	18585379	339164	315.373	46.439
	2-p-opt	18738270	230508	327.254	47.829
$p = 0.5$	2.5-opt-EEs-10	19543192	267281	0.387	0.019
	2.5-opt-EEs-100	19371881	262833	0.454	0.022
	2.5-opt-EEs-1000	19339568	236327	1.238	0.106
	2.5-opt-ACs	19370026	225319	78.407	8.974
	1-shift	20381838	356178	263.344	34.323
	2-p-opt	20242110	234211	270.131	43.578
$p = 0.6$	2.5-opt-EEs-10	20801947	236997	0.364	0.017
	2.5-opt-EEs-100	20700955	218847	0.422	0.022
	2.5-opt-EEs-1000	20686014	209310	1.068	0.088
	2.5-opt-ACs	×	×	×	×
	1-shift	×	×	×	×
	2-p-opt	×	×	×	×
$p = 0.7$	2.5-opt-EEs-10	21968789	294282	0.345	0.014
	2.5-opt-EEs-100	21882804	258048	0.397	0.019
	2.5-opt-EEs-1000	21862114	264868	0.946	0.076
	2.5-opt-ACs	×	×	×	×
	1-shift	×	×	×	×
	2-p-opt	×	×	×	×
$p = 0.8$	2.5-opt-EEs-10	22951573	258260	0.335	0.013
	2.5-opt-EEs-100	22898732	247331	0.381	0.016
	2.5-opt-EEs-1000	22940568	257664	0.852	0.069
	2.5-opt-ACs	×	×	×	×
	1-shift	×	×	×	×
	2-p-opt	×	×	×	×
$p = 0.9$	2.5-opt-EEs-10	23871638	276007	0.328	0.012
	2.5-opt-EEs-100	23848923	278687	0.369	0.015
	2.5-opt-EEs-1000	23870974	247610	0.781	0.045
	2.5-opt-ACs	×	×	×	×
	1-shift	×	×	×	×
	2-p-opt	×	×	×	×

Table 15: Experimental results for `ILS-2.5-opt-EEs-100`,`ILS-2.5-opt-EEs-1000`, and `ILS-2.5-opt-ACs` on clustered instance of size 1000. The table gives the mean and standard deviation (s.d.) regarding the final solution quality and computation time. The results are given for 100 instances at each probability level. Note that the algorithms based on the *analytical computation* techniques do not produce meaningful results for $p > 0.5$ due to the numerical problem. The algorithms based on the *empirical estimation* do not suffer from this problem.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
	`ILS-2.5-opt-EEs-100`	4979874	399713		
$p = 0.1$	`ILS-2.5-opt-EEs-1000`	4843776	394002	3261.500	362.575
	`ILS-2.5-opt-ACs`	4877318	396493		
	`ILS-2.5-opt-EEs-100`	6236305	433606		
$p = 0.2$	`ILS-2.5-opt-EEs-1000`	6212680	435542	1933.850	200.543
	`ILS-2.5-opt-ACs`	6376838	452242		
	`ILS-2.5-opt-EEs-100`	7244262	470112		
$p = 0.3$	`ILS-2.5-opt-EEs-1000`	7259346	471874	1306.395	144.981
	`ILS-2.5-opt-ACs`	7511449	492782		
	`ILS-2.5-opt-EEs-100`	8073761	501743		
$p = 0.4$	`ILS-2.5-opt-EEs-1000`	8120875	502695	1017.768	115.292
	`ILS-2.5-opt-ACs`	8450032	532982		
	`ILS-2.5-opt-EEs-100`	8787387	532259		
$p = 0.5$	`ILS-2.5-opt-EEs-1000`	8845621	525797	842.500	97.555
	`ILS-2.5-opt-ACs`	9235088	560814		

Table 16: p-values of the comparison of `ILS-2.5-opt-EEs-100`, `ILS-2.5-opt-EEs-1000`, and `ILS-2.5-opt-ACs` on clustered instance of size 1000. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in row performs better than algorithm in column, while values in italic mean that algorithm in column performs better than algorithm in row. The results are given for 100 instances at each probability level.

		p-values		
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	*0.000*	*0.000*
$p = 0.1$	`ILS-2.5-opt-EEs-1000`	**0.000**	-	**0.000**
	`ILS-2.5-opt-ACs`	**0.000**	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	*0.000*	**0.000**
$p = 0.2$	`ILS-2.5-opt-EEs-1000`	**0.000**	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	**0.000**	**0.000**
$p = 0.3$	`ILS-2.5-opt-EEs-1000`	*0.000*	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	**0.000**	**0.000**
$p = 0.4$	`ILS-2.5-opt-EEs-1000`	*0.000*	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	**0.000**	**0.000**
$p = 0.5$	`ILS-2.5-opt-EEs-1000`	*0.000*	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-

Table 17: Experimental results for `ILS-2.5-opt-EEs-100`,`ILS-2.5-opt-EEs-1000`, and `ILS-2.5-opt-ACs` on uniform instance of size 1000. The table gives the mean and standard deviation (s.d.) regarding the final solution quality and computation time. The results are given for 100 instances at each probability level. Note that the algorithms based on the *analytical computation* techniques do not produce meaningful results for $p > 0.5$ due to the numerical problem. The algorithms based on the *empirical estimation* do not suffer from this problem.

	Algorithm	Solution Cost		Computation Time	
		mean	s.d.	mean	s.d.
	`ILS-2.5-opt-EEs-100`	9206243	88337		
$p = 0.1$	`ILS-2.5-opt-EEs-1000`	8903742	59781	3090.567	282.429
	`ILS-2.5-opt-ACs`	8994873	69401		
	`ILS-2.5-opt-EEs-100`	12275894	85228		
$p = 0.2$	`ILS-2.5-opt-EEs-1000`	12222805	76168	1492.575	155.310
	`ILS-2.5-opt-ACs`	12589378	119467		
	`ILS-2.5-opt-EEs-100`	14611127	98569		
$p = 0.3$	`ILS-2.5-opt-EEs-1000`	14644106	107165	1015.882	107.653
	`ILS-2.5-opt-ACs`	15189334	141217		
	`ILS-2.5-opt-EEs-100`	16485391	118876		
$p = 0.4$	`ILS-2.5-opt-EEs-1000`	16589825	138826	774.907	79.887
	`ILS-2.5-opt-ACs`	17232291	152276		
	`ILS-2.5-opt-EEs-100`	18062288	141241		
$p = 0.5$	`ILS-2.5-opt-EEs-1000`	18188461	161601	661.669	76.693
	`ILS-2.5-opt-ACs`	18902260	159220		

Table 18: p-values of the comparison of `ILS-2.5-opt-EEs-100`, `ILS-2.5-opt-EEs-1000`, and `ILS-2.5-opt-ACs` on uniform instance of size 1000. The statistical test adopted is the paired Wilcoxon test with p-values adjusted by Holm's method. The confidence level is 95%. Values in bold mean that the algorithm in row performs better than algorithm in column, while values in italic mean that algorithm in column performs better than algorithm in row. The results are given for 100 instances at each probability level.

		p-values		
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	*0.000*	*0.000*
$p = 0.1$	`ILS-2.5-opt-EEs-1000`	**0.000**	-	**0.000**
	`ILS-2.5-opt-ACs`	**0.000**	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	*0.000*	**0.000**
$p = 0.2$	`ILS-2.5-opt-EEs-1000`	**0.000**	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	**0.000**	**0.000**
$p = 0.3$	`ILS-2.5-opt-EEs-1000`	*0.000*	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	**0.000**	**0.000**
$p = 0.4$	`ILS-2.5-opt-EEs-1000`	*0.000*	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-
		`ILS-2.5-opt-EEs-100`	`ILS-2.5-opt-EEs-1000`	`ILS-2.5-opt-ACs`
	`ILS-2.5-opt-EEs-100`	-	**0.000**	**0.000**
$p = 0.5$	`ILS-2.5-opt-EEs-1000`	*0.000*	-	**0.000**
	`ILS-2.5-opt-ACs`	*0.000*	*0.000*	-