Appendix. Online Supplement

A. Parameter Sensitivity Analysis

The enhanced late acceptance hill climbing (ELAHC) procedure reinforces the well-known late acceptance hill climbing (LAHC) method by a relaxed acceptance and replacement strategy (to increase the diversity of the search) and a fast incremental evaluation mechanism (to speed up the computation of cost function for the rank aggregation problem). The length of the history cost list ρ is the only parameter of both the LAHC and ELAHC methods, which strongly influences the convergence speed and solution quality.

To study the effect of the parameter ρ value, we experimentally compare the performance of both LAHC and ELAHC algorithms under different ρ values. Our experiments are conducted on ten representative instances, each solved ten times with $\hat{t} = 1000$ seconds. Detailed comparative results of LAHC and ELAHC algorithms are summarized in Tables 1 and 2, respectively. In these two tables, column 1 presents the instance name (Instance), columns 2-4 report the result of the algorithm with $\rho = 1$, identifying the best result ($\hat{f}$), the average result ($\bar{f}$) and the computation time ($\bar{t}$) over 10 runs. Columns 5-7, 8-10, 11-13 present the corresponding results of the algorithm for other three ρ values.

A.1. Sensitivity of LAHC to ρ

Table 1 Comparisons among LAHC Algorithms with Different ρ Values (i.e., $\rho \in \{1, 1000, 3000, 10000\}$)

	$\rho = 1$			$\rho = 1000$			$\rho = 3000$			$\rho = 10000$		
Instance	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$
MM050n0.001_02	575.780	576.170	0.303	575.680	575.802	3.376	575.620	575.730	9.755	**575.560**	**575.706**	31.835
MM050n0.001_11	561.720	561.968	0.258	561.520	561.668	3.450	**561.500**	561.588	10.233	561.520	**561.578**	33.134
MM100n0.200_01	**416.000**	416.050	1.329	**416.000**	416.040	16.694	**416.000**	**416.012**	48.043	**416.000**	416.014	159.280
MM100n0.200_05	**411.720**	411.730	1.342	**411.720**	411.732	17.407	**411.720**	**411.726**	49.882	**411.720**	411.730	162.409
MM150n0.100_08	1249.720	1249.876	7.092	1249.680	1249.836	43.814	1249.660	1249.736	120.153	**1249.620**	**1249.706**	392.398
MM150n0.100_17	1266.190	1266.262	6.572	**1266.110**	1266.190	45.945	1266.130	1266.196	122.270	**1266.110**	**1266.160**	391.908
MM200n0.010_04	7669.290	7670.330	32.744	7668.570	7668.916	101.805	7668.010	7668.366	267.294	**7667.750**	**7668.006**	850.411
MM200n0.010_13	7706.640	7707.304	28.301	7705.140	7705.628	99.559	7704.820	7705.220	259.608	**7704.720**	**7704.990**	838.776
MM250n0.001_01	14358.490	14359.676	56.230	14354.470	14355.376	196.067	**14353.650**	**14354.220**	432.508	14359.390	14359.774	997.502
MM250n0.001_10	14446.360	14448.256	61.728	14443.260	14444.150	179.042	**14442.300**	**14443.120**	417.091	14448.260	14448.568	997.404
avg.value	4866.191	4866.762	19.590	4865.215	4865.534	70.716	**4864.941**	**4865.191**	173.684	4866.065	4866.223	485.506
avg.rank	3.500	3.650	–	2.500	2.800	–	**1.900**	**1.700**	–	2.100	1.850	–

Table 1 summarizes the results of the LAHC algorithms with different ρ values $\{1, 1000, 3000, 10000\}$. The bottom line gives the average value and average rank of each performance indicator. From this we observe that LAHC with a small ρ value (e.g., $\rho = 1$) quickly becomes trapped in a local optimum, leading to poor performance (with smallest average value and average rank). For large values of ρ (e.g., $\rho = 3000$), the search is less prone to becoming trapped but incurs the cost of a slower convergence speed. The solution quality can be poor if the computation time is not enough. To make a balance between the solution quality and computation time, a $\rho = 3000$ is suitable for LAHC, which achieves the smallest average values in terms of both $\hat{f}$ and $\bar{f}$. For the average rank, LAHC with $\rho = 3000$ also obtains the smallest rank value in terms of $\hat{f}$ and $\bar{f}$.

A.2. Sensitivity of ELAHC to ρ

Table 2 summarizes the results of the ELAHC algorithms with different ρ values $\{1,5,10,15\}$. The bottom line gives the average value and average rank of each performance indicator, disclosing that ELAHC with $\rho=1$ quickly converges to an unattractive local optimum, leading to poor performance (with smallest average value and average rank). For larger values of ρ (e.g., $\rho=5$), the search is less prone to becoming trapped but exhibits a slower convergence speed. The larger ρ value, the slower convergence speed. To make a balance between the solution quality and computation time, $\rho=5$ is suitable for ELAHC, achieving the smallest average values in terms of both $\hat{f}$ and $\bar{f}$. For the average rank, ELAHC with $\rho=5$ also obtains the second smallest rank value in terms of both $\hat{f}$ and $\bar{f}$.

Table 2 Comparisons among ELAHC Algorithms with Different ρ Values (i.e., $\rho\in\{1,5,10,15\}$)

	$\rho=1$			$\rho=5$			$\rho=10$			$\rho=15$		
Instance	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$
MM050n0.001_02	575.900	576.194	0.160	575.640	575.728	1.861	575.600	575.688	8.723	**575.540**	**575.670**	18.766
MM050n0.001_11	561.700	561.922	0.130	561.500	561.574	1.940	**561.480**	561.554	9.326	**561.480**	**561.518**	22.007
MM100n0.200_01	**416.000**	416.048	0.675	**416.000**	416.012	19.953	**416.000**	416.004	110.484	**416.000**	**416.000**	278.871
MM100n0.200_05	**411.720**	411.736	0.738	**411.720**	411.730	20.690	**411.720**	**411.724**	126.336	**411.720**	**411.724**	302.653
MM150n0.100_08	1249.800	1249.918	3.270	**1249.580**	1249.636	55.452	**1249.580**	**1249.612**	263.505	**1249.580**	1249.618	645.462
MM150n0.100_17	1266.110	1266.202	4.579	1266.110	1266.140	48.320	**1266.090**	**1266.110**	249.941	**1266.090**	**1266.110**	599.331
MM200n0.010_04	7670.010	7670.356	15.841	7667.750	7667.952	262.215	**7667.670**	**7667.776**	983.677	7669.250	7669.456	991.769
MM200n0.010_13	7706.240	7707.396	23.169	**7704.580**	7704.756	254.853	7704.600	**7704.698**	989.951	7706.120	7706.390	993.734
MM250n0.001_01	14358.130	14359.780	28.262	**14352.570**	**14353.268**	412.931	14353.970	14354.246	994.869	14358.850	14359.134	994.320
MM250n0.001_10	14447.120	14448.066	44.503	**14441.640**	**14442.230**	405.860	14442.880	14443.088	996.642	14447.480	14447.902	995.623
avg.value	4866.273	4866.762	12.133	**4864.709**	**4864.903**	148.408	4864.959	4865.050	473.345	4866.211	4866.352	584.254
avg.rank	3.450	4.000	–	2.150	2.400	–	**1.900**	**1.600**	–	2.500	2.000	–

ELAHC is an enhanced version of LAHC. Due to the use of a relaxed acceptance and replacement strategy, more non-improving solutions are accepted and their cost values are updated in the list. Therefore, ELAHC requires a smaller ρ value than LAHC, as confirmed by the comparison between results summarized in Tables 1 and 2.

A.3. Sensitivity of HER to ρ

As indicated in the description of HER, the ELAHC method is used to perform local optimization during the search. To investigate the effect of ρ on the performance of HER, we run HER with four different ρ values $\{1,5,10,15\}$. Detailed results in terms of the best result (f_{best}) and average result (f_{avg}) are presented in Figure 1. The x-axis indicates the ρ values, and y-axis shows the performance gaps. By treating HER with $\rho=1$ as a baseline, we calculate the performance gap as $(f-\tilde{f})/\tilde{f}$, where f is the result of the HER algorithm with $\rho\in\{1,5,10,15\}$ and $\tilde{f}$ is the result of the HER algorithm with $\rho=1$. A performance gap smaller than zero means a better result for the corresponding instance.

From Figure 1, we observe that HER with a small $\rho\in\{1,5\}$ value demonstrates a significantly better performance than HER with a large $\rho\in\{10,15\}$. Taking the MM050n0.001_02 instance as an example,

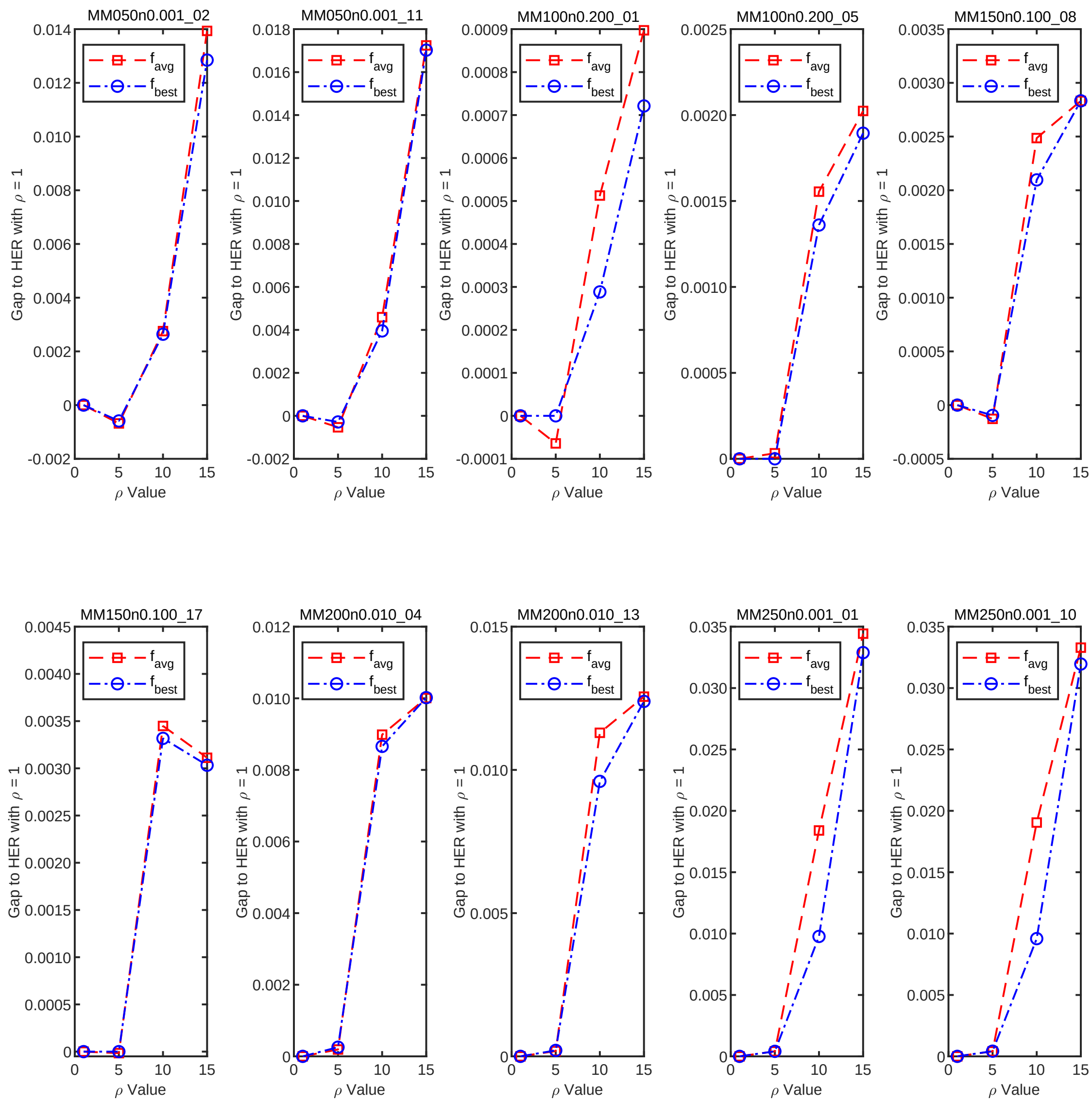

Figure 1 **Comparison among HER Algorithms with different ρ Values ($\rho \in \{1, 5, 10, 15\}$)**

we see that HER with $\rho = 5$ achieves the best performance in terms of both the best result and average result. Similar observations apply to the other nine instances. These results confirm that $\rho = 5$ is a suitable parameter value for ELAHC used in HER.

B. Comparison Between Iterated Local Search and its Improved Version

The original iterated local search (ILS) method for the rank aggregation problem is a multi-start local search algorithm based on the hill climbing (HC) algorithm, which demonstrates the best performance when the algorithms are allowed to perform a large number of fitness evaluations. Note that ILS employs HC to perform local optimization, which causes it to achieve a bad performance. To remedy this, we implemented an improved ILS method by replacing HC with the enhance late acceptance hill climbing (ELAHC) with $\rho = 1$. We conducted a detailed performance comparison between the original ILS and improved ILS on both synthetic and real-world instances with complete rankings, whose outcomes are as follows.

B.1. Results on Synthetic Instances

Table 3 summarizes the detailed comparative results on synthetic instances. For each algorithm, we report the best result ($\hat{f}$), the average result ($\bar{f}$), and the average time in seconds ($\bar{t}$). From Table 3, we observe that the improved ILS method performs better than the original method in all 20 cases in terms of both $\hat{f}$ and $\bar{f}$.

Table 3 Comparisons of the Original ILS and Improved ILS on Synthetic Instances

Instance		Original ILS (with HC)			Improved ILS (with ELAHC)		
θ	m	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$
0.200	050	337.900	357.308	3164.665	**183.140**	**187.913**	723.482
0.100	050	417.180	426.984	3607.289	**311.950**	**320.296**	30.213
0.010	050	571.910	577.889	4331.544	**550.970**	**559.797**	333.231
0.001	050	579.480	584.879	4090.002	**561.740**	**569.968**	1137.292
0.200	100	1590.090	1663.185	4357.084	**405.140**	**411.884**	389.789
0.100	100	1697.160	1756.552	3571.267	**776.020**	**787.745**	1190.498
0.010	100	2303.440	2320.391	3074.636	**2126.890**	**2153.107**	3401.334
0.001	100	2376.530	2393.986	3547.358	**2290.990**	**2306.096**	3662.618
0.200	150	3911.680	4072.344	4091.270	**629.410**	**636.948**	152.134
0.100	150	4028.860	4164.913	3813.916	**1245.730**	**1260.166**	2464.411
0.010	150	5123.720	5188.640	3637.212	**4533.430**	**4586.122**	2082.753
0.001	150	5406.240	5431.022	4491.537	**5144.480**	**5191.362**	3788.802
0.200	200	7491.710	7628.279	2548.153	**851.290**	**862.693**	115.287
0.100	200	7535.900	7714.423	3483.566	**1711.440**	**1734.231**	2286.683
0.010	200	9070.050	9185.279	3625.224	**7625.320**	**7698.460**	3830.479
0.001	200	9667.810	9706.857	3599.662	**9182.900**	**9240.152**	3768.522
0.200	250	11819.520	12354.986	3268.002	**1075.790**	**1087.684**	456.757
0.100	250	11888.110	12359.140	4139.456	**2186.420**	**2206.824**	3190.976
0.010	250	14002.130	14276.435	3707.153	**11182.830**	**11301.168**	3641.871
0.001	250	15180.200	15219.913	3593.352	**14337.260**	**14434.251**	3097.111

* The results of each combination of θ and m are averaged over 20 instances.

B.2. Results on Real-world Instances

Table 4 summarizes the detailed comparative results on real-world instances, leading to the same conclusion as in Table 3, that the improved ILS method significantly outperforms the original ILS method.

Table 4 Comparisons of the Original ILS and Improved ILS on Real-world Instances

	Original ILS (with HC)			Improved ILS (with ELAHC)		
Instance	$\hat{f}$	$\bar{f}$	$\bar{t}$	$\hat{f}$	$\bar{f}$	$\bar{t}$
Sushi	**19.181**	19.190	2772.938	**19.181**	**19.181**	1.003
F1	77.450	79.930	3714.395	**68.750**	**68.750**	0.197
Tour	4882.300	4913.240	4001.894	**3480.100**	**3480.800**	2938.547
ATPWomen_50	353.481	361.592	3531.377	**185.750**	**185.750**	4.264
ATPWomen_100	1693.558	1739.742	3072.500	**715.019**	**715.019**	954.638
ATPWomen_200	7743.269	7858.192	4309.973	**2823.423**	**2823.596**	3467.266
ATPMen_50	346.269	364.565	2777.543	**197.962**	**197.962**	7.795
ATPMen_100	1746.250	1779.123	4194.137	**862.365**	**862.365**	1947.933
ATPMen_200	7742.212	7899.108	4777.542	**2810.519**	**2810.823**	3543.888