Ant Colony Optimization Algorithms for Dynamic Optimization: A Case Study of the Dynamic Travelling Salesperson Problem – Supplementary Material

Michalis Mavrovouniotis, Shengxiang Yang, Mien Van, Changhe Li, and Marios Polycarpou

IV. Experimental Setup

A. Dynamic Test Cases

TSP instances were obtained from the TSPLIB benchmark library [1], which is available at https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/, to generate dynamic test cases as described in Section II of the paper. Specifically, the frequency of change was set proportionally to the size of the problem instance as follows: $f = 2.5n$ and $f = 25n$, indicating quickly (e.g., before the algorithm converges, denoted as `fast`) and slowly (e.g., after the algorithm has converged, denoted as `slow`) changing environments, respectively. Note that the resulting f value is rounded up (if needed), so that the dynamic change will occur at the start or the end of the algorithmic iteration. The magnitude of change was set to $m = 0.1$, $m = 0.25$, $m = 0.5$, and $m = 0.75$, indicating small, to medium, to large dynamic changes, respectively. The dynamic settings for each DTSP test case are selected to systematically analyze the dynamic behavior of ACO algorithms (i.e., their ability to recover fast and produce the best output). Note that usually as the frequency of change is faster and the magnitude of change is increasing the DTSP test case becomes harder to address [2], [3].

B. Parameter Settings

The common parameters of all ACO algorithms used were set to typical values (i.e., $\alpha = 1$ and $\beta = 5$) for all the experiments. The colony size ω for each framework was investigated for the two types of DTSPs separately with values $\omega = \{50, 25, 10, 5\}$. In addition, the key parameter of the evaporation-based framework variants (i.e., the evaporation rate ρ) was investigated with values $\rho = \{0.1, 0.2, 0.5, 0.8\}$ and the key parameter of the population-based framework variants (i.e., the population-list size K) was investigated with values $K = \{2, 3, 5, 10\}$. The replacement ratio r_i of the generated immigrants for RIACO, EIACO, HIACO, HIACO-II, MIACO and EIIACO was investigated with values $r_i = \{0.1, 0.5, 0.8\}$. For $\mathcal{MM}$AS$_S$ the number of discrete rate values available to the self-adaptive evaporation mechanism was investigated with values ranging from 5 to 50. For $\mathcal{MM}$AS$_{caste}$, one caste uses the random proportional decision rule while the other uses the pseudorandom proportional decision rule, and MC-$\mathcal{MM}$AS uses two independent colonies.

The combination of these parameters that were found to yield reasonable performance is $\omega = 5$ for most DTSPs with node changes and $\omega = 25$ for most DTSPs with weight changes for all ACO algorithms. Furthermore, for both DTSPs with node and weight changes the remaining parameters are $\rho = 0.8$, $K = 3$, $r_i = 0.5$ and 20 discrete rate values for ACO algorithms using these parameters.

For each ACO algorithm on each DTSP test case, 30 independent runs were executed on the same set of random seed numbers. For each run, 100 environmental changes were allowed and the value of the best-so-far ant since the last change of the environment was recorded.

Michalis Mavrovouniotis, KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, CYPRUS

Shengxiang Yang, School of Computer Science and Informatics, De Montfort University, Leicester, UK

Mien Van, School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK

Changhe Li, School of Automation and Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, China University of Geosciences, Wuhan, CHINA

Marios Polycarpou, KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, CYPRUS

V. Experimental Results and Their Analysis

A. Comparison Between Evaporation-Based and Population-Based Frameworks

TABLE III: Experimental results regarding $\bar{P}_{offline}$, $\bar{P}_{change}$, and $\bar{P}_{robust}$ (averaged over 30 runs) of evaporation-based and population-based frameworks for DTSPs.

Metric	ACO Framework	`kroA200`				`rd400`				`u1060`			
DTSPs with Weight Changes													
	`fast`, $m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\bar{P}_{offline}$	Evaporation	**29285**	**30140**	**30938**	31420	15798	16361	16664	16752	251389	254703	257307	258630
	Population	29712	30552	31064	**31240**	15801	**16276**	**16541**	**16609**	**250003**	**252955**	**255351**	**256632**
$\bar{P}_{change}$	Evaporation	**29033**	**29620**	**30142**	30479	15609	15992	16177	16209	247975	249219	250328	250979
	Population	29479	30101	30419	30535	15644	15983	**16151**	16195	**246670**	**247801**	**249051**	**249767**
$\bar{P}_{robust}$	Evaporation	0.98	0.95	0.93	0.92	0.97	0.93	0.91	0.89	0.96	0.92	0.89	0.88
	Population	0.98	**0.96**	**0.94**	**0.93**	0.97	**0.95**	**0.92**	**0.92**	0.96	0.92	**0.90**	**0.89**
	`slow`, $m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\bar{P}_{offline}$	Evaporation	**28075**	**28300**	**28509**	**28821**	**14600**	**14947**	**15234**	**15309**	240124	241128	242023	242629
	Population	28400	28745	29052	29304	14803	15154	15393	15477	**238941**	**239814**	**240815**	**241573**
$\bar{P}_{change}$	Evaporation	**27897**	**27964**	**27954**	**28116**	**14487**	**14641**	**14729**	**14758**	237205	236858	236861	237100
	Population	28200	28363	28450	28596	14678	14870	14974	15014	**235937**	**235303**	**235368**	**235786**
$\bar{P}_{robust}$	Evaporation	0.97	0.94	0.90	0.88	0.97	0.93	0.88	0.85	0.95	0.91	0.86	0.84
	Population	0.97	0.94	**0.91**	**0.89**	0.97	0.93	**0.89**	**0.87**	0.95	0.91	**0.87**	**0.85**
DTSPs with Node Changes													
	`fast`, $m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\bar{P}_{offline}$	Evaporation	33923	34620	34751	34787	17208	17322	17414	17391	321956	325217	326427	326616
	Population	**33787**	**34430**	**34373**	**34189**	**17116**	**17203**	**17186**	**17099**	**319221**	**321851**	**321462**	**320080**
$\bar{P}_{change}$	Evaporation	**32599**	**33021**	**33032**	33093	16598	16602	16657	16650	313117	314794	315653	315710
	Population	32651	33118	33091	33000	**16571**	16594	**16613**	**16576**	**310786**	**312261**	**312616**	**312250**
$\bar{P}_{robust}$	Evaporation	0.80	0.77	0.75	0.76	0.79	0.76	0.76	0.76	0.79	0.76	0.76	0.77
	Population	**0.84**	**0.82**	**0.83**	**0.84**	**0.83**	**0.82**	**0.83**	**0.85**	**0.83**	**0.82**	**0.84**	**0.86**
	`slow`, $m \Rightarrow$	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
$\bar{P}_{offline}$	Evaporation	**31186**	**31635**	**31621**	**31654**	**15897**	**15892**	**15952**	**15933**	305124	306628	307077	307743
	Population	31416	31887	31855	31782	15943	15952	15982	15957	**303528**	**304820**	**305068**	**304999**
$\bar{P}_{change}$	Evaporation	**30498**	**30771**	**30726**	**30736**	**15469**	**15403**	**15446**	**15416**	300109	300317	300536	301261
	Population	30796	31113	31053	30993	15575	15527	15554	15533	**298819**	**299506**	**299983**	**299952**
$\bar{P}_{robust}$	Evaporation	0.76	0.72	0.70	0.70	0.75	0.71	0.70	0.70	0.74	0.73	0.73	0.74
	Population	**0.80**	**0.78**	**0.78**	**0.79**	**0.79**	**0.77**	**0.78**	**0.79**	**0.80**	**0.79**	**0.80**	**0.82**

Bold values indicate statistical significance

B. Effect of Main Framework Features

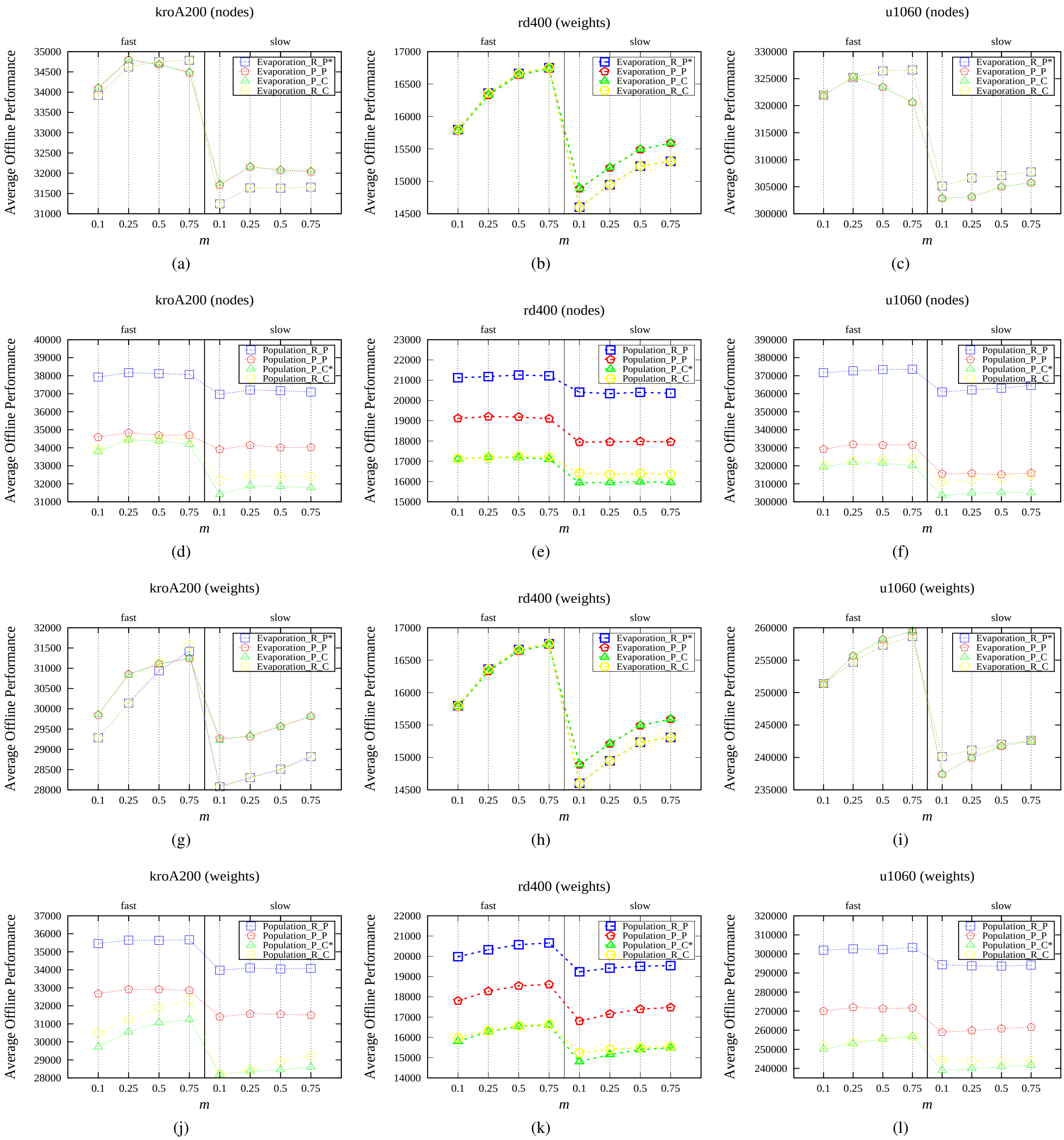

Fig. 3: Averaged (over 30 runs) offline performance for evaporation-based (a)(b)(c)(g)(h)(i) and population-based (d)(e)(f)(j)(k)(l) frameworks with alternative decision rules and pheromone update policies for different DTSPs. *These combinations are the default ones.

C. *Effect of Dynamic Strategies*

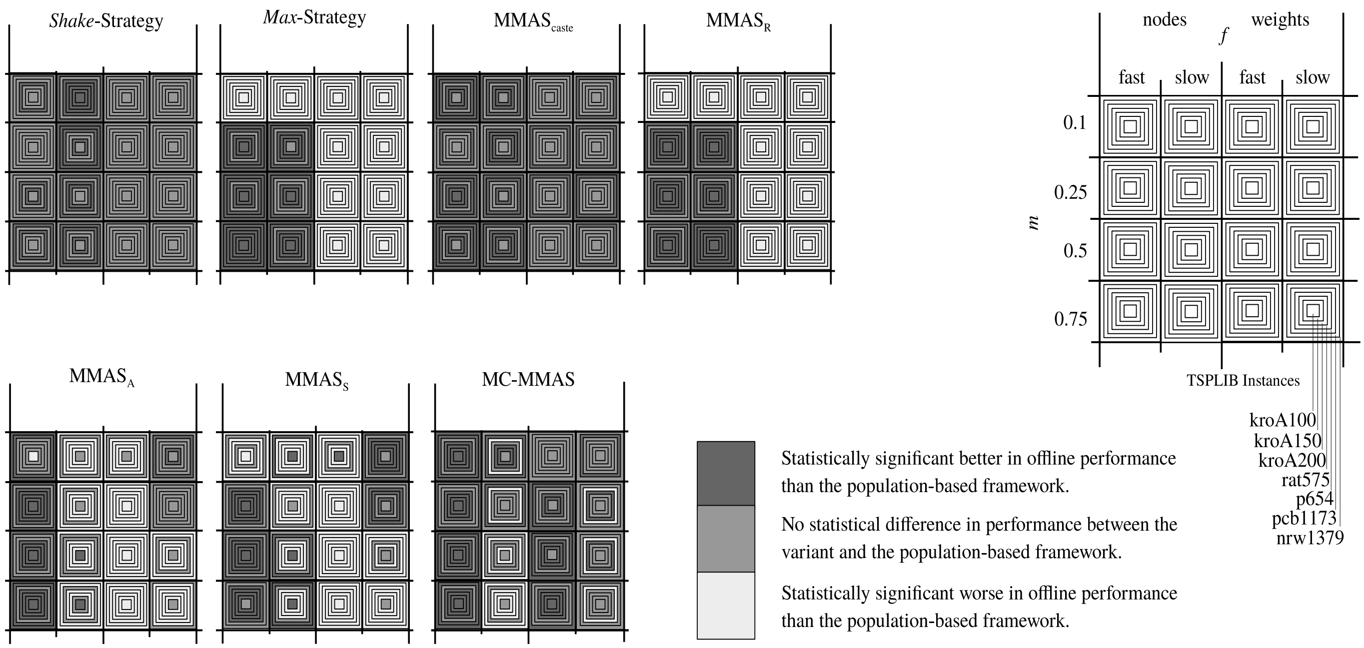

Fig. 4: Each square represents the comparisons of the statistical tests of the aforementioned ACO variant against the evaporation-based framework. Each square is subdivided into sixteen smaller squares that represent the dynamic settings of the DTSP. The squares are grouped by the type of change. Each smaller square contains a stack of increasingly larger boxes that represents a set of increasingly larger problem instances.

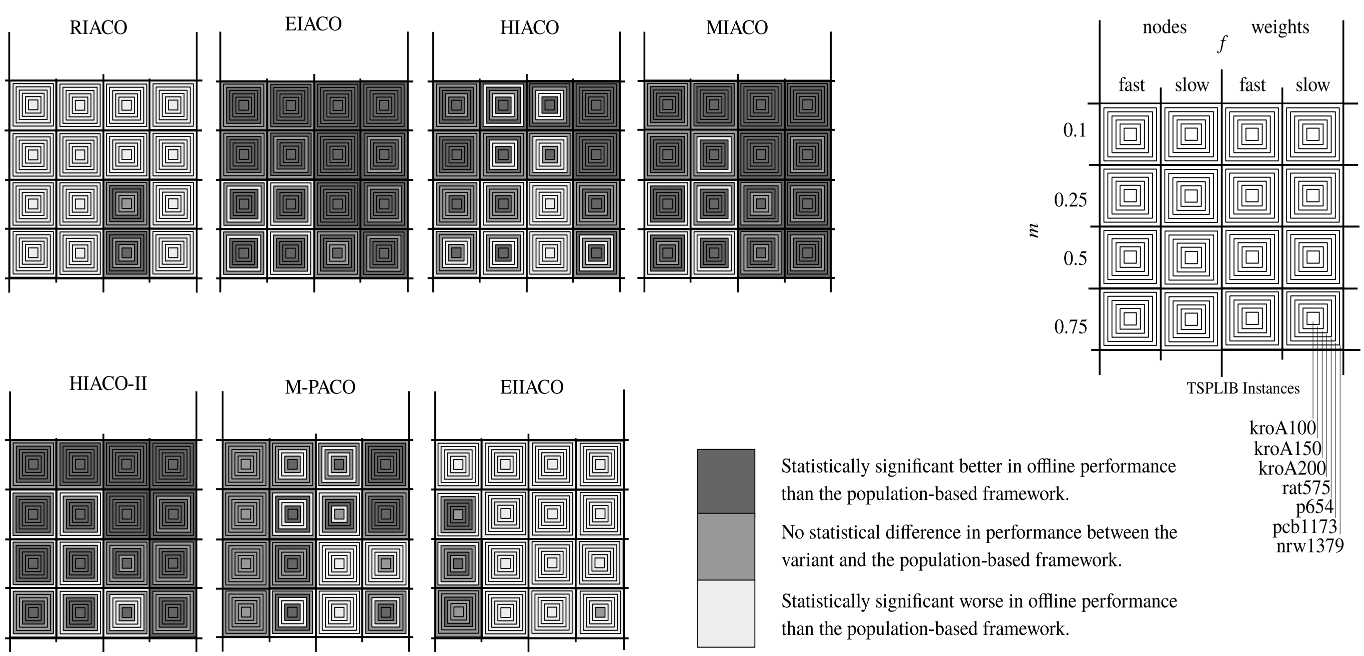

Fig. 5: Each square represents the comparisons of the statistical tests of the aforementioned ACO variant against the population-based framework. Each square is subdivided into sixteen smaller squares that represent the dynamic settings of the DTSP. The squares are grouped by the type of change. Each smaller square contains a stack of increasingly larger boxes that represents a set of increasingly larger problem instances.

D. *Effect of Utilizing Change-Related Information*

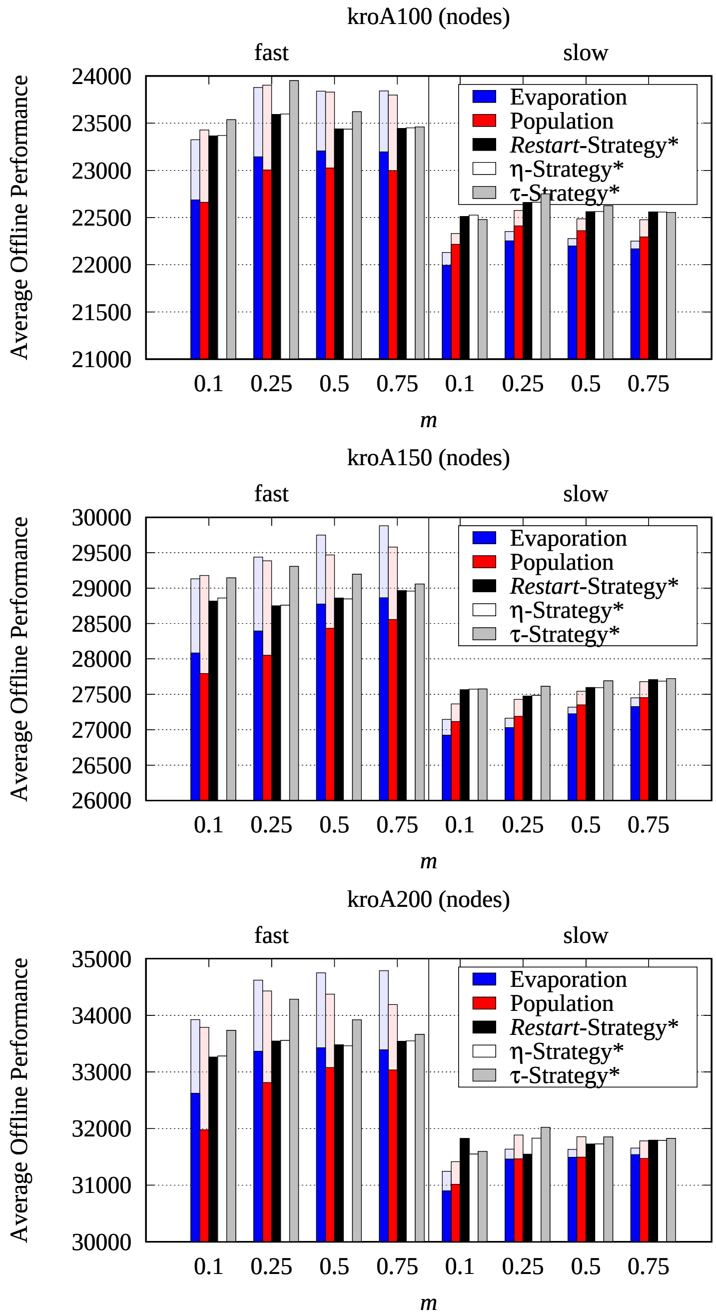

Fig. 6: [Note that Fig. 6 here, is Fig. 4 on the paper] $\bar{P}_{offline}$ (averaged over 30 runs) results of evaporation-based and population-based frameworks, and three evaporation-based variants when utilizing change-related information for DTSPs with node changes. Each bar is divided into two parts that represent the results when utilizing change-related information (darker) or not (lighter). *These strategies have been designed to utilize change-related information and, thus, the values when information is not utilized do not exist.

E. Comparisons with Evolutionary Algorithms

TABLE IV: Experimental results regarding the $\bar{P}_{offline}$ (averaged over 30 runs) of ACO algorithms with state-of-the-art evolutionary algorithms for DTSPs with weight changes with $f = n \cdot 100$ and m randomly chosen from a uniform distribution in $(0.0, 0.5]$.

TSPLIB Instance	P-ACO	$\mathcal{MM}$AS	EIGA	GPX	EIACO	MC-$\mathcal{MM}$AS
`berlin52`	7261	7195	7414	7392	**7177**	<u>7191</u>
`eil101`	572	<u>568</u>	581	578	<u>569</u>	**567**
`kroB200`	28481	<u>28261</u>	29161	29026	**28231**	<u>28345</u>
`lin318`	40154	<u>39957</u>	41543	41054	40456	**39932**
`pr439`	105376	104591	105904	106193	104633	**103918**
`p654`	49138	49415	48178	**47921**	49127	49533
`rat783`	**8434**	8521	8515	8509	<u>8436</u>	<u>8444</u>
`pr1002`	270701	274370	281952	279321	**268532**	274301
`u1432`	158822	160881	163427	161203	**157503**	161244
DTSP with Node Changes						
`berlin52`	8080	8046	8749	8700	**8004**	8034
`eil101`	560	<u>558</u>	599	596	**555**	<u>557</u>
`kroB200`	31465	31245	33890	33632	**31095**	31278
`lin318`	47854	**47496**	49434	49345	<u>47593</u>	<u>47499</u>
`pr439`	159001	**156316**	167630	167557	158538	157975
`p654`	66474	**66075**	69902	69834	66712	66338
`rat783`	8837	<u>8634</u>	9234	9237	**8609**	8755
`pr1002`	**308512**	309991	322655	320925	310153	309137
`u1432`	**157658**	157875	178913	176435	158894	157907

Bold values indicate statistical significance
<u>Underline</u> values indicate no statistical difference with the bold value

References

[1] G. Reinelt, "TSPLIB–A traveling salesman problem library," *ORSA J. Comput.*, vol. 3, no. 4, pp. 376–384, Nov. 1991.
[2] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors," *Appl. Soft Comput.*, vol. 13, no. 10, pp. 4023–4037, Oct. 2013.
[3] M. Mavrovouniotis, S. Yang, and X. Yao, "A benchmark generator for dynamic permutation-encoded problems," in *Parallel Problem Solving from Nature, PPSN XII*, LNCS, vol. 7492, C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, Eds. Berlin, Heidelberg: Springer, 2012, pp. 508–517.