Artificial Intelligence **2023/2024**

Student: LUCA PERNIGO StudentID:19-993-658

Assignment 1 Due date: 18 October 2023, 23:59

Iterated Local Search for Travelling Salesman Problem

1. Implement Iterated Local Search

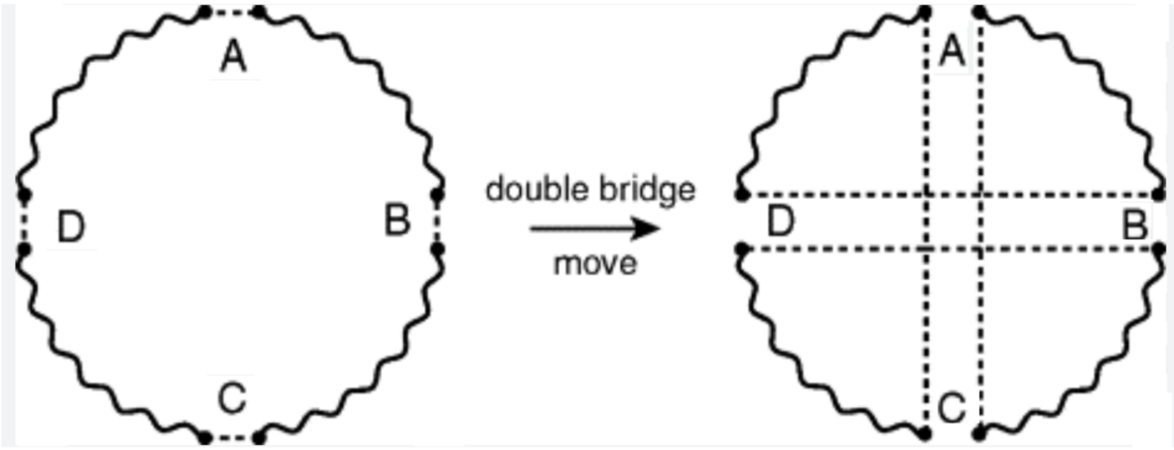
To get started the three acceptance criteria(Better, Random Walk, Large Step Markov Chain) were implemented. The idea of each of those criteria is reported hereafter.

- Better: accept candidate solution if and only if it is better than the current best.
- Random Walk: always accept the lastly sampled solution
- Large Step Markov Chain: always accept the candidate solution if it is better than the current best, otherwise if it is worse accept it with probability $exp\{\frac{C(s^*)-C(s^{*'}))}{T}\}$, where C is a cost function, s^* is the current best solution, $s^{*'}$ is the candidate solution and T is the temperature parameter which gets cooled by a parameter α as the algorithm iterates over its for loop.

2. Implementation & Evaluation

My double bridge implementation involves the following blocks:

- sample randomly four numbers without replacement in the range $[1, \#nodes]$, and then sort them. These numbers represent the edges that will be removed
- compute the new solution by building it from the solution passed as input. To better understand the provided implementation, consider the following picture that gives a visual representation of what the code is doing (the clockwise convention has been used with A the first removed edge and D the last edge that is removed).



Essentially, applying double bridge to the input solution consists of changing the path of the input through indexing.

- Finally the new cost is computed by removing from the old cost the cost of the removed edges and by adding the cost of the inserted edges; in this way there is no need to recompute the entire cost from scratch, this trick results in an increase in performance.

After having implemented the double bridge perturbation and the acceptance criteria, different combinations of temperatures, cooling parameters and local search algorithms have been tested on each of the assigned instances.
Following are reported for each problem, the pairs of temperatures and alphas that performed relatively better between the set of all conducted experiments. Notice how the chosen initial value of the temperature changes depending on the size of the problem; this follows from the fact that smaller problems carry out more iterations in three minutes while bigger problem perform less iterations in the same time window.

2.1. d198

Temperature=100000 $\alpha = 0.90$

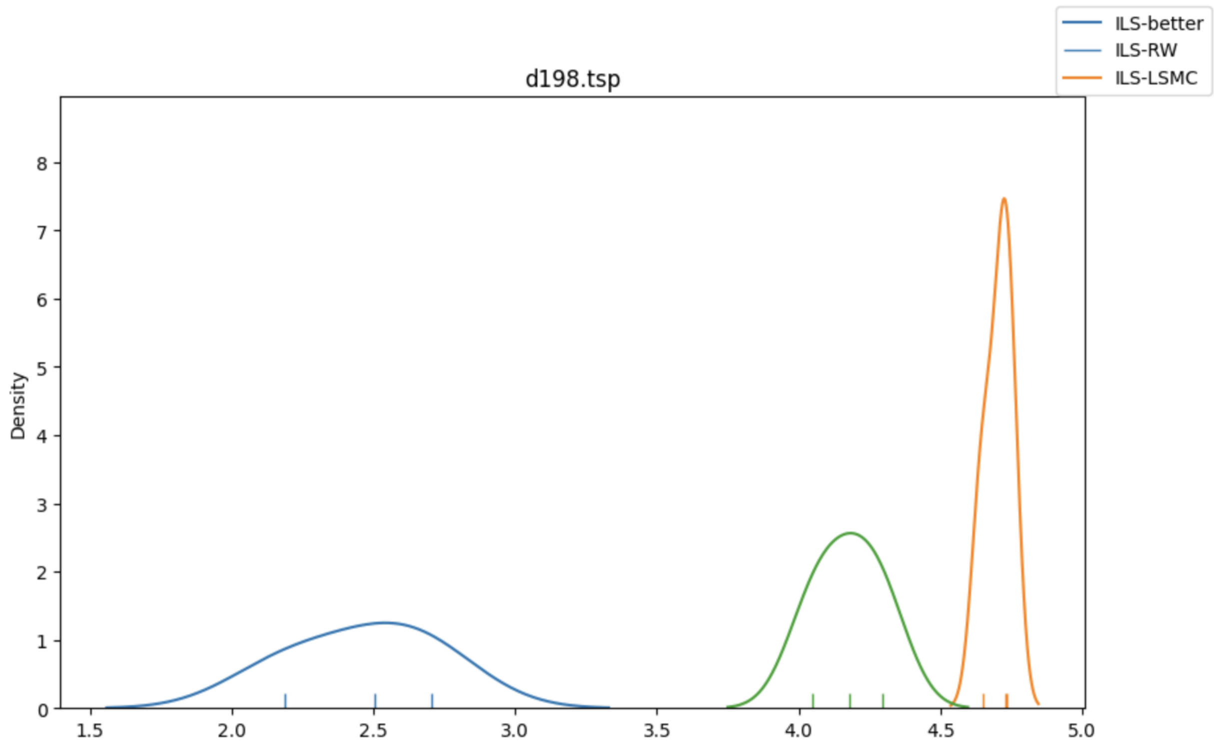
				tour length	gap	time to solve	calls Local Search
problem	**optimal length**	**method**	**seed**				
d198.tsp	**15780.0**	**initialized with random, improved with ILS-better**	**0**	15880.0	0.63	180.033	630
		initialized with random, improved with ILS-RW	**0**	16782.0	6.35	180.006	608
		initialized with random, improved with ILS-LSMC	**0**	15875.0	0.60	180.018	670
		initialized with random, improved with ILS-better	**123**	15987.0	1.31	180.122	696
		initialized with random, improved with ILS-RW	**123**	16931.0	7.29	180.219	637
		initialized with random, improved with ILS-LSMC	**123**	15910.0	0.82	180.039	670
		initialized with random, improved with ILS-better	**333**	15861.0	0.51	180.037	680
		initialized with random, improved with ILS-RW	**333**	16680.0	5.70	180.053	638
		initialized with random, improved with ILS-LSMC	**333**	15856.0	0.48	180.131	662

Temperature=1000000 $\alpha = 0.96$

problem	optimal length	method	seed	tour length	gap	time to solve	calls Local Search
d198.tsp	15780.0	initialized with random, improved with ILS-better	0	15852.0	0.46	180.205	686
		initialized with random, improved with ILS-RW	0	16627.0	5.37	180.233	645
		initialized with random, improved with ILS-LSMC	0	15962.0	1.15	180.090	644
		initialized with random, improved with ILS-better	123	15987.0	1.31	180.086	695
		initialized with random, improved with ILS-RW	123	16319.0	3.42	180.009	632
		initialized with random, improved with ILS-LSMC	123	15949.0	1.07	180.244	664
		initialized with random, improved with ILS-better	333	15861.0	0.51	180.051	674
		initialized with random, improved with ILS-RW	333	16656.0	5.55	180.149	636
		initialized with random, improved with ILS-LSMC	333	15890.0	0.70	180.342	663

See that the results in the first table are slightly better than the second one, this may suggest that between the two, the first choice of parameters is more suitable for the d198.tsp problem.
Observing table 1 we can see that LSMC on average outperformed the Better acceptance criterion in terms of gap. Additionally, notice that the Random criterion performed really bad in finding an optimal solution.
Hereafter is a visualization for the results in table 1.

Temperature=100000 $\alpha = 0.90$



This kind of chart visualizes the distribution of the average between the set of solutions that were selected as new best for each acceptance criteria. We can notice that the Better criterion is more moved to the left while the Random and the LSMC are more moved to the right; this follows from the fact that, with the Better criterion a solution selected to be the new best is always strictly smaller than the previous best. On the other hand, with Random and LSMC it can happen to accept as new best a solution that is worse than the current best; thus, this is why the orange and green curve are moved right.

2.2. pr439

Next the same kind of analysis was applied to the pr439.tsp problem.

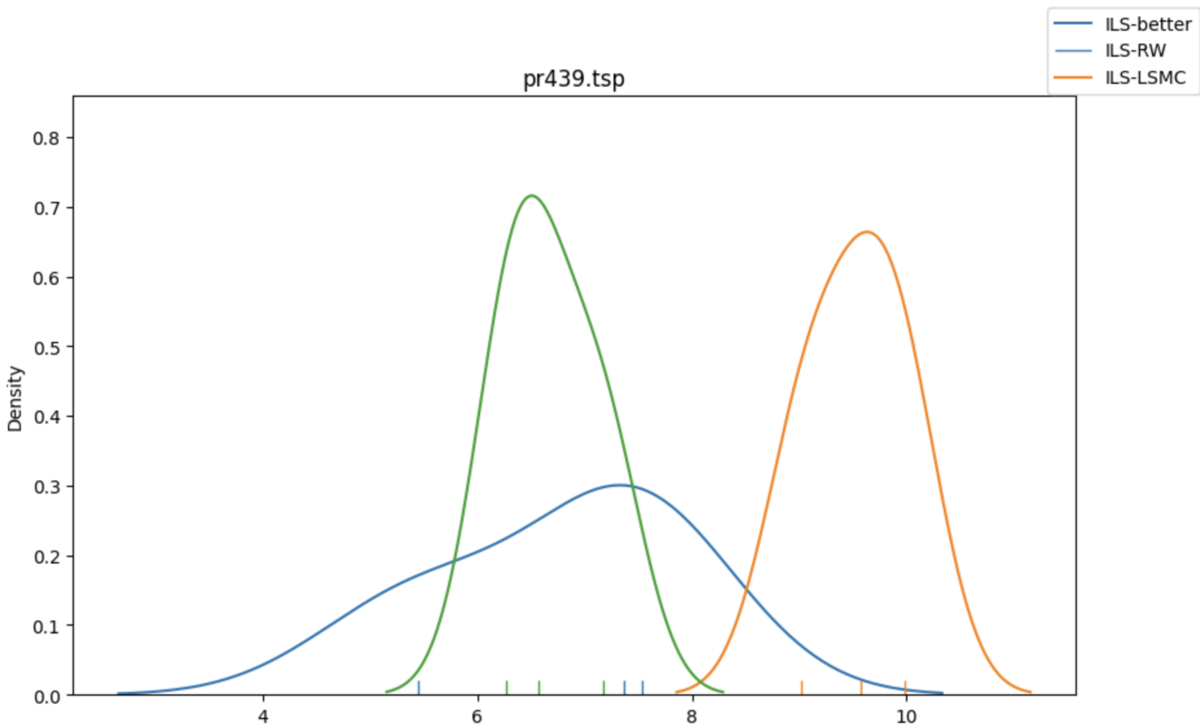
Temperature=1000 $\alpha = 0.85$

problem	optimal length	method	seed	tour length	gap	time to solve	calls Local Search
pr439.tsp	**107217.0**	**initialized with random, improved with ILS-better**	**0**	109170.0	1.82	181.069	84
		initialized with random, improved with ILS-RW	**0**	115545.0	7.77	181.217	82
		initialized with random, improved with ILS-LSMC	**0**	109358.0	2.00	181.068	87
		initialized with random, improved with ILS-better	**123**	111524.0	4.02	181.372	85
		initialized with random, improved with ILS-RW	**123**	115681.0	7.89	180.871	83
		initialized with random, improved with ILS-LSMC	**123**	108935.0	1.60	180.516	87
		initialized with random, improved with ILS-better	**333**	111642.0	4.13	180.431	85
		initialized with random, improved with ILS-RW	**333**	112265.0	4.71	181.525	85
		initialized with random, improved with ILS-LSMC	**333**	110990.0	3.52	180.140	87

Temperature=100000 $\alpha = 0.90$

problem	optimal length	method	seed	tour length	gap	time to solve	calls Local Search
pr439.tsp	**107217.0**	**initialized with random, improved with ILS-better**	**0**	111222.0	3.74	181.556	83
		initialized with random, improved with ILS-RW	**0**	115228.0	7.47	181.329	84
		initialized with random, improved with ILS-LSMC	**0**	109208.0	1.86	181.796	92
		initialized with random, improved with ILS-better	**123**	111524.0	4.02	180.397	84
		initialized with random, improved with ILS-RW	**123**	115681.0	7.89	180.603	83
		initialized with random, improved with ILS-LSMC	**123**	110766.0	3.31	181.322	82
		initialized with random, improved with ILS-better	**333**	111642.0	4.13	180.559	85
		initialized with random, improved with ILS-RW	**333**	112265.0	4.71	181.862	85
		initialized with random, improved with ILS-LSMC	**333**	111914.0	4.38	181.266	93

Temperature=1000 $\alpha = 0.85$



Again the Random criterion was the worst between the threes. LSMC and Better performed similarly with LSMC achieving lower gaps in two seeds out of three in both experiments.

2.3. u1060

Finally ILS was executed on the u1060.tsp problem.

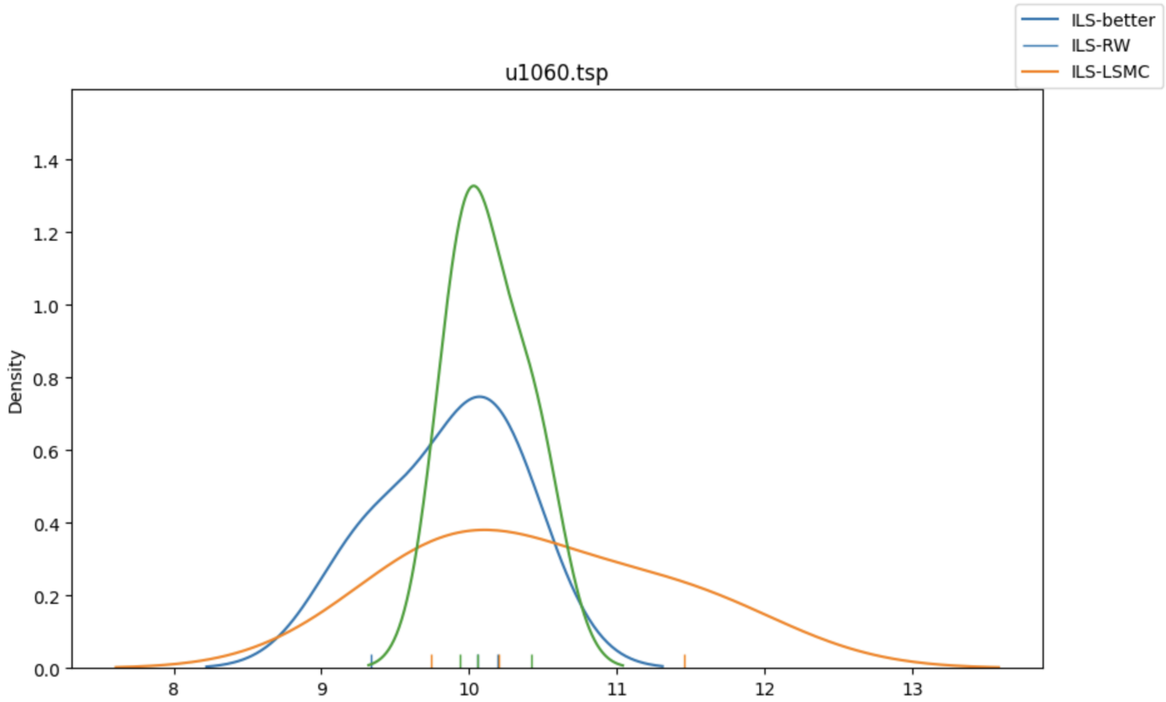
Temperature=100 $\alpha = 0.85$

problem	optimal length	method	seed	tour length	gap	time to solve	calls Local Search
u1060.tsp	224094.0	initialized with random, improved with ILS-better	0	242711.0	8.31	195.206	7
		initialized with random, improved with ILS-RW	0	244538.0	9.12	181.729	5
		initialized with random, improved with ILS-LSMC	0	241295.0	7.68	183.567	7
		initialized with random, improved with ILS-better	123	241678.0	7.85	191.254	5
		initialized with random, improved with ILS-RW	123	248652.0	10.96	182.650	4
		initialized with random, improved with ILS-LSMC	123	246757.0	10.11	187.725	4
		initialized with random, improved with ILS-better	333	245312.0	9.47	187.680	6
		initialized with random, improved with ILS-RW	333	243604.0	8.71	187.421	6
		initialized with random, improved with ILS-LSMC	333	245312.0	9.47	188.447	6

Temperature=1000 $\alpha = 0.85$

problem	optimal length	method	seed	tour length	gap	time to solve	calls Local Search
u1060.tsp	224094.0	initialized with random, improved with ILS-better	0	242888.0	8.39	181.073	6
		initialized with random, improved with ILS-RW	0	243989.0	8.88	193.800	7
		initialized with random, improved with ILS-LSMC	0	245581.0	9.59	180.533	6
		initialized with random, improved with ILS-better	123	241678.0	7.85	193.283	5
		initialized with random, improved with ILS-RW	123	248652.0	10.96	184.528	4
		initialized with random, improved with ILS-LSMC	123	246757.0	10.11	190.904	4
		initialized with random, improved with ILS-better	333	245312.0	9.47	192.186	6
		initialized with random, improved with ILS-RW	333	243604.0	8.71	194.383	6
		initialized with random, improved with ILS-LSMC	333	245312.0	9.47	195.091	6

Temperature=100 $\alpha = 0.85$



For the biggest problem, there is no clear winner between the criteria. Additionally, there is no superior set of parameters between those tried. This comes from the fact that in this setting the number of iterations performed in three minutes is really slow.
In order to address this problem, the cython class two_opt_with_candidate was tried first with a candidate list of 20 neighbours and then with a list of 100 neighbours.

Temperature=100 $\alpha = 0.85$ two_opt_with_candidate 20 neighbours

				tour length	gap	time to solve	calls Local Search
problem	**optimal length**	**method**	**seed**				
u1060.tsp	**224094.0**	**initialized with random, improved with ILS-better**	**0**	367198.0	63.86	180.803	151
		initialized with random, improved with ILS-RW	**0**	303532.0	35.45	181.343	146
		initialized with random, improved with ILS-LSMC	**0**	294279.0	31.32	180.305	142
		initialized with random, improved with ILS-better	**123**	306473.0	36.76	180.818	142
		initialized with random, improved with ILS-RW	**123**	269330.0	20.19	181.266	141
		initialized with random, improved with ILS-LSMC	**123**	301273.0	34.44	180.225	139
		initialized with random, improved with ILS-better	**333**	325568.0	45.28	180.222	142
		initialized with random, improved with ILS-RW	**333**	286602.0	27.89	181.286	139
		initialized with random, improved with ILS-LSMC	**333**	318751.0	42.24	180.554	134

Temperature=1000 $\alpha = 0.85$ two_opt_with_candidate 100 neighbours

				tour length	gap	time to solve	calls Local Search
problem	**optimal length**	**method**	**seed**				
u1060.tsp	**224094.0**	**initialized with random, improved with ILS-better**	**0**	275885.0	23.11	181.070	34
		initialized with random, improved with ILS-RW	**0**	284419.0	26.92	180.046	38
		initialized with random, improved with ILS-LSMC	**0**	276787.0	23.51	184.324	37
		initialized with random, improved with ILS-better	**123**	256787.0	14.59	184.649	34
		initialized with random, improved with ILS-RW	**123**	256787.0	14.59	182.949	34
		initialized with random, improved with ILS-LSMC	**123**	264895.0	18.21	180.292	37
		initialized with random, improved with ILS-better	**333**	274353.0	22.43	181.186	35
		initialized with random, improved with ILS-RW	**333**	274362.0	22.43	184.846	36
		initialized with random, improved with ILS-LSMC	**333**	269676.0	20.34	183.581	37

As we can see from the above tables, there was a gain in terms of iterations. However, there was also a significant increase in the gap from the optimal solution.
In addition, notice the trade off we experience by making the pool of candidate nodes bigger; we have a decrease in the gap but also the number of calls to local search decreases.

To conclude, with small sized problems iterated local search converges in a reasonable amount of time. On the contrary for bigger problems it might take a bit before a meaningful approximate solution is attained.
Finally, the Random criterion performs really poorly while the results obtain by Better and LSMC are good enough.