Artificial Intelligence **2023/2024**

Student: LUCA PERNIGO StudentID:19-993-658

Assignment 3

Due date: 6 December 2023, 23:59

Ant Colony System for Travelling Salesman Problem

Goal of this project is implementing the ACS for solving the TSP problems; that is finding the global minimum of TSP instances.

1. Target Problems

The problems that have been considered in this study are the **eil76**, **ch130** and the **d198**.

2. Methods Applied

The ACS has been implemented and tested with the following four variants:

- Basic Ant Colony System (ACS)
- ACS enhanced with 2-opt
- ACS with 2-opt and candidate list
- ACS with 2-opt and Popmusic candidate list

3. Parameter Configuration

The ACS has been configured using the following settings:

- Number of ants (m): 10
- Importance of trail (β): 2
- Trail persistence (α) and pheromone evaporation rate (ρ): 0.1
- Initial pheromone level (τ_0): $(n \cdot Lnn)^{-1}$
- Probability (q_0) values: 0.5, 0.98, $(1 - \frac{13}{n})$

- Candidate List (cl):
  - Candidate list with a fixed size of 15
  - Popmusic candidate list
- Global updating rule: Global best approach
- 2-opt applied to the the best ant of each iteration

4. Implementation

4.1. get_pheromone

Takes as input ant number k and returns the pheromone levels for the feasible neighborhood of the node where ant k is currently standing.

4.2. get_eta

Takes as input ant number k and returns the eta between the node r_k and every node belonging to $J_k(r_k)$. r_k is the current position of ant k while $J_k(r_k)$ is the neighborhood of feasible nodes of r_k.

4.3. initialize

This method initializes the variables storing the tour, the position and the neighborhood of each ant. Furthermore, this function randomly positions each ant on one node; starting nodes are unique to ants.

4.4. transition

It implements the transition rule. With probability q_0 we transition to the node with highest pheromone, otherwise we sample randomly the next move from the space of feasible neighbours excluding the best move.

4.5. local

It calls the transition function and implements the local pheromone update mechanism of ACS.

4.6. best

If there is an ant that during the last iteration achieved a better solution than the global one, we write the former solution as the new global one. In addition, we update the variable storing the best length ever achieved accordingly.

4.7. best variations

There are three additional variations of **best**: **best_twoopt**, **best_twoopt_cl** and **best_twoopt_popmusic_cl**. The idea here is to apply two opt in order to improve the iteration best solution and then check whether the variable storing the global best solution has to be updated. **best_twoopt** applies the classical twoopt algorithm.
best_twoopt_cl applies the twoopt algorithm with the aid of a candidate list, where the candidate nodes are selected according to the less distant criteria.
best_twoopt_popmusic_cl applies the twoopt algorithm with the aid of the candidate list specified by the popmusic criteria. Note that for the eil76 problem instance, the popmusic candidate list was not available, therefore ACS+popmusic has not been applied to the instance eil76.

5. global

This method updates the pheromone level by applying the global update rule of ACS.

6. solve

While the given time has not expired and we have not found the best solution, we iterate the steps of ACS. The routine is: **initialize** → **local** → **best_*** → **global**.

7. Output requirements

7.1. Choice of q_0

A couple of experiments were run with different q_0. As expected $q_0 = 0.50$ performed badly, as outlined by the following picture. In Figure 1 the ACS algorithm has been run for five seconds testing for the proposed q_0.

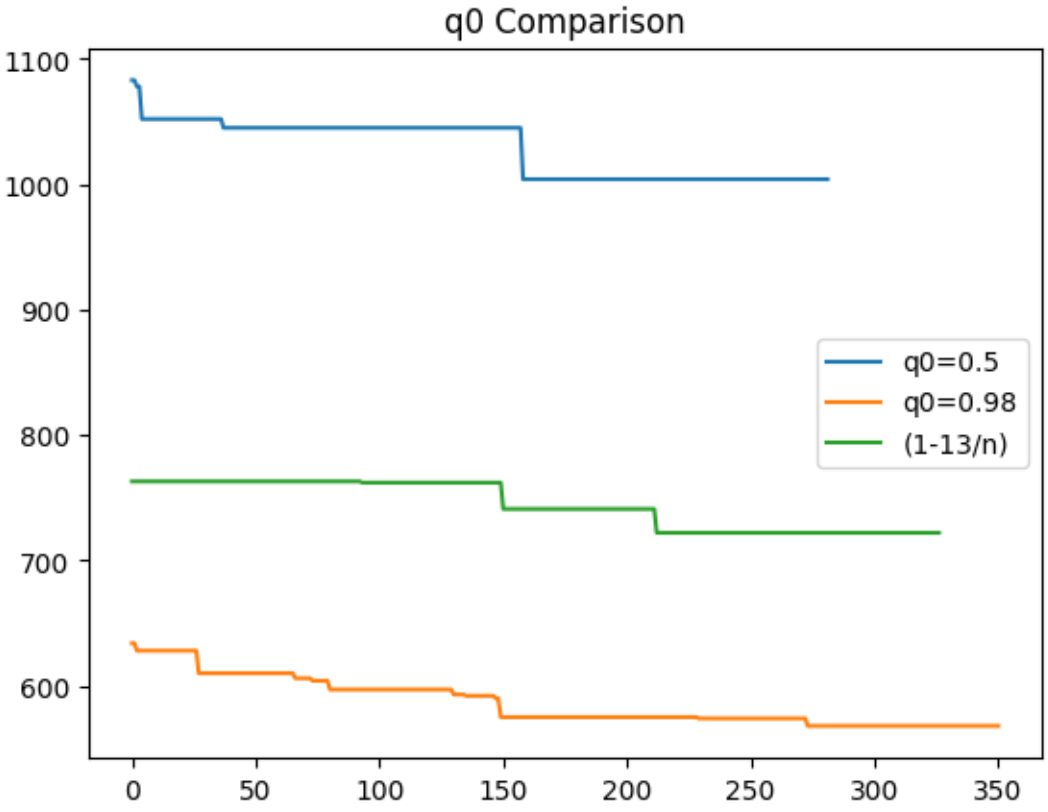

Figure 1

Given this, from now on all the experiments have been carried out using $q_0 = 0.98$.

7.2. Results

In Table 1 are reported the results of various runs of the algorithm. Every pair of method and instance has been run three times with different seeds.
Termination conditions for the runs were convergence and exceeding the time limit. Furthermore, note that the results that follow hereafter were obtained by setting a time limit of three minutes.
Tours Generated is how many tours were generated until termination condition was met.
Best Cost is the best cost ever achieved during the run.
Best Gap% is the best gap ever achieved during the run. Note that it is expressed in percentages.

Table 1

	Method	Instance	Seed	# Tours Generated	Best Cost	Best Gap%
0	ACS	eil76.tsp	4	128000	554.000000	2.973978
1	ACS	eil76.tsp	100	128830	547.000000	1.672862
2	ACS	eil76.tsp	4096	129320	559.000000	3.903346
3	ACS two_opt	eil76.tsp	4	51414	538.000000	0.000000
4	ACS two_opt	eil76.tsp	100	119130	539.000000	0.185874
5	ACS two_opt	eil76.tsp	4096	45155	538.000000	0.000000
6	ACS two_opt candidate_list	eil76.tsp	4	89199	548.000000	1.858736
7	ACS two_opt candidate_list	eil76.tsp	100	89045	549.000000	2.044610
8	ACS two_opt candidate_list	eil76.tsp	4096	89309	539.000000	0.185874
9	ACS two_opt popmusic	eil76.tsp	4	0	inf	inf
10	ACS two_opt popmusic	eil76.tsp	100	0	inf	inf
11	ACS two_opt popmusic	eil76.tsp	4096	0	inf	inf
12	ACS	ch130.tsp	4	56450	6554.000000	7.266776
13	ACS	ch130.tsp	100	55870	6435.000000	5.319149
14	ACS	ch130.tsp	4096	55760	6371.000000	4.271686
15	ACS two_opt	ch130.tsp	4	29271	6110.000000	0.000000
16	ACS two_opt	ch130.tsp	100	35959	6155.000000	0.736498
17	ACS two_opt	ch130.tsp	4096	38181	6128.000000	0.294599
18	ACS two_opt candidate_list	ch130.tsp	4	32659	6170.000000	0.981997
19	ACS two_opt candidate_list	ch130.tsp	100	33616	6184.000000	1.211129
20	ACS two_opt candidate_list	ch130.tsp	4096	32087	6177.000000	1.096563
21	ACS two_opt popmusic	ch130.tsp	4	48026	6340.000000	3.764321
22	ACS two_opt popmusic	ch130.tsp	100	47047	6255.000000	2.373159
23	ACS two_opt popmusic	ch130.tsp	4096	47773	6247.000000	2.242226
24	ACS	d198.tsp	4	27840	17623.000000	11.679341
25	ACS	d198.tsp	100	27830	17446.000000	10.557668
26	ACS	d198.tsp	4096	27560	17244.000000	9.277567
27	ACS two_opt	d198.tsp	4	9207	15850.000000	0.443599
28	ACS two_opt	d198.tsp	100	9526	15834.000000	0.342205
29	ACS two_opt	d198.tsp	4096	9042	15867.000000	0.551331
30	ACS two_opt candidate_list	d198.tsp	4	15609	15986.000000	1.305450
31	ACS two_opt candidate_list	d198.tsp	100	14344	16174.000000	2.496831
32	ACS two_opt candidate_list	d198.tsp	4096	16159	15920.000000	0.887199
33	ACS two_opt popmusic	d198.tsp	4	23067	16235.000000	2.883397
34	ACS two_opt popmusic	d198.tsp	100	22682	16238.000000	2.902408
35	ACS two_opt popmusic	d198.tsp	4096	22737	15989.000000	1.324461

Next the above results were averaged across the seed, in order to better interpret the numbers.

Table 2

Instance	Method	Tours Generated	Best Cost	Best Gap%
eil76.tsp	ACS	128716.666667	553.333333	2.850062
eil76.tsp	ACS two_opt	71899.666667	538.333333	0.061958
eil76.tsp	ACS two_opt candidate_list	89184.333333	545.333333	1.363073
eil76.tsp	ACS two_opt popmusic	0.000000	inf	inf
ch130.tsp	ACS	56026.666667	6453.333333	5.619203
ch130.tsp	ACS two_opt	34470.333333	6131.000000	0.343699
ch130.tsp	ACS two_opt candidate_list	32787.333333	6177.000000	1.096563
ch130.tsp	ACS two_opt popmusic	47615.333333	6280.666667	2.793235
d198.tsp	ACS	27743.333333	17437.666667	10.504858
d198.tsp	ACS two_opt	9258.333333	15850.333333	0.445712
d198.tsp	ACS two_opt candidate_list	15370.666667	16026.666667	1.563160
d198.tsp	ACS two_opt popmusic	22828.666667	16154.000000	2.370089

Analyzing Table 1 and Table 2, it can be concluded that ACS plus twoopt is the best method in terms of best gap. Furthermore, for what regards the candidate list, the standard one always beat the popmusic one when it comes to best gap. Finally, we have that vanilla ACS was the worst between the proposed methods.

8. Drill Down on ACS Variant

Next the experimental results have been grouped by ACS variant. For each of such methods, the following tables report the best cost ever achieved between the experiments, the number of tours generated to reach such optimum, the average and standard deviation of tour lengths across iterations, the true optimum and the relative error.

8.1. ACS

Table 3: ACS

	Best Cost	Tours Generated	Average Cost	SD	Optimum	Best Gap%
eil76.tsp	547.000000	128830	553.333333	6.027714	538	1.672862
ch130.tsp	6371.000000	55760	6453.333333	92.867289	6110	4.271686
d198.tsp	17244.000000	27560	17437.666667	189.637373	15780	9.277567

8.2. ACS twoopt

Table 4: ACS twoopt

	Best Cost	Tours Generated	Average Cost	SD	Optimum	Best Gap%
eil76.tsp	538.000000	51414	538.333333	0.577350	538	0.000000
ch130.tsp	6110.000000	29271	6131.000000	22.649503	6110	0.000000
d198.tsp	15834.000000	9526	15850.333333	16.502525	15780	0.342205

8.3. ACS twoopt candidate list

Table 5: ACS twoopt candidate list

	Best Cost	Tours Generated	Average Cost	SD	Optimum	Best Gap%
eil76.tsp	539.000000	89309	545.333333	5.507571	538	0.185874
ch130.tsp	6170.000000	32659	6177.000000	7.000000	6110	0.981997
d198.tsp	15920.000000	16159	16026.666667	131.792767	15780	0.887199

8.4. ACS twoopt popmusic candidate list

Table 6: ACS twoopt popmusic candidate list

	Best Cost	Tours Generated	Average Cost	SD	Optimum	Best Gap%
eil76.tsp	inf	0	inf	NaN	538	inf
ch130.tsp	6247.000000	47773	6280.666667	51.539629	6110	2.242226
d198.tsp	15989.000000	22737	16154.000000	142.902064	15780	1.324461

Again what can be concluded is that the twoopt method was the one that performed best in terms of gap. Furthermore, it is also the one with the lowest average cost and the lowest standard deviation. Similarly to above, the classic candidate list outperformed the popmusic candidate list in terms of gap, average cost and standard deviation while on the other hand, popmusic was the one that generated more tours between the two within the three minutes limit.

9. ACS Comparison

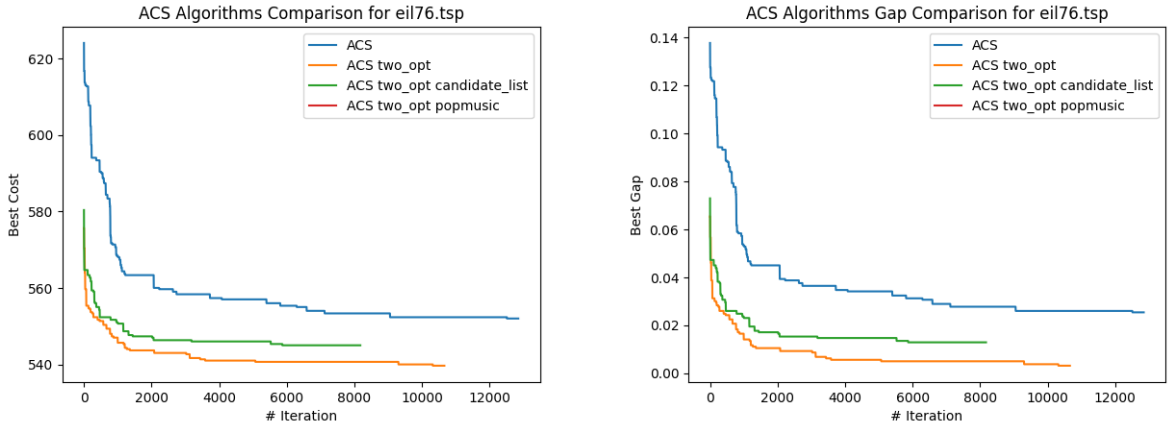

Figure 2

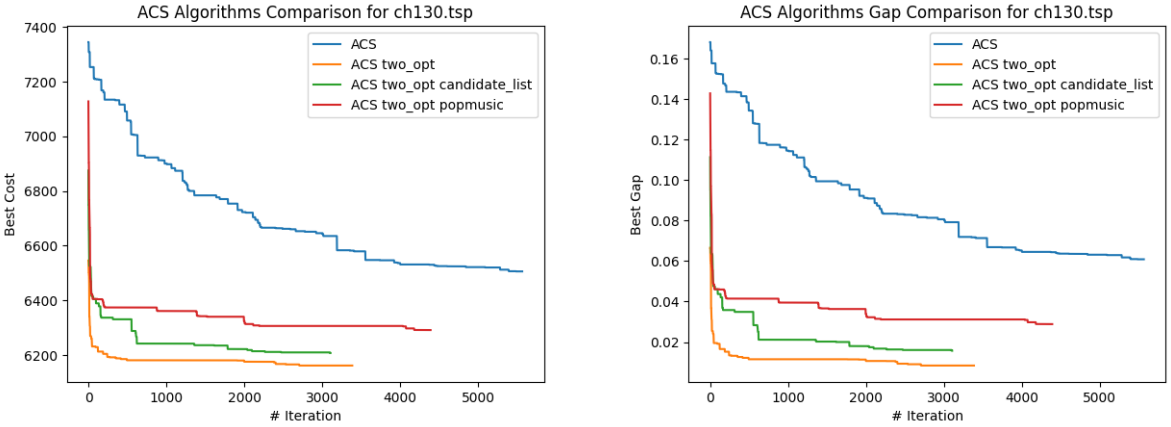

Figure 3

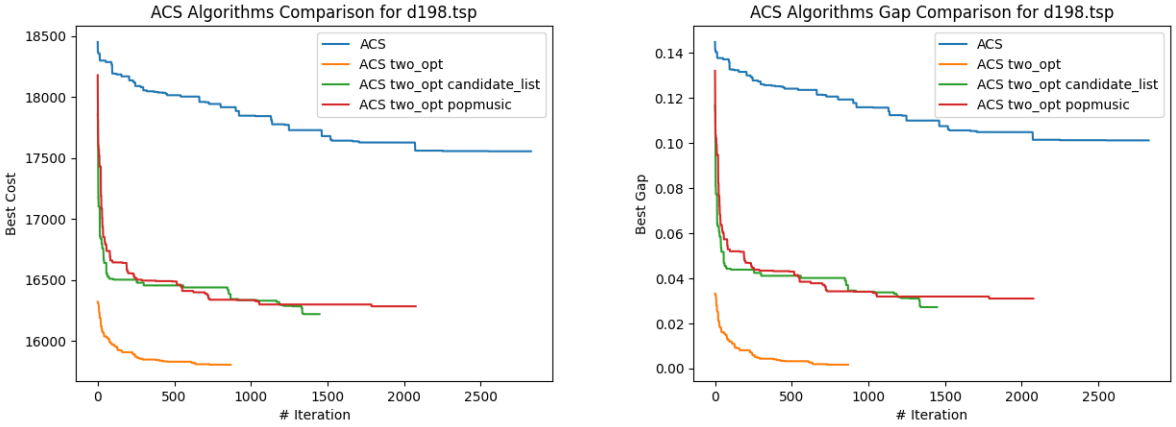

Figure 4

Finally, the average cost and gap across the seeds of each instance-method pair has been plotted. Notice that for the different seeds the number of iterations were different, so it has been necessary to pad the arrays in order to calculate their average. Padding was performed by repeating the last element of every list, until each array with the same method-instance pair had the same length The conclusion is the same that results from the tables above; the methods sorted from best

to worst, for the considered instances, are ACS twoopt, ACS twoopt candidate list, ACS twoopt popmuic candidate list and ACS. Still we have to keep in mind that during this study, we considered only small TSP problems. What might be interesting, is how would the methods rank on medium and big instances; what I expect is the importance of using a candidate list to increase.