Assignment.3 Selected topics in AI.2
Ant colony Optimization(ACO)

	Name	ID
1	Esraa yazid Ahmed	20190093
2	Mariam Ihab Mohamed	20190801
3	Christina Montasser Saad	20190382

Table of contents :

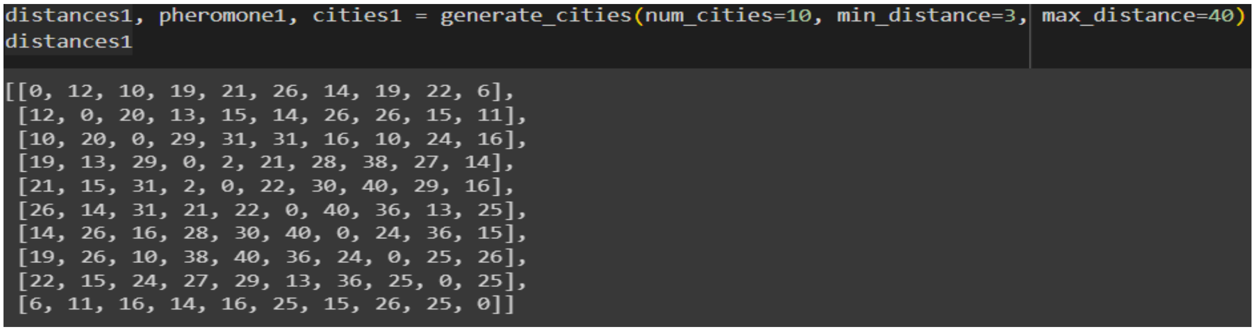
1. How does Ant colony work to solve this problem?

- **How does ACO work to solve TSB?**

In our code we have created generated set of cities , distances and pheromones set some hyperparameters to be used , to solve TSB using ACO it starts with each ant constructing a set of cities to visit putting in consideration that it should visit all the cities this set is constructed using probabilities that depend on the concentration of pheromone levels on edges between cities , after constructing the set the ant starts choosing the next city in the path this is has two conditions we choose a random value and if it's smaller than our exploration rate we choose the first city in the list of city with highest probability , if not we choose city randomly from the cities . After that we start updating pheromones levels on the edges this based on the quality of constructed solutions by ants , the best solutions deposit higher amounts of pheromones , updating pheromone levels on the edges also considers evaporating pheromones so the pheromone levels reduces gradually from edges till it reaches the smallest concentration on the least visited edges, this process is repeated till number of iterations ends then the optimal solution by the best ant after a specific batch of iterations will be printed

- **How do Distance matrices look ?**

As we have created a function called generate cities that takes the required number of cities as a parameter and returns a set of generated cities , initial pheromone matrix and a distance matrix that contains the euclidean distance between each two cities; we have used it to generate two sets of cities the first set includes 10 cities and the following figure shows the distance matrix of the 10 cities which is a matrix of size 10*10 that includes 100 elements and zero in its diagonal that notates the distance between the city and itself .



```
distances1, pheromone1, cities1 = generate_cities(num_cities=10, min_distance=3, max_distance=40)
distances1
```

```
[[0, 12, 10, 19, 21, 26, 14, 19, 22, 6],
 [12, 0, 20, 13, 15, 14, 26, 26, 15, 11],
 [10, 20, 0, 29, 31, 31, 16, 10, 24, 16],
 [19, 13, 29, 0, 2, 21, 28, 38, 27, 14],
 [21, 15, 31, 2, 0, 22, 30, 40, 29, 16],
 [26, 14, 31, 21, 22, 0, 40, 36, 13, 25],
 [14, 26, 16, 28, 30, 40, 0, 24, 36, 15],
 [19, 26, 10, 38, 40, 36, 24, 0, 25, 26],
 [22, 15, 24, 27, 29, 13, 36, 25, 0, 25],
 [6, 11, 16, 14, 16, 25, 15, 26, 25, 0]]
```

The other set of cities includes 20 cities , the following matrix shows how the distance matrix of 20 cities looks like and as shown it's a matrix of size (20*20)that includes 400 elements and zero in its diagonal that notates the distance between the city and itself .

```
35, 17,13,18,18,33,7,28,15,31],[31, 21, 38, 21, 10, 37, 16, 21, 22, 35, 0,
21, 25, 35, 18, 9, 35, 8, 20, 20],[10, 12, 17, 1, 15, 16, 24, 23, 16, 17,
21, 0, 15, 14, 13, 23, 14, 16, 9, 14],[22, 4, 27, 16, 14, 25, 35, 12, 3,
13, 25, 15, 0, 25, 6, 21, 16, 17, 6, 28],[4, 23, 3, 13, 30, 2, 34, 36, 27,
18, 35, 14, 25, 0, 26, 38, 11, 31, 21, 21],[23, 3, 29, 14, 8, 27, 30, 9,
4, 18, 18, 13, 6, 26, 0, 15, 20, 10, 5, 24],[34, 18, 41, 24, 8, 39, 25,
13, 18, 33, 9, 23, 21, 38, 15, 0, 34, 7, 18, 26],[11,  17,  12,  14,  26,
  11,  38,  29,  19,  7,  35,  14,  16,  11,  20,  34,  0,  28,  16,  26],
 [27, 13, 34, 17, 2, 32, 21, 14, 14, 28, 8, 16, 17, 31, 10, 7, 28, 0, 13,
20], [18, 3, 24, 10, 11, 22, 29, 14, 7, 15, 20, 9, 6, 21, 5, 18, 16, 13,
0, 21], [17,  24,  24,  13,  21,  23,  13,  33,  28,  31,  20,  14,  28,
21,24,26,26,20,21,0]]
```

---

2. Testing ant colony optimization on first set of cities (10 Cities):

- **Development of pheromone map :**

our pheromone map here are matrices that represent the concentration of pheromones over edges that link between cities; its size is equal to (number of ant agents * number of cities ants go through) , so in the following section for the single ant agent we have printed the full pheromone map for 1 ant , and for the other tests where we used different numbers of ants (5,10,20)we have printed the best ant pheromone that represents the concentration of pheromones over the edges of the optimal path.

2.1. "1" ant agent :

1. After 10 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.05594576553878604, 0.04966459189106596,
0.04671707577780853, 0.04477611155483193, 0.009851601422913504,
0.05082648923567699, 0.05052500124123736, 0.009975594581603414,
0.01156974179447115, 0.01194344959193464 7]]
```

2. After 20 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.05306608397893175, 0.05535111227099436,
0.049848238828907265, 0.040972807920559524, 0.03167267760354803,
0.04558546328391084, 0.04419381511786048, 0.05138283563926325,
0.05215479609263862, 0.028216293918810576]]
```

3. After 30 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.05630203803243486, 0.05439859818777367,
0.05011754480504096, 0.04702287340070295, 0.043780118469040646,
0.05069121535209145, 0.050205977640384966, 0.05301494568193637,
0.05328411164863818, 0.03957246047203558]]
```

4. After 40 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.006706043012956858, 0.05346653329186994,
0.01005221741025294, 0.04836333806386869, 0.040786142107096905,
0.039655463375898314, 0.05666619961534556, 0.005882353485353748,
0.005882352941393593, 0.0060066568292004826]]
```

5. After 50(iterations the pheromone map looked this way :

```
Pheromone Map = [[0.05909254341528774, 0.05802266987992792,
0.05540071460092812, 0.052758709734388905, 0.05236446643814032,
0.056342803944161135, 0.04983043041293453, 0.05718008875255336,
0.05788717433491362, 0.04767004392111571]]
```

2.2. “5” ant agents:

1. After 10 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.07473740895726141, 0.06325968546917517,
0.063054508788438, 0.01161468129359903, 0.0666652566936468,
0.06738021208715633, 0.013117130095370792, 0.07492931867240366,
0.013322306776107946, 0.01161468129359903]
```

2. After 20 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.06864387879372463, 0.05270193319184476,
0.053341304892377914, 0.04310566379797864, 0.0503461573946656,
0.0504426518960543, 0.039892900479427534, 0.04644673332489867,
0.056805127771178715, 0.04254923479580986]
```

3. After 30 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.00492083083304947, 0.05443466512886845,
0.004940547872463463, 0.0672263116009685, 0.02591805775808228,
0.006476582556051684, 0.05141836859103232, 0.006804193093280121,
0.00492083083304947, 0.011733519800492217]
```

4. After 40 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.0584762951690446, 0.0594334667177927,
0.064322811087438, 0.05691801556730083, 0.060396153070439086,
0.047831961885655847, 0.0404662113650035, 0.06876902141770357,
0.058545137081896785, 0.035336837578313685]
```

5. After 50 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.0709804085333784, 0.07164174178281771,
0.06459433306746391, 0.06337829242333182, 0.06292898175928703,
0.060663699216342536, 0.055264671839010814, 0.05442379696312934,
0.05392986283729888, 0.04274981554908746]
```

2.3. “10” ant agents :

1. After 10 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.06941018677808751, 0.06265045613411621,
0.060412733190535035, 0.0064031145737225015, 0.011469137784393861,
0.007376471348482652, 0.017119798986381753, 0.021106644799358233,
0.020851304098988842, 0.009194838767068985]
```

2. After 20 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.006019285416501922, 0.05976267552865127,
0.06036145193818292, 0.006068046568027085, 0.01741710749460714,
0.015311717575153396, 0.006370768196631413, 0.006517752694787738,
0.006019286990432842, 0.006148607449122482]
```

3. After 30 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.07225689895513285, 0.07094492266686393,
0.006451336975733949, 0.057507186577757601, 0.058680499947752096,
0.007710975643430554, 0.063673846528961, 0.009272794780894872,
0.006451328588413163, 0.006489061376775425]
```

4. After 40 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.062154830457397434, 0.06568050759988234,
0.05927447954856235, 0.005921926525922771, 0.06415678171761578,
0.011587937075583195, 0.005804011602876464, 0.005889631963481601,
0.06898924507552848, 0.005693585836541463]
```

5. After 50 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.06391573469407101, 0.06750987379909311,
0.06920787609098142, 0.06216003600207873, 0.06532232640393923,
0.05798878170942004, 0.007193401177692046, 0.055165246189329394,
0.02231302504400479, 0.007181713989932307]
```

2.4. “20” ant agents :

1. After 10 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.006525570044436712, 0.04765726913597191,
0.018620594748483276, 0.011194954059354107, 0.05392338708147479,
0.06196021687791272, 0.01828064569447372, 0.01607320379174435,
0.008146673700478765, 0.006668683181361994]
```

2. After 20 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.07049774938610166, 0.06077049840261894,
0.001963341389230084, 0.05708355165987006, 0.06507874425418532,
0.011771762271464007, 0.052693472538701186, 0.05927038863550082,
0.06244401132754993, 0.0026886250674072625]
```

3. After 30 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.06632981720315366, 0.0035057159015760065,
0.05262824221440144, 0.06329180848713023, 0.0019423691711708 69,
0.05099325693882555, 0.06332629101502989, 0.06187422056756705,
0.07070931669301844, 0.004416596948266136]
```

4. After 40 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.06345380827858171, 0.0019774938534835866,
0.05728026411392915 5, 0.06564202065836867, 0.0014758567876116875,
0.00908723558519697, 0.008690442540359954, 0.009165989311834637,
0.009109396787654 27, 0.0018860839347759099]
```

5. After 50 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.063103905669722, 0.06199620190269356 7,
0.06034705594721849 5, 0.05957955520575808 4, 0.06437685725227985,
0.05918084027443254 5, 0.05902293685273761 5, 0.06998611039551435,
0.06451272992227254, 0.057425140043371 9]
```

- **Results :**

In the following table we show the best distance (the shortest path between 10 cities starting and ending at the same city”0”) after each 10 iterations and the best path that the ant agent has taken

Number of ants	**Number of iterations**	**Best Distance**	**Best path**
1 ant	**10**	**124.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**
	20	**124.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**

	30	**124.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**
	40	**124.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**
	50	**124.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**
5 ants	**10**	**105.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**
	20	**124.0**	**[0, 1, 8, 4, 9, 6, 7, 2, 3, 5, 0]**
	30	**120.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**
	40	**124.0**	**[0, 8, 4, 6, 7, 9, 2, 3, 5, 1, 0]**
	50	**124.0**	**[0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0]**
10 ants	**10**	**114.0**	**[0, 1, 8, 4, 9, 6, 7, 3, 2, 5, 0]**
	20	**109.0**	**[0, 1, 8, 4, 7, 6, 9, 3, 2, 5, 0]**
	30	**113.0**	**[0, 5, 2, 3, 7, 6, 9, 4, 8, 1, 0]**
	40	**118.0**	**[0, 1, 8, 9, 6, 7, 2, 3, 5, 4, 0]**
	50	**121.0**	**[0, 1, 8, 4, 6, 7, 9, 3, 5, 2, 0]**
20 ants	**10**	**112.0**	**[0, 5, 2, 3, 9, 6, 7, 4, 8, 1, 0]**
	20	**106.0**	**[0, 2, 3, 5, 9, 6, 7, 4, 8, 1, 0]**
	30	**106.0**	**[0, 1, 8, 9, 6, 7, 4, 5, 2, 3, 0]**
	40	**113.0**	**[0, 1, 9, 6, 7, 8, 4, 5, 2, 3, 0]**
	50	**124.0**	**[0, 1, 9, 6, 7, 4, 8, 3, 2, 5, 0]**

- **Comments and notes :**

1. We have tested our ACO algorithm on 10 cities with different number of ants (1,5,10,20)

2. We can see from the previous table that our optimal path that goes from starting city(0) and ends at the same city is given by the configuration in which we have used 5 ants.
3. The optimal path is [0, 1, 8, 4, 5, 2, 3, 9, 6, 7, 0] and it results in an overall distance that is equal to 105.0 after
4. Also by noticing pheromones maps that provided in the notebook after running the algorithm on each different configuration we can see that even in the cases in which we got the same best distances through different batches of iteration using same number of ants we still have different values in the pheromone maps (different concentrations).
5. We can also see that going deeper (running more iterations ex: going from 10 to 50 iterations)increases the concentration of the pheromone on the edge (link between two cities)

---

3. Testing ant colony optimization on second set of cities (20 Cities):

- Development of pheromone map : as the size of the pheromone map matrix is going to be a bit bigger due to working on 20 cities knowing that the size of the matrix will be equal to number ant agents multiplied by the number of cities) so here in the following section we have showed the results of

3.1. “1” ant agents

1. After 10 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.04427515919097719, 0.0045727437879401365,
0.012067343141633014, 0.009183326037917102, 0.008288066265496991,
0.030096632295025075, 0.0072158422484457545, 0.008430132340077223,
0.02470516552242824, 0.029721363839265068, 0.033861565815372925,
0.037056019719369795, 0.027900451325769328, 0.037014486435199216,
0.007864922601825931, 0.00559098081448083, 0.004908786001389632,
0.04388631761978506, 0.04250090413038391, 0.004575284763220957]]
```

2. After 20 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.04105688784475334, 0.03251738120055242,
0.03576521661299854, 0.034235085230163484, 0.024263165729353692,
0.027596792344671387, 0.01957721535273778, 0.03302453672375678,
0.03272244188336852, 0.027609414304809635, 0.023143327541227172,
0.0299172256580457, 0.022121864975395325, 0.02457693695717731,
0.03279028747261794, 0.03447719850870357, 0.03287728838451024,
0.027167586973537612, 0.03138628387699577, 0.012859101735244578]]
```

3. After 30 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.04438755317063949, 0.03738238700532366,
0.004128440366972477, 0.03982584328870765, 0.032935134568167525,
0.032099885958476286, 0.004458915080871194, 0.024257151724433965,
0.029665102416960847, 0.03727680664596183, 0.032691328170165644,
0.026095598660027523, 0.03611294430113859, 0.004147357079285655,
0.03595868561598881, 0.009242468884400844, 0.0058895417359467505,
0.016337277096810293, 0.004128440367151236, 0.008202095098070997]]
```

4. After 40 iterations the pheromone map looked this way :

```
Pheromone Map = [[0.04558121563059251, 0.036661506785162654,
0.0038297974431555646, 0.03457497135972648, 0.020009869639049754,
0.0038297891959110843, 0.006906452643560058, 0.004591539951628055,
0.01302507474049585, 0.0065320659416797325, 0.028917910144708656,
0.03755767877025663, 0.03551726267389133, 0.003948647959883976,
0.009526243527639823, 0.007964488554925936, 0.0056370861913795705,
0.003918688761699985, 0.01602929172888824, 0.0038299636488773876]]
```

5. After 50(iterations the pheromone map looked this way :

```
Pheromone Map = [[0.03462031068007081, 0.03552594844108965,
0.026660659302241707, 0.03245573723992096, 0.02860556297104252,
0.036734700133358965, 0.025812289721459762, 0.030647427605298386,
0.028990925491584565, 0.03772629573877094, 0.0347018955696558,
0.038194624617870654, 0.02880872157332994, 0.017591716271801276,
0.032330476379188594, 0.038783878836368094, 0.03168607377927945,
0.02598388104599825, 0.037631738782478885, 0.018944914892692274]]
```

3.2. “5” ant agents:

1. After 10 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.04084910539881985, 0.0368144006916206,
0.03823166972023811, 0.0364488858052447, 0.0034045657609950044,
0.02986396049142043, 0.02429016727582482, 0.03287709382859289,
0.027233974794887907, 0.035606215926220736, 0.0033742285714285715,
0.03983305750743922, 0.03294572505589428, 0.0408244069295755,
0.003374228804906076, 0.006091192649788651, 0.0036328887744552165,
0.03554663483853244, 0.007724033833471351, 0.006421813091458755]
```

2. After 20 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.037505986864187935, 0.04010213184218377,
0.03989050269312889, 0.03925845814436529, 0.03881983763204315,
0.04069668944821978, 0.03430119727382439, 0.041636348614120584,
0.030895959481685054, 0.039577818826345715, 0.034555697989660655,
0.040248881835878145, 0.03759180531720032, 0.029882523160747194,
0.0366125215447482, 0.03551961260639017, 0.0356541934380328,
0.032877163993443946, 0.038942306675106696, 0.014139728031883706]
```

3. After 30 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.0413888985168649826, 0.0411113251571303,
0.039609597953851695, 0.040211706835599874, 0.039440469070608886,
0.040909135799234476, 0.035640945636800016, 0.030977075240827268,
0.027192648168663878, 0.03631494914272287, 0.029330055960291028,
0.0283076652556789, 0.02881827019767767, 0.016888876474240894,
0.0346098831821739, 0.021882156231017327, 0.02878260739766697,
0.019733177236616557, 0.029870579284567968, 0.008064236334142225]
```

4. After 40 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.002986555260684398, 0.02481043746096451,
0.018676288410254063, 0.027222576864463534, 0.01029358908204752,
0.029322713690616755, 0.03291602148981046, 0.031509900107112694,
0.02632200523022952, 0.010536733786102372, 0.03302289032750761,
0.006092202413424073, 0.03160312120249029, 0.003852049946460039,
0.003578331367483756 7, 0.0036702924400949023,
0.003039345448097376, 0.004718027665016638, 0.006820712345310789,
0.003553567052656755]
```

5. After 50 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.04310392207252323, 0.04186171712644314,
0.041864825700262426, 0.037450017829964065, 0.034689117529045424,
0.03712407279301979, 0.03103047645892568, 0.003417187558872167 6,
0.017407429209749196, 0.03479846625636644, 0.02636599776388738,
0.021447596203520915, 0.026233646324235346, 0.013415235215519895,
0.03807218540100468, 0.003417524516797724, 0.00668292114661294 3,
0.004862163568058709 5, 0.03164422377657911, 0.007179972467736669]
```

3.3. “10” ant agents

1. After 10 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.03472789775396439, 0.03139935994673246,
0.026980073253986502, 0.034468338093403744, 0.03673715177766688,
0.03074159329842599, 0.023714584322973693, 0.03803940324023524,
0.030898846874109247, 0.03217004558420344, 0.030341092230022494,
0.028924974715076116, 0.023541828384553178, 0.020124416170766248,
0.02098832100178486, 0.03476614250549448, 0.02758348732624598,
0.0198183331724492, 0.02933296740304386, 0.01565216668996618]
```

2. After 20 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.03900746533765927 4, 0.04035649004235935 4,
0.040569415065875115, 0.041383559696866 79, 0.03664027325317756,
0.038893911539322465, 0.034585090009438456, 0.043465787502252246,
0.02823946693688282, 0.03989968723508417, 0.03112658090320291 6,
0.03080589507649169, 0.03433800332110842, 0.02654454810919699,
0.02823410263003582 3, 0.03861078719862878, 0.02978108675229928 4,
0.02608974237626396, 0.03689308017176331, 0.01960387990330135]
```

3. After 30 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.04324427090120477, 0.04571145108173362,
0.003073935565653983 6, 0.016929347821292615, 0.04272040143294656 6,
0.01128343733860095 7, 0.01445942599108916 9, 0.00782660788489809 2,
0.003549491997459576 4, 0.00337579256558776 83,
0.02737961389468260 3, 0.04334627012966223, 0.02490525487145871,
0.02127479268057814, 0.03667946584813868, 0.01252292176334724,
0.03154316794784139, 0.004718587023411437 5, 0.01320787368267446,
0.003078653213122543 7]
```

4. After 40 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.03799943061405241, 0.03532406097106481 6,
0.02912671615823437 7, 0.02463420523275371 5, 0.03081574791686056 8,
0.03569731907071425, 0.02322307388509384 2, 0.03609011407560209,
0.0340887520118492, 0.03192385659363325, 0.03435438423338091,
0.03368659964520550 5, 0.01886044051663041, 0.01569965080189114 6,
0.01418306552757303 6, 0.03562149236211661, 0.02843314313462957,
0.01856789873433495 7, 0.02188740062830667, 0.00806752669371027 9]
```

5. After 50 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.0383105458131975, 0.03355197443665226,
0.02649649376237958, 0.03318659122356506 4, 0.03004461225171726 3,
0.03433832916476165 6, 0.02184035870707213, 0.0350146116951168 5,
0.03714860863866658 5, 0.03703813067794744 4, 0.03290379953883601 5,
0.03416314754789133, 0.02976558637563073 5, 0.02354935722184644,
0.02984710096698696, 0.02817548667002069 8, 0.03373309311040731,
0.03018630034490783 4, 0.03485185665037792, 0.01482917992097577]
```

3.4. “20” ant agents

1. After 10 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.040525108252583826, 0.036915359979584034,
0.03776510412628691, 0.040166882399998828, 0.04128935205270259,
0.0374429429740165, 0.0013590844700530696, 0.024464394172002597,
0.03536051557606341, 0.03126820402344573, 0.02494239609700262,
0.041698761631298695, 0.016487503250871453, 0.042879502788344226,
0.04141615854994922, 0.034956076555107005, 0.03903812575838722,
0.03532851585557807, 0.01577798379217804, 0.00474640553519225 3]
```

2. After 20 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.04132714244691361, 0.03923634330756328,
0.04039929460810234, 0.0007694725046193 02, 0.023760583397341037,
0.040559962321186444, 0.026176372568253097, 0.020385085858435823,
0.0291487831598948, 0.007141897137377233, 0.03833473060603968,
0.04173129612718609, 0.03099456161631 1724, 0.028677547145656797,
0.025990992403054608, 0.016805439419486846, 0.005487996226650049,
0.008716231952798184, 0.0020186138408462145, 0.0018785800126168 31]
```

3. After 30 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.04320041793519471, 0.04360265205475284,
0.0325024839471034, 0.004152531655696907, 0.02093907523674027,
0.035798456439628 77, 0.033240010607162 94, 0.019857040426006323,
0.0312610686080788, 0.013743892008482032, 0.03867713745209345,
0.038985023555002905, 0.031488605106951 37, 0.039023246834886574,
0.03505673382971681, 0.022756597916113838, 0.012617232461887 45,
0.017413465055077 64, 0.009994876410999533, 0.004367059069283739]
```

4. After 40 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.03710927676 4964214, 0.03678483167997187,
0.03109317984833891, 0.03530031791064277, 0.026342978151344038,
0.03232568272861503, 0.008371502586618158, 0.0280909483375023,
0.03829613972187168, 0.03861257748083267, 0.025988765871314722,
0.03382661243317935, 0.009120974660966577, 0.03352153260662238,
0.03580909168391326, 0.03250233260746425, 0.03574514869065844,
0.032913793041326564, 0.018985751355292183, 0.010168021101918525]
```

5. After 50 iterations the best ant pheromone map looked this way :

```
Pheromone Best Ant = [0.04203316936281561, 0.03987063150257774,
0.03850710143961625, 0.03603233171492463, 0.03281292131686415,
0.031882238817680764, 0.03147163744538859, 0.03301820562339486,
0.03129262500488729, 0.028115014736488936, 0.03284572958014865,
0.02984283861816719, 0.02767961820644443, 0.016732610454598975,
0.028140714987754507, 0.032364715878099456, 0.024366761223749242,
0.013825112354852431, 0.029307582765429902, 0.007100306073561225]
```

- **Results:**

In the following table we show the best distance (the shortest path between 20 cities starting and ending at the same city”0”) after each 10 iterations and the best path that the ant agent has taken

Number of ants	**Number of iterations**	**Best Distance**	**Best path**
1 ant	**10**	**204.0**	**[0, 11, 13, 5, 17, 3, 12, 10, 9, 7, 14, 1, 4, 18, 15, 6, 19, 16, 8, 2, 0]**
	20	**181.0**	**[0, 11, 9, 10, 12, 1, 3, 14, 7, 15, 6, 19, 16, 8, 2, 18, 4, 13, 5, 17, 0]**
	30	**218.0**	**[0, 11, 13, 5, 17, 4, 18, 14, 1, 3, 12, 10, 9, 7, 15, 6, 19, 16, 8, 2, 0]**
	40	**235.0**	**[0, 11, 13, 14, 1, 3, 12, 7, 15, 6, 19, 16, 8, 2, 4, 18, 17, 9, 10, 5, 0]**

	50	181.0	[0, 11, 13, 8, 2, 3, 18, 4, 17, 5, 15, 6, 19, 16, 9, 10, 12, 1, 7, 14, 0]
5 ants	10	175.0	[0, 11, 13, 5, 17, 4, 18, 14, 3, 12, 10, 9, 7, 15, 6, 19, 16, 8, 2, 1, 0]
	20	181.0	[0, 11, 13, 5, 17, 15, 6, 19, 16, 8, 2, 14, 1, 3, 12, 9, 10, 4, 18, 7, 0]
	30	181.0	[0, 11, 13, 5, 17, 4, 18, 14, 1, 3, 12, 10, 9, 7, 15, 6, 19, 16, 8, 2, 0]
	40	198.0	[0, 11, 13, 5, 17, 4, 18, 14, 1, 3, 12, 10, 9, 7, 15, 6, 19, 16, 8, 2, 0]
	50	192.0	[0, 19, 16, 8, 2, 3, 12, 10, 9, 7, 14, 1, 4, 18, 17, 5, 13, 11, 6, 15, 0]
10 ants	10	181.0	[0, 11, 13, 5, 17, 15, 6, 19, 16, 8, 2, 1, 3, 12, 10, 9, 7, 14, 18, 4, 0]
	20	181.0	[0, 11, 13, 5, 17, 4, 18, 14, 1, 3, 12, 10, 9, 7, 15, 6, 19, 16, 8, 2, 0]
	30	173.0	[0, 11, 13, 5, 17, 4, 18, 14, 1, 3, 12, 10, 9, 7, 15, 6, 19, 16, 8, 2, 0]
	40	181.0	[0, 11, 13, 7, 14, 1, 18, 4, 17, 5, 15, 6, 19, 16, 8, 2, 3, 12, 9, 10, 0]
	50	181.0	[0, 11, 13, 5, 17, 4, 18, 14, 1, 3, 12, 10, 9, 7, 15, 6, 19, 16, 8, 2, 0]
20 ants	10	171.0	[0, 11, 13, 18, 4, 17, 5, 12, 10, 9, 7, 14, 1, 3, 2, 8, 16, 6, 19, 15, 0]
	20	158.0	[0, 11, 13, 5, 17, 4, 18, 15, 6, 19, 16, 8, 2, 1, 3, 12, 10, 9, 7, 14, 0]
	30	158.0	[0, 11, 13, 5, 15, 6, 19, 16, 8, 2, 1, 3, 12, 10, 9, 7, 14, 18, 4, 17, 0]

	40	**171.0**	**[0, 11, 13, 5, 15, 6, 19, 16, 8, 2, 1, 3, 12, 10, 9, 7, 14, 18, 4, 17, 0]**
	50	**181.0**	**[0, 11, 13, 5, 17, 4, 18, 15, 6, 19, 16, 8, 2, 1, 3, 12, 10, 9, 7, 14, 0]**

- **Comments and notes :**

1. We have tested our ACO algorithm on 20 cities with different number of ants (1,5,10,20)
2. We can see from the previous table that we have 2 paths with the smallest best distance value which is equal to 158, we can see that we got this optimal distance by running our agent using 20 ants .
3. The first optimal distance appeared after 20 iterations resulting in this optimal path : [0, 11, 13, 5, 17, 4, 18, 15, 6, 19, 16, 8, 2, 1, 3, 12, 10, 9, 7, 14, 0]
4. The other path that also gave optimal distance value appeared after 30 iterations with this optimal path :[0, 11, 13, 5, 15, 6, 19, 16, 8, 2, 1, 3, 12, 10, 9, 7, 14, 18, 4, 17, 0]
5. We can see that despite the fact that we got the same optimal distance after 2 different batches of iterations ; the path that the ants took is different (the optimal path after 20 iterations is not the same path after 30 iterations.
6. Also by noticing pheromones maps that provided in the notebook after running the algorithm on each different configuration we can see that even in the cases in which we got the same best distances through different batches of iteration using same number of ants we still have different values in the pheromone maps (different concentrations).
7. We can also see that going deeper (running more iterations ex: going from 10 to 50 iterations)increases the concentration of the pheromone on the edge (link between two cities)

4. General comments and Results :

- We can see that using larger number of cities (20 cities instead of 10) gave us higher variety of best distances among number of iterations and different number of ant agents ; in the first configuration in which we used 10 cities we can see that using a single ant gave us the same best distance after 10 , 20 , 30, 40 and 50 iterations this doesn't mean that they have the same pheromone map ; meanwhile running it with 20 cities using a single ant also resulted in various values of best distances among different number of iterations
- In the case of 20 cities we can see that 20 ants gave us the optimal distance and path which is 158.0 , the smallest next distance is 173.0 and it was given by 10 ants , the one after this is 175 and was given by 5 ants and the largest best distance is 181.0 and was given by a single ant .
- Based on the previous note we can say that in the case of 20 cities increasing the number of ant agents helped to have a better exploration through space (20 cities) and helped us reach the most optimal distance and most optimal path.
- In the case of 10 cities we can see that the best distance appeared when using only 5 ants , and using bigger number of ants (10,20) did not gave us smaller distances through batches of iterations , we can say that for 10 cities only 5 ants were enough to explore the search space and get the optimal distance and path, and higher number of ants were redundant and couldn't give better results
- As a conclusion we can say that increasing the number of ants was effective in the case of relatively big search space as it led to more exploration and in the case of the relatively small search space the big number of ants was not effective .