Project Report

On

**Quick and Speedy Delivery**

Design and Analysis of Algorithms

By

S.J Sumanth (2010030377)

K. Sreevarun (2010030451)

E. Pavan Sai (2010030538)

E. Shiva Goud (2010030542)

under the supervision of

# **Mrs. Deepthi Kalavala**

# Assistant Professor



Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

March, 2022

**Declaration**

The Project Report entitled “QUICK AND SPEEDY DELIVERY” is a record of bonafide work of SJ Sumanth (2010030377), K. SreeVarun (20100030451),

E. Pavan Sai (2010030538), E. Shiva Goud (2010030542) submitted in partial fulfillment for the award of B. Tech in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/ University/ Institute.

**Certificate**

This is to certify that the Project Report entitled “QUICK AND SPEEDY DELIVERY” is being submitted by SJ Sumanth (2010030377), K. SreeVarun (20100030451), E. Pavan Sai (2010030538), E. Shiva Goud (2010030542), submitted in partial fulfillment for the award of B. Tech in CSE to the K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision. The results embodied in this report have not been copied from any other departments/ University/Institute.

## Signature of the Supervisor

Mrs. Deepthi Kalavala

Assistant Professor

## Signature of the HOD Signature of the External Examiner

**ACKNOWLEDGEMENT**

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for completing this Project successfully.

We take grateful opportunity to thank our beloved Founder and Chairman who has given constant encouragement during our course and motivated us to do this Project. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgement & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike** for his exemplary guidance, monitoring and constant encouragement throughout the course of the Project. We thank **Ms.** **Deepthi Kalavala** of our department who has supported throughout this project holding a position of supervisor.

We whole heartedly thank all the teaching and non-teaching staff of our department without whom we won’t have made this project a reality. We would like to extend our sincere thanks especially to our parent, our family members and friends who have supported us to make this project a grand success.

**ABSTRACT**

We have many food delivery apps like Swiggy, Zomato, Eat Sure etc. Daily thousands of orders are made through these apps. In most cases, a driver should take more than one order at a time and deliver the same to respective customers in the given time. So, in order to deliver all orders in the given time the delivery executive should choose the optimal path from source to destination. He should delivery in time with minimum distance so he can save his time and get profit. But the original problem arises here, now what if there are multiple delivery executives with ‘x’ orders which are to be delivered in ‘n’ cities, now this ultimately becomes a multiple travelling salesman problem which can be solved by various methods like ACO (Ant Colony Optimisation) algorithm, GA (Genetic Algorithm), Two phase heuristic algorithm etc. Multiple Traveling Salesman Problem (MTSP) is a generalization of the Traveling Salesman Problem (TSP). MTSP is an optimization problem to find the minimum total distance of m salesmen tours to visit several cities in which each city is only visited exactly by one salesman, starting from origin city called depot and return to depot after the tour is completed.

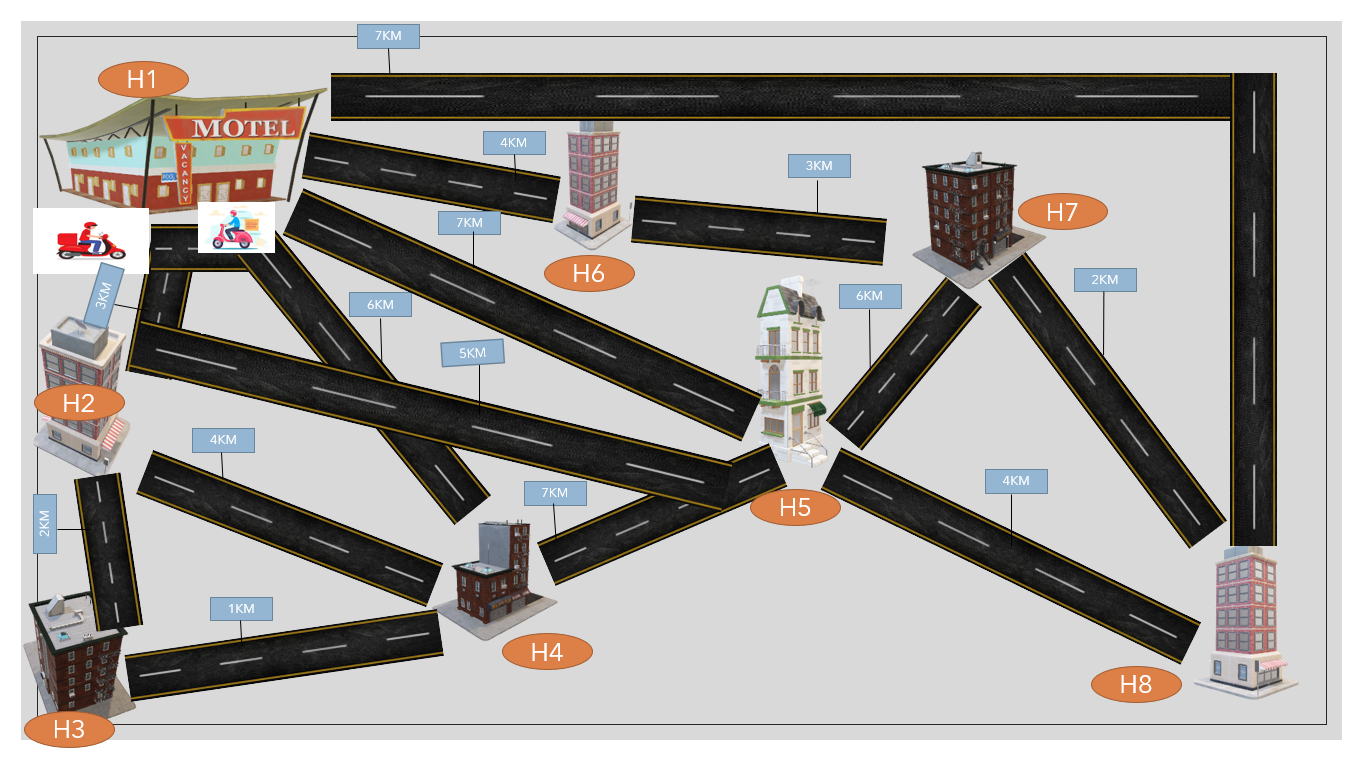
# **1.Introduction**

One of the most fundamental goals of machine learning is to provide learned approximations to combinatorial optimization problems of intractable complexities. Image segmentation, pose estimation, sequence alignment, parsing, machine translation, protein design, etc. could be defined as large-scale combinatorial problems, in which variables are assigned discrete values, based on a cost term that involves the observations and perhaps elements of the training data. Classical NP hard combinatorial problems pose a unique set of challenges. First, obtaining a training sample often requires solving the problem itself, which is feasible only at relatively small scales. Second, unlike perceptual data, these problems are often defined on unordered or specially structured inputs. Third, the feasible solutions are subject to a complex set of constraints. These challenges call for dedicated research and the design of differentiable architectures that can:

(i) well represent the invariances in the input, and

(ii) enforce the constraints on the output.

In addition to the scientific challenge of approximating these abstract problems with Machine Learning (ML), solving them has a practical value. The classical combinatorial optimization problems have numerous real-world applications, many of which are in management: resource allocation, optimal queuing, task planning, etc. The Multiple Traveling Salesmen Problem (mTSP), which is the focus of this work, is a generalization of the well-known Traveling Salesman Problem (TSP). Given a set of cities, m ≥ 1 salesmen, one depot where salesmen are initially located and to which they return, and a pairwise distance matrix, the objective of the mTSP is to determine a route for each salesman, such that the total length of the routes is minimized, and such that each city is visited exactly once by any of the salesmen. In comparison to the TSP, the mTSP has been the focus of relatively little research. However, as a natural generalization, it captures many more real-world problems, like Logistics Delivery, Food Delivery etc. So, in order to solve this complex problem, we are using ACO (Ant Colony Optimization) technique.

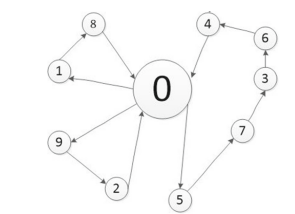
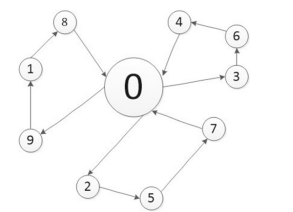
****

**2.RELATED WORK**

A Multi-travelling salesman problem (MTSP) is characterized by more than one salesman and as a consequence, MTSP typically has a higher level of complexity compared to TSP.MTSP can be converted into TSP, and Gorenstein (1970) propose a basic strategy to achieve this, based on m salesmen, andm−1 virtual city. These are used to define a gap between different travelling salesmen, whilst the distance between the virtual cities is considered infinite. Yuan et al. (2013) discuss a new crossover operator called two-part chromosome crossover for the genetic algorithm in order to obtain nearoptimal solutions of MTSP. However, this method is affected by the growth of the chromosome length and the overall cost of the solution. Kalia Perumal et al. (2015) present the Modified Two-Part Chromosome Crossover to address MTSP by employing a genetic algorithm for nearby optimal solutions. However, this method allocates a different number of the cities for each salesman, and therefore, it cannot successfully address MTSP with workload balance. Osaba et al. Hossein Abadi et al. (2014) propose the Real-World Dial-a-Ride problem, which is modelled as a MTSP. In particular, they propose

GELS-GA, a new hybrid algorithm, which achieves optimal values even in highly complex scenarios. Finally, Alves and Lopes (2015) consider the workload balance of MTSP and develop GA to reduce both the overall distance and the difference between the distances travelled by each salesman.

**3.PROBLEM DEFINITION AND FORMULATION**

****  ****

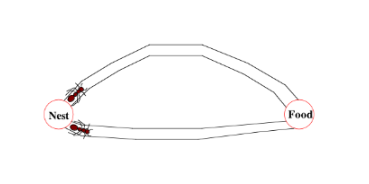
**Fig 1(a). mTSP Fig 1(b). mTSP with balance workload**

Typically, MTSP only aims to obtain the least total cost of distance or time. Fig. 1 depicts a scenario defined by 3 salesmen and 9 cities, where node 0 is the starting and ending point. As shown in Fig. 1a, there are two salesmen traversing two cities, respectively, whereas all the other ones are traversed by the third salesman. It is clear that the salesmen’s workloads are unbalanced. As discussed above, the main objective of MTSP is to minimize the overall distance travelled by all salesmen, which may cause an unbalanced workload problem. Figure 1b shows an ideal solution for MTSP with workload balance. In order to achieve a balanced allocation of workload, there are typically two different strategies. The first aims to balance the number of cities assigned to each salesman, whereas the second focuses on optimizing the balance of the distances travelled by each salesman.

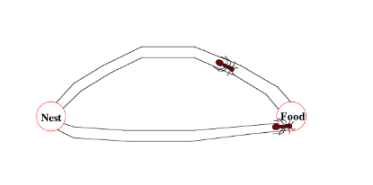
**4.ANT COLONY OPTIMIZATION**

Optimization problems are very important in the field of both scientific and industrial. Some real-life examples of these optimization problems are time table scheduling, nursing time distribution scheduling, train scheduling, capacity planning, traveling salesman problems, vehicle routing problems, Group-shop scheduling problem, portfolio optimization, etc. Many optimizations’ algorithms are developed for this reason. Ant colony optimization is one of them. Ant colony optimization is a probabilistic technique for finding optimal paths. In computer science and researches, the ant colony optimization algorithm is used for solving different computational problems. Ant colony optimization (ACO) was first introduced by Marco Dorigo in the 90s in his Ph.D. thesis. This algorithm is introduced based on the foraging behavior of an ant for seeking a path between their colony and source food. Initially, it was used to solve the well-known traveling salesman problem. Later, it is used for solving different hard optimization problems. Ants are social insects. They live in colonies. The behavior of the ants is controlled by the goal of searching for food. While searching, ants roaming around their colonies. An ant repeatedly hops from one place to another to find the food. While moving, it deposits an organic compound called pheromone on the ground. Ants communicate with each other via pheromone trails. When an ant finds some amount of food it carries as much as it can carry. When returning it deposits pheromone on the paths based on the quantity and quality of the food. Ant can smell pheromone. So, other ants can smell that and follow that path. The higher the pheromone level has a higher probability of choosing that path and the more ants follow the path, the amount of pheromone will also increase on that path.

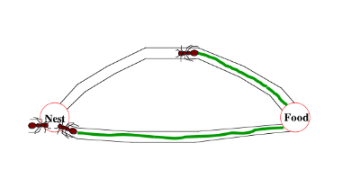
Let’s see an example of this. Let consider there are two paths to reach the food from the colony. At first, there is no pheromone on the ground. So, the probability of choosing these two paths is equal that means 50%. Let consider two ants choose two different paths to reach the food as the probability of choosing these paths is fifty-fifty.



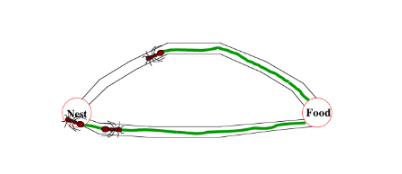
The distances of these two paths are different. Ant following the shorter path will reach the food earlier than the other.



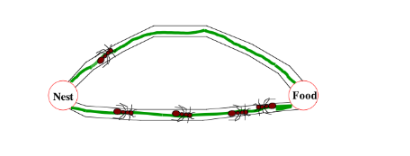
After finding food, it carries some food with itself and returns to the colony. When it tracking the returning path, it deposits pheromone on the ground. The ant following the shorter path will reach the colony earlier.



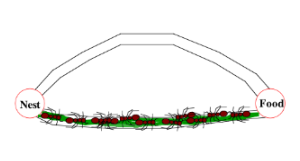
When the third ant wants to go out for searching food it will follow the path having shorter distance based on the pheromone level on the ground. As a shorter path has more pheromones than the longer, the third ant will follow the path having more pheromones.



By the time the ant following the longer path returned to the colony, more ants already have followed the path with more pheromones level. Then when another ant tries to reach the destination(food) from the colony it will find that each path has the same pheromone level. So, it randomly chooses one. Let consider it choose the above one(in the picture located below)



Repeating this process again and again, after some time, the shorter path has a more pheromone level than others and has a higher probability to follow the path, and all ants next time will follow the shorter path.



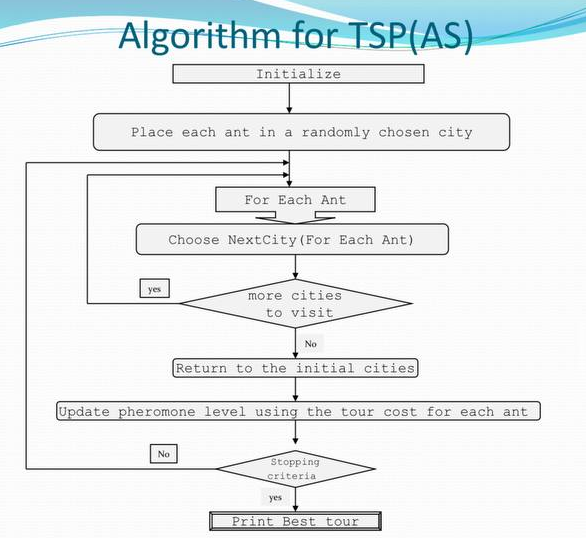
For solving different problems with ACO, there are three different proposed version of Ant-System:

**Ant Density**& **Ant Quantity:**Pheromone is updated in each movement of an ant from one location to another.

**Ant Cycle:**Pheromone is updated after all ants completed their tour.

**5. Working of Ant Colony Optimisation**

* First, divide the cities into clusters, then assign each ant to a random city.
* If there are more cities to travel, move the ants to next cities.
* After successful traversal of all cities, make ants to return to their initial position.
* Now, each ants in the colony constructs a solution based on previously deposited pheromone trails.
* Next ants will lay pheromone trails on the components of their chosen solution, depending on the solution’s quality.
* An ant will often follow the strongest pheromone trail when constructing a solution.
* However, for the ant to consider solutions other than the current best, a small amount of randomness is required in its decision process.
* In addition to this, a heuristic value is also computed and considered helping to guide the search process towards the best solutions.
* In the example of the multiple traveling salesman problem this heuristic will typically be the length of the edge between the destinations being considered - the shorter the edge, the more likely an ant will pick it.



**6.Implementation**

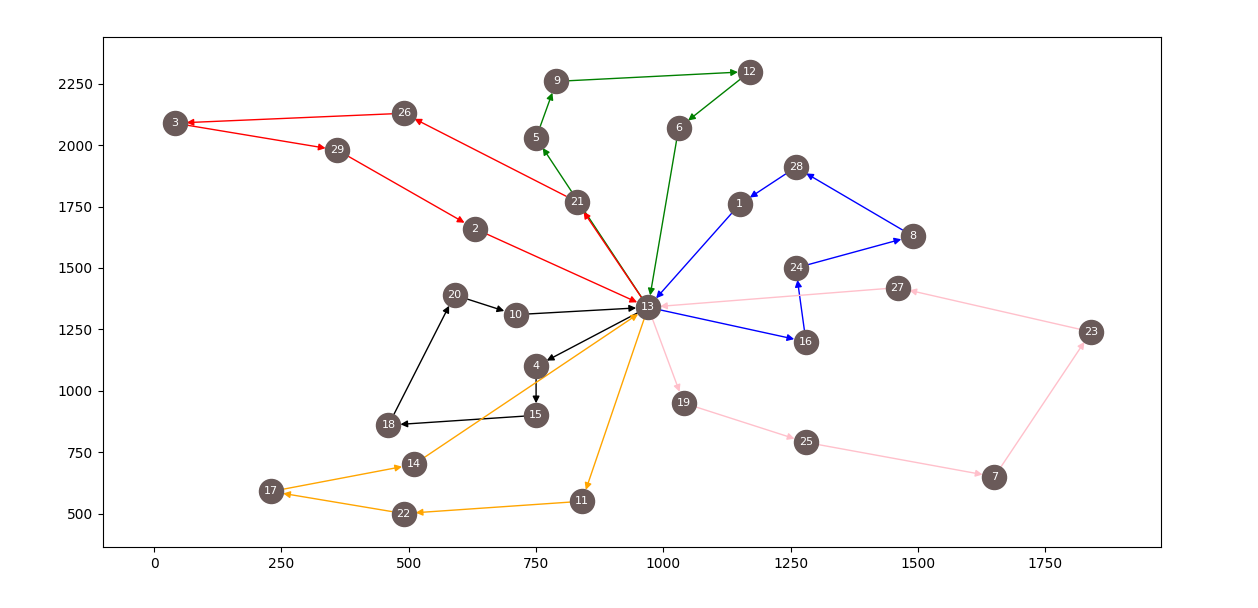
NAME: bays29  
TYPE: TSP  
COMMENT: 29 cities in Bavaria  
DIMENSION: 29  
EDGE\_WEIGHT\_TYPE: EXPLICIT  
EDGE\_WEIGHT\_FORMAT: FULL\_MATRIX   
DISPLAY\_DATA\_TYPE: TWOD\_DISPLAY  
EDGE\_WEIGHT\_SECTION  
 0 107 241 190 124 80 316 76 152 157 283 133 113 297 228 129 348 276 188 150 65 341 184 67 221 169 108 45 167  
 107 0 148 137 88 127 336 183 134 95 254 180 101 234 175 176 265 199 182 67 42 278 271 146 251 105 191 139 79  
 241 148 0 374 171 259 509 317 217 232 491 312 280 391 412 349 422 356 355 204 182 435 417 292 424 116 337 273 77  
 190 137 374 0 202 234 222 192 248 42 117 287 79 107 38 121 152 86 68 70 137 151 239 135 137 242 165 228 205  
 124 88 171 202 0 61 392 202 46 160 319 112 163 322 240 232 314 287 238 155 65 366 300 175 307 57 220 121 97

80 127 259 234 61 0 386 141 72 167 351 55 157 331 272 226 362 296 232 164 85 375 249 147 301 118 188 60 185  
 316 336 509 222 392 386 0 233 438 254 202 439 235 254 210 187 313 266 154 282 321 298 168 249 95 437 190 314 435  
 76 183 317 192 202 141 233 0 213 188 272 193 131 302 233 98 344 289 177 216 141 346 108 57 190 245 43 81 243  
 152 134 217 248 46 72 438 213 0 206 365 89 209 368 286 278 360 333 284 201 111 412 321 221 353 72 266 132 111  
 157 95 232 42 160 167 254 188 206 0 159 220 57 149 80 132 193 127 100 28 95 193 241 131 169 200 161 189 163  
 283 254 491 117 319 351 202 272 365 159 0 404 176 106 79 161 165 141 95 187 254 103 279 215 117 359 216 308 322  
 133 180 312 287 112 55 439 193 89 220 404 0 210 384 325 279 415 349 285 217 138 428 310 200 354 169 241 112 238  
 113 101 280 79 163 157 235 131 209 57 176 210 0 186 117 75 231 165 81 85 92 230 184 74 150 208 104 158 206  
 297 234 391 107 322 331 254 302 368 149 106 384 186 0 69 191 59 35 125 167 255 44 309 245 169 327 246 335 288  
 228 175 412 38 240 272 210 233 286 80 79 325 117 69 0 122 122 56 56 108 175 113 240 176 125 280 177 266 243  
 129 176 349 121 232 226 187 98 278 132 161 279 75 191 122 0 244 178 66 160 161 235 118 62 92 277 55 155 275  
 348 265 422 152 314 362 313 344 360 193 165 415 231 59 122 244 0 66 178 198 286 77 362 287 228 358 299 380 319  
 276 199 356 86 287 296 266 289 333 127 141 349 165 35 56 178 66 0 112 132 220 79 296 232 181 292 233 314 253  
 188 182 355 68 238 232 154 177 284 100 95 285 81 125 56 66 178 112 0 128 167 169 179 120 69 283 121 213 281  
 150 67 204 70 155 164 282 216 201 28 187 217 85 167 108 160 198 132 128 0 88 211 269 159 197 172 189 182 135  
 65 42 182 137 65 85 321 141 111 95 254 138 92 255 175 161 286 220 167 88 0 299 229 104 236 110 149 97 108  
 341 278 435 151 366 375 298 346 412 193 103 428 230 44 113 235 77 79 169 211 299 0 353 289 213 371 290 379 332  
 184 271 417 239 300 249 168 108 321 241 279 310 184 309 240 118 362 296 179 269 229 353 0 121 162 345 80 189 342

67 146 292 135 175 147 249 57 221 131 215 200 74 245 176 62 287 232 120 159 104 289 121 0 154 220 41 93 218

221 251 424 137 307 301 95 190 353 169 117 354 150 169 125 92 228 181 69 197 236 213 162 154 0 352 147 247 350  
 169 105 116 242 57 118 437 245 72 200 359 169 208 327 280 277 358 292 283 172 110 371 345 220 352 0 265 178 39  
 108 191 337 165 220 188 190 43 266 161 216 241 104 246 177 55 299 233 121 189 149 290 80 41 147 265 0 124 263  
 45 139 273 228 121 60 314 81 132 189 308 112 158 335 266 155 380 314 213 182 97 379 189 93 247 178 124 0 199  
 167 79 77 205 97 185 435 243 111 163 322 238 206 288 243 275 319 253 281 135 108 332 342 218 350 39 263 199 0

For implementing mTSP we need data. So, we have considered the distances among the 29 cities in Bavaria and saved it as bays29.tsp.



Now we have imported the following packages in-order to solve the problem:

1. **tsplib95** - TSPLIB 95 is a library for working with TSPLIB 95 files. It easily

reads and writes“. tsp” files. It easily converts problems into

“**Networkx. Graph”** instances.

2. **matplotlib** - Matplotlib is a comprehensive library for creating static,

animated, and interactive visualizations in Python.

3. **Networkx** - NetworkX is a Python package for the creation, manipulation,

and study of the structure, dynamics, and functions of complex

networks.

4. **Itertools** - **This module implements a number of iterator building blocks**

**inspired by constructs from APL, Haskell, and SML.**  In simple

words, the number of iterators can together create 'iterator algebra'

which makes it possible to complete the complex task. The

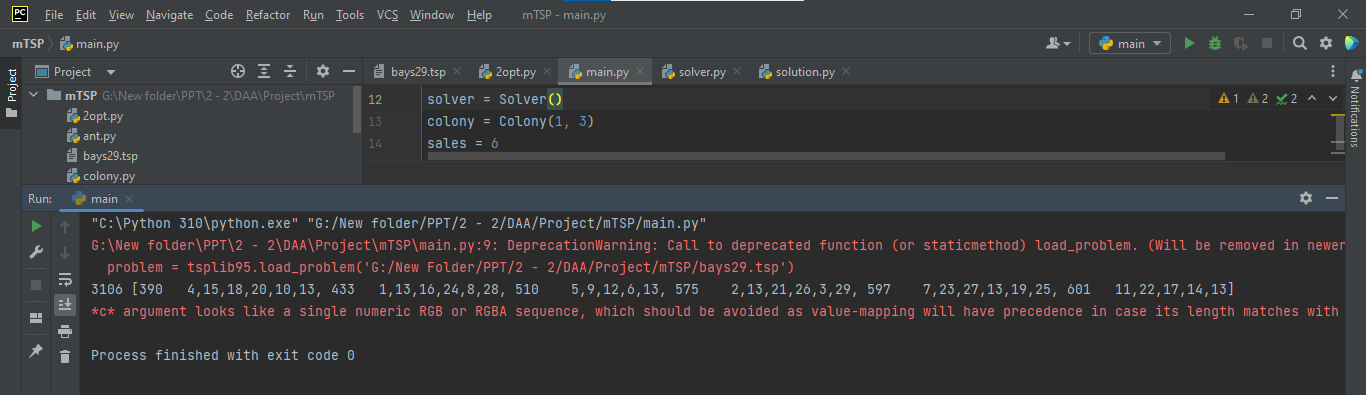
functions in itertools are used to produce more complex iterators.

5. **Bisect** - This module provides support for maintaining a list in sorted order

without having to sort the list after each insertion.

To solve this problem, first we need to install the above packages and import them to our workspace. Now, we need to load the graph and divide the graph into clusters of our choice. After dividing into clusters, we need to calculate the best route using ACO technique and print the solution.

**7.Results Discussion**



As mentioned in the above figure, we have divided the graph into 6 clusters and calculated the cost of each cluster. From the above results we can say that by using ACO we might not get an optimal solution because as the number of clusters increases the individual cost decreases but the total cost increases.

**8.Conclusion and Future Work**

In this project, we have presented a method for solving the multiple traveling salesman problem based on the ant colony optimization. The pheromone values helped a lot to achieve a better solution. The experiment results have shown that the new method has quick convergence speed and can be well applied to find best solutions. In future, we decided to combine multi-objective TSP with parallel processing Parallel processing can improve algorithm speed and reduce algorithm execution time. This method may be greatly improving the speed of finding best solutions.

**9.REFERENCES**

[1] X. Sun, S. Wang, Y. Xia, and W. Zheng, ‘‘Predictive-trend-aware

composition of Web services with time-varying quality-of-service,’’ IEEE

Access, vol. 8, pp. 1910–1921, Dec. 2020.

[2] S. Wang, D. You, and M. Zhou, ‘‘A necessary and sufficient condition for a

resource subset to generate a strict minimal siphon in S 4PR,’’ IEEE Trans.

Autom. Control, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.

[3] S. Pang, M. Wang, S. Qiao, X. Wang, and H. Chen, ‘‘Fault diagnosis for

service composition by spiking neural P systems with colored spikes,’’ Chin.

J. Electron., vol. 28, no. 5, pp. 1033–1040, Sep. 2019.

[4] M. Drigo, V. Maniezzo, and A. Colorni, ‘‘The ant system: Optimization by a

colony of cooperation agents,’’ IEEE Trans. Syst. Man, Cybern. B, Cybern.,

vol. 26, no. 1, pp. 29–41, Feb. 1996.

[5] Y. Feng, M. Zhou, G. Tian, Z. Li, Z. Zhang, Q. Zhang, and J. Tan, ‘‘Target

disassembly sequencing and scheme evaluation for CNC machine tools

using improved multiobjective ant colony algorithm and fuzzy integral,’’

IEEE Trans. Syst., Man, Cybern. Syst., vol. 49, no. 12, pp. 2438–2451, Dec.

2019.

[6] Alves RMF, Lopes CR (2015) Using genetic algorithms to minimize the

distance and balance the routes for the multiple traveling salesman problem.

In: Proceedings Sendai, CEC, pp 1–8

[7] Bektas T 2006 Omega 34 209–19

[8] Russel R A 1977 Oper. Res. 25 517–24s