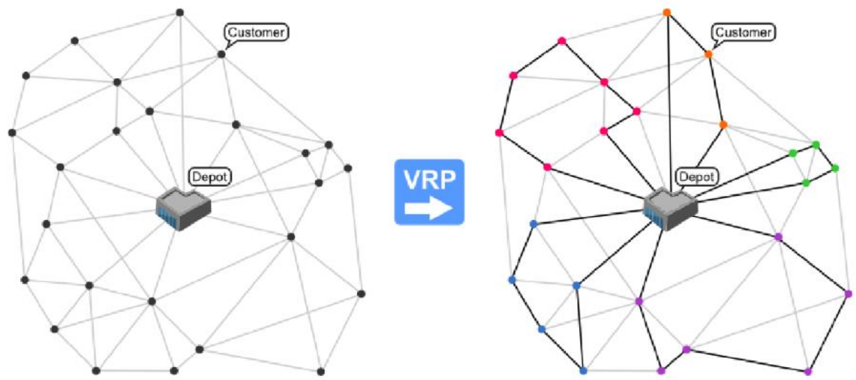
INTRODUCTION:

Vehicle Routing Problem is a generalization of the Travelling Salesman problem. There are nodes which represent customers/delivery spots/end points and a depot. Each customer position is defined on a planar surface and the distance to this point is known. The Vehicle Routing Problem (VRP) is a complex combinatorial optimization problem that belongs to the NP-complete class.

AIM OF THE STUDY:

The VRP can be described as follows: Given a fleet of vehicles with uniform capacity, a common depot, and several customer demands, finds the set of routes with overall minimum route cost which service all the demands.

All the itineraries start and end at the depot and they must be designed in such a way that each customer is served only once and just by one vehicle.



Genetic algorithms provide a search technique used in computing to find true or approximate solution to optimization and search problems. However, we used some heuristic in addition during crossover or mutation for tuning the system to obtain better result which is Genetic K-means algorithm and implemented these two variants' algorithms in code and compared results. Also studied if Particle Swarm Optimization (PSO) could perform better than Genetic algorithm.

DATA COLLECTION PROCESS:

Data points include the location of both the depot and customer locations. A location contains: X – coordinate, Y – coordinate, Demand. Demand at depot is taken zero.

Data used: 36 nodes including depot -

https://drive.google.com/file/d/1mWdgVMsE04vh0MJN9mZeVbQFqhLuYkdm/view?us=sharing

APPROACHES/ METHODS USED:

1. GRAPH THEORY- Nodes are customer locations and numbers on arc represent distance for a delivery person to travel from one node to another and can find shortest path to every node. This can be solved using Dijkstra's or A* algorithm.

Drawbacks: Due to the nature of the problem it is not possible to use exact methods for large instances of the VRP as complexity of using graph theory is n! where n is number of nodes.

2. GENETIC ALGORITHMS - Genetic algorithms have been inspired by the natural selection mechanism introduced by Darwin.

<u>Steps</u> involved in the algorithm:

- Generate Population (Random)
- *Fitness function*
- *Selection* (Based on Fitness Function)
- *Crossover function:* In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is an analogy to reproduction and biological crossover, upon which genetic algorithms are based. Both implemented crossovers don't do mutual exchange of genetic material between two parents. They take information from one individual and insert it in the other to create a new child. The probability which crossover method should be used can be configured.
- *Mutation function*: In genetic algorithms, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation. The probability which mutations will take place and if mutation takes place at all can be configured.

Drawbacks: Depends on selection of initial population, takes more time to converge.

3. GENETIC K-MEANS ALGORITHM - This approach tries to work out the initial population generation shortcoming of the first genetic algorithm approach. The idea is to generate a robust population initially for genetic algorithm to solve the sub-problems generated by clustering. Valid sub-solutions can be formed using clusters made by ordering nodes in radial order, keeping a constraint to not exceed a vehicle's maximum capacity. Now genetic algorithm can be applied to these sub-solutions. Each sub-solution will be represented by a chromosome, where each node will be a gene. Genetic operators (crossover and mutation) will be applied on the chromosome. Apply partially matched crossover and uniform mutation with a pre-decided probability and add children to the population for a given number of generations. Update the sub-solution if and only if the cost of the new sub-solution is lower than the current sub-solution cost. Repeat the process for all sub-solutions.

4. PARTICLE SWARM OPTIMIZATION - The first step of the traditional PSO algorithm is to initialize the particle swarm. Then, the fitness of the particles is calculated. The global optimal solution and local optimal solution are updated according to the fitness. Finally, the velocity and position of the next generation of particles are calculated by updating the velocity and position formula until the maximum number of iterations is reached. The updating formula of the velocity and position of particle *i* is

$$v_i^k = \omega v_i^{k-1} + a_1 r_1 (Pbest_i - y_i^{k-1}) + a_2 r_2 (Gbest_i - y_i^{k-1})$$

$$y_i^k = y_i^{k-1} + v_i^k$$

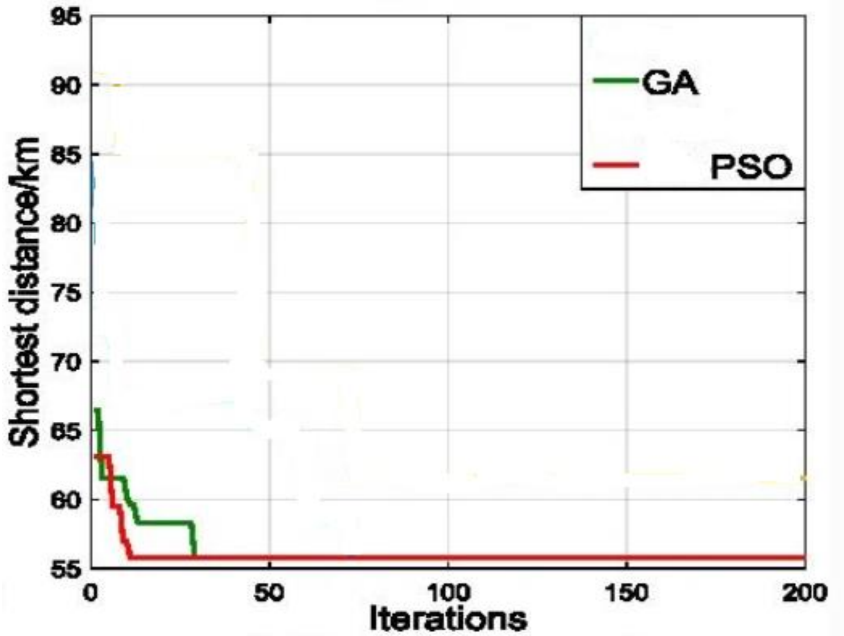
where v_i^k is the component of the velocity vector of particle i in iteration k. y_i^k is component of the position vector of particle i in iteration k. a_1 and a_2 are acceleration constants, which are responsible for adjusting the maximum speed of particle learning. r_1 and r_2 are random functions with values ranging from 0 to 1. *w* is the inertia weight (nonnegative), reflecting the influence of the individual particle history at present. In first Formula, the first part represents the previous velocity of the particle. The second part is the “cognition” part, which represents the distance between the current position of particle i and its historical optimal position, which is equivalent to the local optimal solution. The third part is the “society” part, which represents the distance between the current position of particle i and the optimal position of the population, which is equivalent to the global optimal solution. The final motion direction of particles is affected by the above three parts.

RESULTS:

On using the data from the link in data collection section, the results for the GA algorithms are as below:

Method	**Genetic Algorithm**	**Genetic K-means Algorithm**
Implementation of code:	https://drive.google.com/file/d/1nGnYhkonRwJ9Z7DpRtzRo3fjnzwAHGNM/view?usp=sharing	https://drive.google.com/file/d/1XswghcljgGF5JCMlNWm8NdWO54a5icP/view?usp=sharing
Optimal number of vehicles:	3	5
Optimal distance travelled:	1702 units	956.27 units

Comparison between GA and PSO:



CONCLUSION:

Traditional approaches fail to scale efficiently for large datasets. When we want to dynamically update routes computational time matters. Meta heurists like GA and PSO scale efficiently. Search is needed almost everywhere at some stage. High exploration rate doesn't necessarily generate the best solution.

It was observed that GA when used with clustering give better optimal results. It was also observed that PSO convergence rate is high than GA. GA and PSO are widely used in real life applications for routing problems. Eg.: Amazon delivery.

Many domestic and foreign researchers have proposed some improvement methods. An improved particle swarm optimization combined with a genetic algorithm (GA-PSO) is one. By introducing crossover and variation, the velocity and position updating formula of PSO are improved to increase the diversity of the search space and avoid falling into the local optimal solution.

REFERENCES:

[1]. Ochelska-Mierzejewska, J.; Poniszewska-Mara ´nda, A.; Mara ´nda, W. Selected Genetic Algorithms for Vehicle Routing Problem Solving. Electronics 2021, 10, 3147. https://doi.org/10.3390/electronics10243147

[2]. G. Clarke and J. Wright “Scheduling of vehicles from a central depot to a number of delivery points”, Operations Research, 12 #4, 568-581, 1964.

[3]. M. L. Fisher and R. Jaikumar. “A Generalized Assignment Heuristic for Vehicle Routing”. Networks, 11:109-124, 1981.