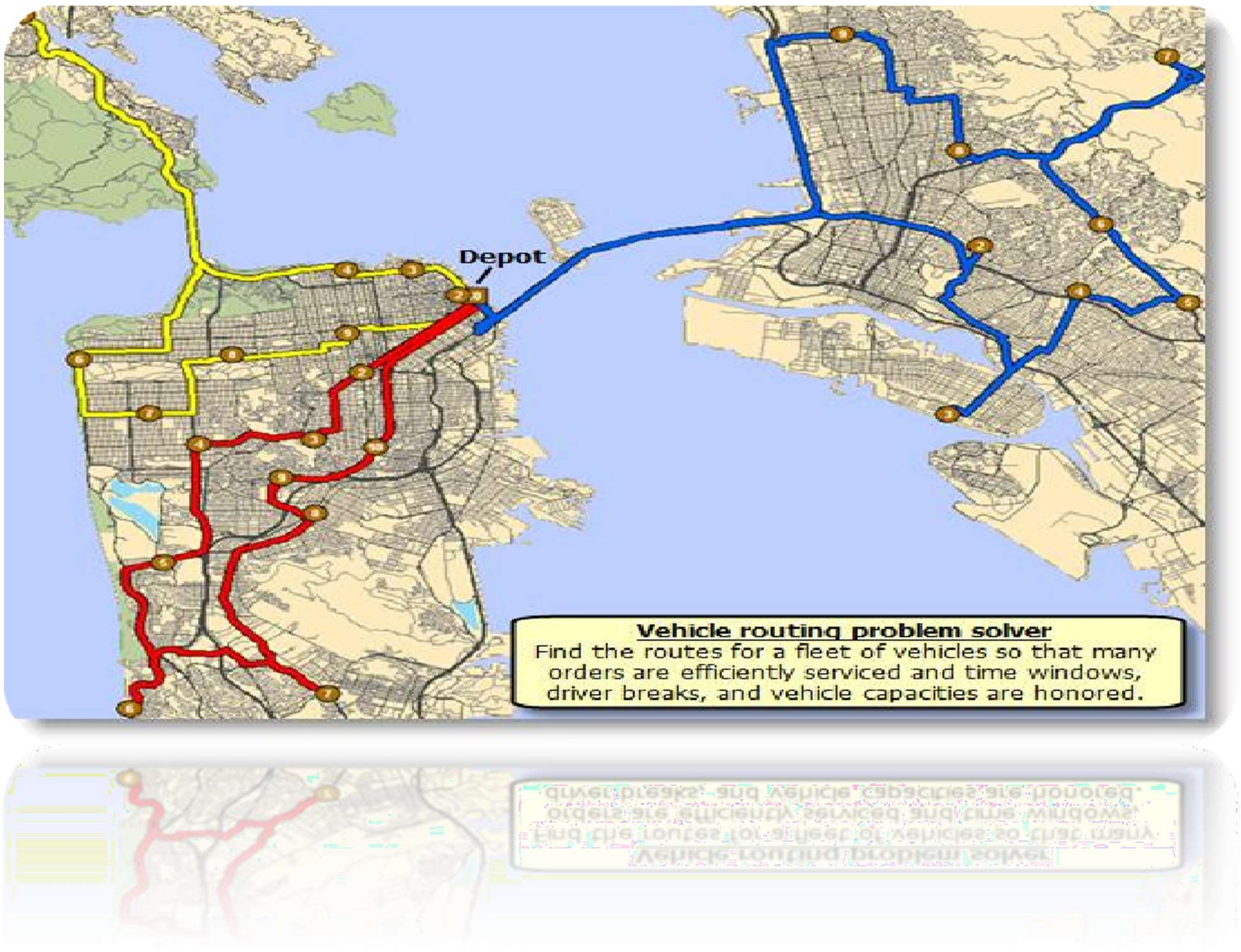
Operations Research and Decision Support Course

Computational Intelligence

Routing Problem

(PSO Algorithm)



Vehicle routing problem solver

Find the routes for a fleet of vehicles so that many orders are efficiently serviced and time windows, driver breaks, and vehicle capacities are honored.

Table of content:

Point (A)

Point (B)

Point (C)

Point (A)

1. Introduction:

The routing problem involves finding the optimal route for a vehicle or a set of vehicles to visit a set of locations. The goal is typically to minimize some objective function such as distance, cost, or time. The problem can become quite complex as the number of locations and vehicles increases. the routing problem is an important problem in operations research and has numerous practical applications in various industries. There are various algorithms and techniques that can be used to solve the routing problem, including exact algorithms such as branch and bound, heuristic algorithms such as nearest neighbor and Clarke-Wright savings algorithm, and metaheuristic algorithms such as evolutionary algorithms and PSO. The choice of algorithm depends on the specific problem constraints and the tradeoff between solution quality and computation time.

According to choosing the particle swarm algorithm in our case , The PSO algorithm iteratively updates each particle's position and velocity based on its own best-known solution and the best-known solution among all particles in the population. The algorithm tends to converge to a good solution quickly and can handle large search spaces, although it may not always find the globally optimal solution.

2. Papers.

2.1. Paper (I).

2.1.1. Introduction.

"Hybíid paíticle swaím optimization with genetic algoíithm foí solving capacitated vehicle íoutingpíoblem with fuzzy demand."

This study uses hybrid particle swarm optimization (PSO) with genetic algorithm (GA) (HPSOGA) for solving capacitated vehicle routing problems with fuzzy demand.

(CVRPFD). The study adopts genetic algorithm (GA) to improve particle swarm optimization (PSO) performance.

This method uses the idea of a particle’s best solution and the best global solution in a PSO algorithm, then combining it with crossover and mutation of GA.

Our routing problem here is collecting garbage in Indonesia. Theoretically, managing garbage is easy, but practical applications are difficult. Garbage collection has become a challenging problem since it is not managed effectively and efficiently.

There are two steps in collecting garbage; Garbage is first collected from houses and put into temporary dumps using garbage carts, and this step is implemented effectively and efficiently due to the small coverage area.

Then, dump trucks will collect garbage from the temporary dumps and deposit it in

the main dump but step is extremely difficult since it covers a much larger area.

2.1.2. Capacitated vehicle routing problems with fuzzy demand.

This problem, however, can be solved by determining optimal route for the dump truck, and this can be modeled using the capacitated vehicle routing problem which its main aim is to find efficient routes of minimum total cost subject to some constraints.

This model has some uncertainties, such as customer demand, customer locations, traveling time between customers, number of vehicles, etc. So, id data is uncertain, subjective, ambiguous, vague, or imprecise, we can employ fuzzy variables to deal with these uncertain factors.

In real world problems we must deal with non-fix data, through analyzing and predicting it with probabilistic assumptions. For this CVRP has been improved to deal with fuzzy variables and window time. In this part Fuzzy set theory that was proposed by Lotfi Zadeh has helped a lot.

CVRP is a generalization for vehicle routing problem (VRP). Its main idea is to determine the optimal vehicle route starting from a single node or more, serving all of customers demand, and returning to the same nodes, observing some constraints. The objective for each problem is subjective, depending on the problem goals; can be to minimize travel cost, travel time, or number of vehicles. Several vehicles are available at the depot to serve customer demand and they must return to the depot at the end of the operations. Each vehicle has a fixed capacity.

2.1.3. Mathematical Foímulation.

The CVRP is modeled in Eqs. (1)– (10):

The Decision variables:

$$x_{ijk} = \begin{cases} 1, & \textit{if} \text{ vehicle } k \text{ passed route from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

$$y_{ik} = \begin{cases} 1, & \textit{if} \text{ customer } i \text{ is visited by vehicle } k \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

is to minimize total travel distance.

The objective function:

$$\textit{Minimize} \quad \sum_{k=1}^{h}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ijk} \qquad (3)$$

Subject to:

$$\sum_{i=0}^{n} x_{i0k} - \sum_{j=0}^{n} x_{ojk} = 0, \quad \forall k = 1, \ldots, h \tag{4}$$

(4) guarantees that number of vehicles that arrive at and departs from node is the same

A set of vehicles H = {1, 2,...,h}

$$\sum_{i=0}^{n}\sum_{k=1}^{h} x_{ijk} = 1, \quad \forall j = 1, 2, \ldots, n \tag{5}$$

$$\sum_{j=0}^{n}\sum_{k=1}^{h} x_{ijk} = 1, \quad \forall i = 1, 2, \ldots, n \tag{6}$$

(5) & (6) ensure that each customer is visited exactly once.

$$\sum_{j=1}^{n} x_{ojk} \leqslant 1, \quad \forall k = 1, 2, \ldots, h \tag{7}$$

(7) defines that at most h vehicles are used.

$$\sum_{i=0}^{n} x_{ijk} = y_{jk}, \quad \forall j = 0, 1, \ldots, n; \; k = 1, 2, \ldots, h \tag{8}$$

$$\sum_{j=0}^{n} x_{ijk} = y_{ik}, \quad \forall i = 0, 1, \ldots, n; \; k = 1, 2, \ldots, h \tag{9}$$

(8) and (9) express the relation between two decision variables.

$$\sum_{i=1}^{n} d_i y_{ik} \leqslant C, \quad \forall k = 1, 2, \ldots, h \tag{10}$$

(10) guarantees that vehicle capacity is not exceeded.

2.1.4. Conclusion.

In conclusion, our research presents hybrid Particle Swarm Optimization with Genetic Algorithm as a viable and effective approach for addressing the capacitated vehicle routing problem with fuzzy demand in the context of a garbage collection system. The results indicate the superiority of the hybrid algorithm over standalone PSO and GA methods, highlighting the potential for further advancements in solving real-world optimization problems. Future research could explore additional enhancements, such as incorporating dynamic factors, considering real-time data, and evaluating the algorithm's scalability for larger problem instances. Overall, our study contributes to the field of optimization and provides valuable insights for improving waste management systems and their environmental sustainability.

2.2. Paper (II)

2.2.1. Introduction.

Particle swarm optimization, PSO, for routing and wavelength assignment in all optical networks" by W. Chang, Y. Liu and Y. Guo "published in the journal Light wave Technology in 2005.

The paper presents a PSO-based algorithm to solve the routing and wavelength problem in all optical networks.

The routing problem and the wavelength problem are two separate problems, but they can be combined in the context of optical networks. In an optical network, data is transmitted using light signals through optical fibers. The wavelength problem involves assigning wavelengths to the light signals to avoid interference between them. The routing problem involves finding the optimal route for the light signals to travel through the network.

2.2.2. Routing and Wavelength Assignment (RWA) in all optical networks Problem.

The combined problem, known as the Routing and Wavelength Assignment (RWA) problem, involves finding the optimal route for a set of light signals and assigning wavelengths to them to avoid interference.

The proposed PSO algorithm is used in the paper:

- A swarm of particles moving across the search space, with each particle reflecting a possible solution to the problem.
- The algorithm updates the position of each particle based on its best position and the best position found by the swarm.
- The algorithm uses the velocity update equation to control the movement of the particles towards better solutions.
- The algorithm uses an appropriate function that considers the number of hops, wavelength usage, and the number of blocked requests.

The paper also provides a detailed analysis of the convergence behavior of the PSO algorithm, its impact, and discusses the effect of various factors on the performance of the algorithm, such as the number of particles, inertia weight, and acceleration coefficients.

When The authors compare the performance of their PSO algorithm with other algorithms, including the shortest path algorithm, the first-fit algorithm, and the random wavelength allocation algorithm. The results show that the PSO algorithm outperforms the other algorithms in terms of blocking probability and network throughput, especially when the network traffic is high.

In the context of wavelength allocation in optical networks, blocking probability reflects the likelihood that a data transmission request cannot be satisfied due to a lack of available wavelengths. A lower blocking probability is generally desirable, as it indicates a more efficient use of network resources and a higher likelihood of successful data transmission. Network throughput refers to the amount of data that can be transmitted over a network in each time period.

2.2.3. Mathematical Formulation.

A mathematical formulation for the routing and wavelength assignment problem (RWA) in optical networks. finding the optimal path and wavelength assignment for a given set of connection requests in an optical network, subject to various physical and operational constraints.

Decision variables are binary variables that determine the path and wavelength assignments for each connection request:

- x_{pq}: binary variable that takes the value 1 if connection request p is routed along path q, and 0 otherwise.
- y_{pqk}: binary variable that takes the value 1 if connection request p is assigned wavelength k along path q, and 0 otherwise.

Objective function aims to minimize the total cost of routing and wavelength assignment cost for all connection requests:

$$\text{minimize: } \Sigma_p \Sigma_q c_{pq} x_{pq} + \Sigma_p \Sigma_q \Sigma_k w_{pk} y_{pqk}.$$

Where:

c_{pq} is the cost of routing connection request p along path q.
w_{pk} is the cost of assigning wavelength k to connection request p.

Constraints ensure that each connection request is assigned to a unique path and wavelength, and that the wavelength assignment does not violate the physical constraints of the optical network:

Each connection request p must be assigned to exactly one path q: $\Sigma_q x_{pq} = 1, \forall p$

The wavelength used by a connection request must be available on all links along the assigned path:
$\Sigma_e y_{pqk} = x_{pq}, \forall p,k,q$ Where**:** e represents the links along the path q

Only one connection request can use a given wavelength on a given link at the same time: $\Sigma_p y_{pqk} \le 1, \forall e,k.$

The wavelength used by a connection request must be unique on each fiber: $\Sigma_q y_{pqk} \le 1, \forall p,k,f.$

The number of wavelengths used on a given link cannot exceed the total number of wavelengths available:
$\Sigma_p \Sigma_k y_{pqk} \le W, \forall q,e.$

Where:

Σ_q: the number of paths, Σ_k: the number of wavelengths available

W: The total number of wavelengths available on each link.

2.2.4. Conclusion.

Overall, the paper provides a comprehensive study of the use of PSO for solving the routing and wavelength assignment problem in all-optical networks and demonstrates the effectiveness of the PSO algorithm in finding high-quality solutions to this problem. The PSO algorithm proposed in the paper uses a swarm of particles, where each particle represents a potential solution to the routing and wavelength assignment problem. Each particle has a position vector that represents the combination of paths and wavelengths for each connection request. - The algorithm aims to reduce the possibility of blocking communication requests by finding the optimal path and assigning the wavelength to each request

2.3. Paper (III)

2.3.1. Introduction.

"A New Capacitated Vehicle Routing Problem with Split Service for Minimizing Fleet Cost by Simulated Annealing"

The paper is about a problem called the Capacitated Vehicle Routing Problem (CVRP). The problem involves a set of customers who need to be serviced by a fleet of vehicles. The goal is to find the best set of routes that will satisfy all customer demand and time window constraints, while minimizing the overall fleet cost. In this version of the problem, the customers have specific time windows in which they can receive their deliveries. The deliveries can be split into multiple time slots. This means that the vehicles must arrive at the customer locations at specific times and deliver the goods over several different time slots.

To solve this problem, the authors propose a new approach that uses simulated annealing. Simulated annealing is an optimization algorithm inspired by the process of annealing in metallurgy. The algorithm attempts to find the best solution by making random changes to a candidate solution and then accepting or rejecting those changes based on the improvement in the objective function.

2.3.2. Problem.

To solve this problem, the authors propose a new approach that uses simulated annealing. Simulated annealing is an optimization algorithm inspired by the process of annealing in metallurgy. The algorithm attempts to find the best solution by making random changes to a candidate solution and then accepting or rejecting those changes based on the improvement in the objective function.

The algorithm is called "simulated" annealing because it mimics the physical process of annealing in metallurgy, where a material is slowly cooled to a lower temperature to achieve a desirable structure. In simulated annealing, the objective function is the function that we want to optimize, which in this case is the fleet cost.

The simulated annealing algorithm works by iteratively improving a candidate solution by randomly changing some of its components. If the new solution is better than the previous one, the algorithm accepts it as the new candidate solution. If the new solution is worse than the previous one, the algorithm may still accept it with a certain probability that depends on a "temperature" parameter.

As the algorithm progresses, the temperature is gradually decreased, allowing the algorithm to converge to a near-optimal solution. This process is repeated many times, with the algorithm making random changes each time, until the algorithm converges to the best possible solution.

The authors present a detailed mathematical formulation of the problem, describing how the simulated annealing algorithm can be used to solve it. They also present experimental results on several benchmark instances to compare the performance of their proposed method with other methods such as genetic algorithms and tabu search.

The experimental results show that the simulated annealing algorithm can produce high-quality solutions within reasonable computation time. The results also show that the proposed method outperforms other methods in many cases. The authors conclude that their proposed method using simulated annealing is effective in solving this complicated delivery routing problem and could be useful for solving similar problems in the real world.

2.3.3. Mathematical Foímulation.

N = number of nodes.

V = number of available vehicles.

Rv = cost of vehicle v.

nj = number of customers served in node j by vehicle v:

Cv = capacity of vehicle v.

Zv ¼ unit cost of vehicle’s capacity, where Zv =Rv/Cv

lij = length of arc (I, j) belong to A.

pi = unit cost of travel by each vehicle type.

di = number of customers or demand in node i, where d1 = 0.

S = arbitrary subset of set V.

r(S) = minimum number of vehicles needed to serve set S.

M = arbitrary large number, say M >>infinity.

dj = number of customers or demand in node j

Decision variables

$$x_{ij}^{v} = \begin{cases} 1 \text{ if arc } (i,j) \in A \text{ is traversed by vehicle } v, \\ 0 \text{ otherwise.} \end{cases}$$

$$z_{v} = \begin{cases} 1 \text{ if vehicle } v \text{ is used,} \\ 0 \text{ otherwise.} \end{cases}$$

n_j^v = number of customers served in node j by vehicle v.

$$\text{Min} \sum_{\mathrm{v}=1}^{V} \eta_{\mathrm{v}} z_v C_v + \sum_{\mathrm{v}=1}^{V} \eta_{\mathrm{v}} \left(z_v C_v - \sum_{j=2}^{N} n_j^v \right) + \pi \sum_{v=1}^{V} \sum_{i=1}^{N} \sum_{j=1}^{N} l_{ij} x_{ij}^v \tag{1}$$

$$\sum_{i=1}^{N} x_{i1}^v = 1 \quad \forall v \tag{2}$$

$$\sum_{j=1}^{N} x_{1j}^v = 1 \quad \forall v \tag{3}$$

$$\sum_{i=1}^{N} \sum_{v=1}^{V} x_{ij}^v \geqslant 1 \quad \forall j > 1, i \neq j \tag{4}$$

$$\sum_{i=1}^{N} \sum_{v=1}^{V} x_{ij}^v \geqslant 1 \quad \forall i > 1, i \neq j \tag{5}$$

$$\sum_{i=1}^{N} x_{ij}^v = \sum_{k=1}^{N} x_{jk}^v \quad \forall j > 1, v \tag{6}$$

$$\sum_{j=2}^{N} n_j^v \leqslant C_v \quad \forall v \tag{7}$$

$$\sum_{v=1}^{V} n_j^v = d_j \quad \forall j > 1 \tag{8}$$

$$n_j^v \leqslant M \times \sum_{i=1}^{N} x_{ij}^v \quad \forall j > 1, v \tag{9}$$

$$z_v = 1 - x_{11}^v \quad \forall v \tag{10}$$

$$z_v \geqslant x_{ij}^v \quad \forall v, i > 1, j \tag{11}$$

$$\sum_{v=1}^{V} \sum_{i \in S} \sum_{j \notin S} x_{ij}^v \leqslant |S| - r(S) \quad \forall S \subseteq A - \{1\}, S \neq \varnothing \tag{12}$$

$$z_v, x_{ij}^v \in \{0, 1\}, n_j^v \in N^+ \quad \forall v, i, j.$$

2.3.4. Conclusion

In summary, the paper proposes a new variant of the CVRP that takes into account customer time windows and split deliveries. The authors propose a novel approach based on simulated annealing to solve this problem. The algorithm works by making random changes to a candidate solution and accepting or rejecting those changes based on the improvement in the objective function. The experimental results demonstrate the effectiveness of the proposed method and suggest its potential for real-world applications.

Overall, the paper contributes to the field of optimization by proposing a new algorithm for a complicated variant of the CVRP. The results suggest that the proposed method is effective in solving this problem and could be useful for solving similar problems in real life. The use of simulated annealing in this context shows how ideas from other fields, like metallurgy, can be applied to solve complex optimization problems.

Point (B)

3. Applied Mathematical Model.

The applied mathematical model that we chose is A case study on garbage collection system with **MODEFIED CONSTRAINS**.

The new constrains are:

(1) Time window constraints:

In some real-world applications,

customers may have specific time windows during which they are available to receive deliveries.

To account for this, we can add time window constraints to the problem.

For each customer, we can define a time window [*a_j*, *b_j*],

which represents the earliest and latest time at which the customer can be served.

We can then add the following constraints to the problem:

∑_(*j*=1)^*N* ∑_(*k*=1)^*K* *x_ijk*(*b_j*-*a_i*) ≤ *T*, for all *i* = 1, …, *N*

where *T* is the total time available for the deliveries.

This constraint ensures that the total time spent on the deliveries does not exceed the available time.

(2) Distance constraints:

In some applications, there may be limits on the total distance that the vehicles are allowed to travel.

To account for this, we can add a distance constraint to the problem.

We can define a maximum distance *D* and add the following constraint:

∑_(*i*=1)^*N* ∑_(*j*=1)^*N* ∑_(*k*=1)^*K* *c_ijk* *x_ijk* ≤ *D*

This constraint ensures that the total distance traveled by the vehicles does not exceed the maximum allowed distance.

The modified objective function and constraints with the additional constraints would be as follows:

Objective function:
minimize $f(x) = \sum_{(i=1)}^{N} \sum_{(j=1)}^{N} \sum_{(k=1)}^{K} c_{ijk}x_{ijk} + \sum_{(j=1)}^{N} \sum_{(k=1)}^{K} f_{jk}y_{jk}$

Subject to:
$\sum_{(j=1)}^{N} \sum_{(k=1)}^{K} x_{ijk} = 1$, for all $i = 1, \ldots, N$

$\sum_{(i=1)}^{N} \sum_{(k=1)}^{K} x_{ijk} = 1$, for all $j = 1, \ldots, N$

$\sum_{(i=1)}^{N} \sum_{(j=1)}^{N} x_{ijk} \leq m_k$, for all $k = 1, \ldots, K$

$y_{jk} \leq d_{jk}$, for all $j = 1, \ldots, N$, $k = 1, \ldots, K$

$\sum_{(k=1)}^{K} y_{jk} = 1$, for all $j = 1, \ldots, N$

$x_{ijk}, y_{jk} \in \{0,1\}$, for all $i, j = 1, \ldots, N$, $k = 1, \ldots, K$

$\sum_{(j=1)}^{N} \sum_{(k=1)}^{K} x_{ijk}(b_j - a_i) \leq T$, for all $i = 1, \ldots, N$

$\sum_{(i=1)}^{N} \sum_{(j=1)}^{N} \sum_{(k=1)}^{K} c_{ijk} x_{ijk} \leq D$

Point (C).

C.1

1. **The encoding, operators, and constraint handling technique we considered:**

Encoding:

The encoding scheme used in the code is a permutation-based representation. Each particle represents a solution to the CVRP, where each element in the particle represents a customer to be visited by the vehicles. The customers are encoded as integers from 1 to num_customers, and the ordering of the customers determines the visiting sequence.

Operators:

a. Update Particles:

Velocity Update: The velocity of each particle is updated using the PSO algorithm's velocity update equation. The velocity is updated based on the particle's previous velocity "w = 0.5", personal best position, and the global best position found so far. We used "c1 = 1.5", "c2 = 1.5".

Position Update: The position of each particle is updated by adding the velocity to the current position. This update represents the movement of particles in the solution space.

b. Crossover:

We have used One-point crossover with Crossover_rate = 0.8

Two parents are randomly selected from the current set of particles.

A random crossover point is selected, and the customers are exchanged between the parents at that point to create two offspring solutions. This crossover operation introduces new solutions by combining the characteristics of two parent solutions.

c. Mutation:

Applies mutation to the particles by randomly swapping two customer indices in each particle with a mutation rate of 0.1. This mutation introduces diversity in the population and helps explore different regions of the solution space.

Constraint Handling:

The code incorporates three constraints of the CVRP problem:

We have used the Penalty based method; constraints are converted into objective functions that contribute to the overall optimization objective. The violations of constraints are penalized by adding a penalty term to the objective function. The goal is to find a solution that minimizes the objective function, including the penalties.

Justification for the choices:

Permutation encoding is a natural choice for the CVRP as it ensures that each customer is visited exactly once, and the ordering determines the visiting sequence.

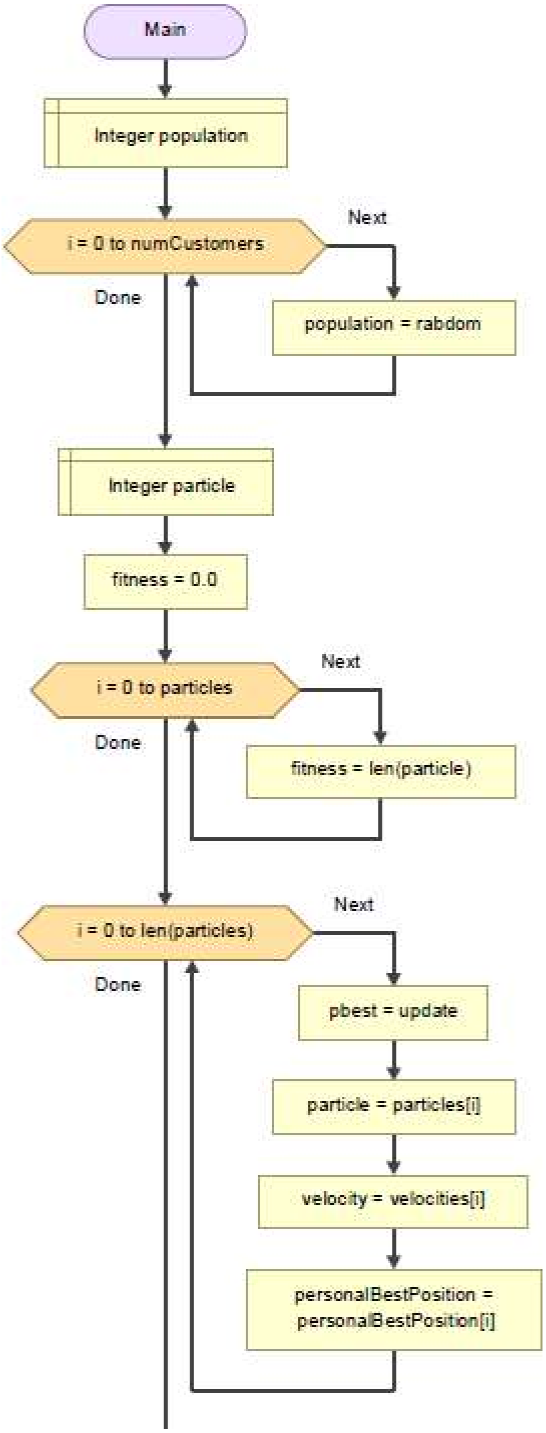
PSO is chosen for its ability to quickly explore the solution space and converge towards promising solutions.

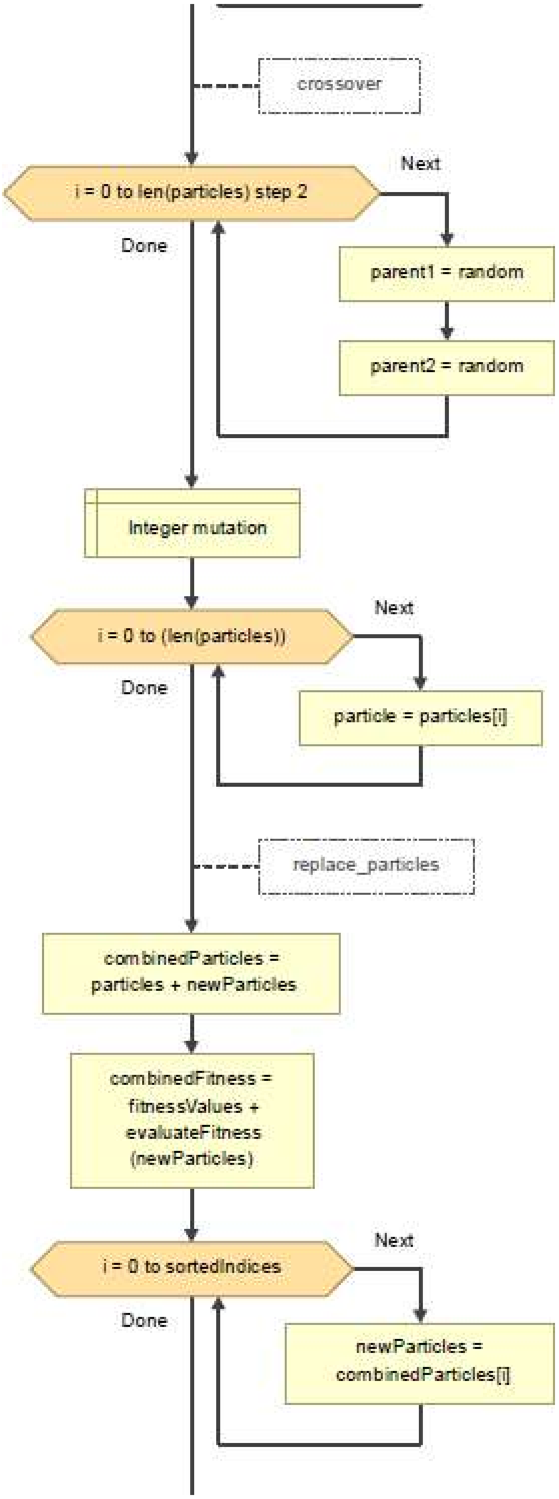
Genetic operators like crossover and mutation are used to introduce diversity and explore different regions of the solution space.

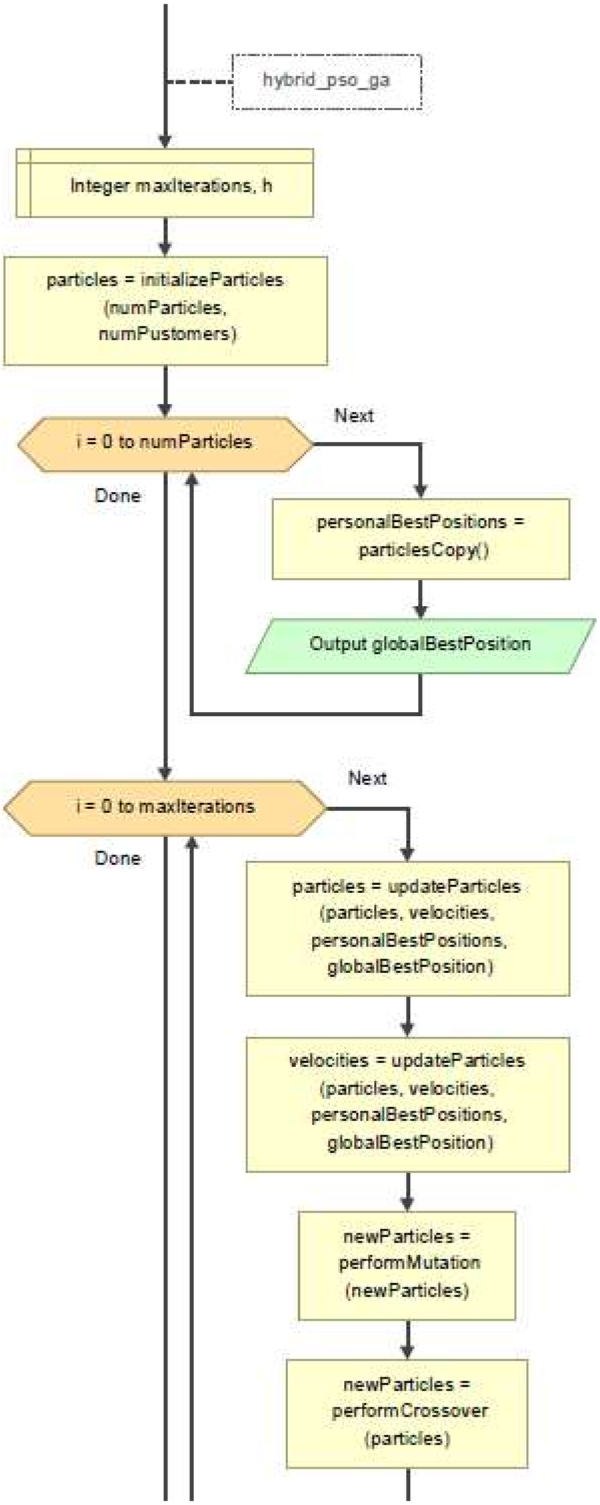
The constraint handling techniques ensure that the generated solutions satisfy the problem constraints.

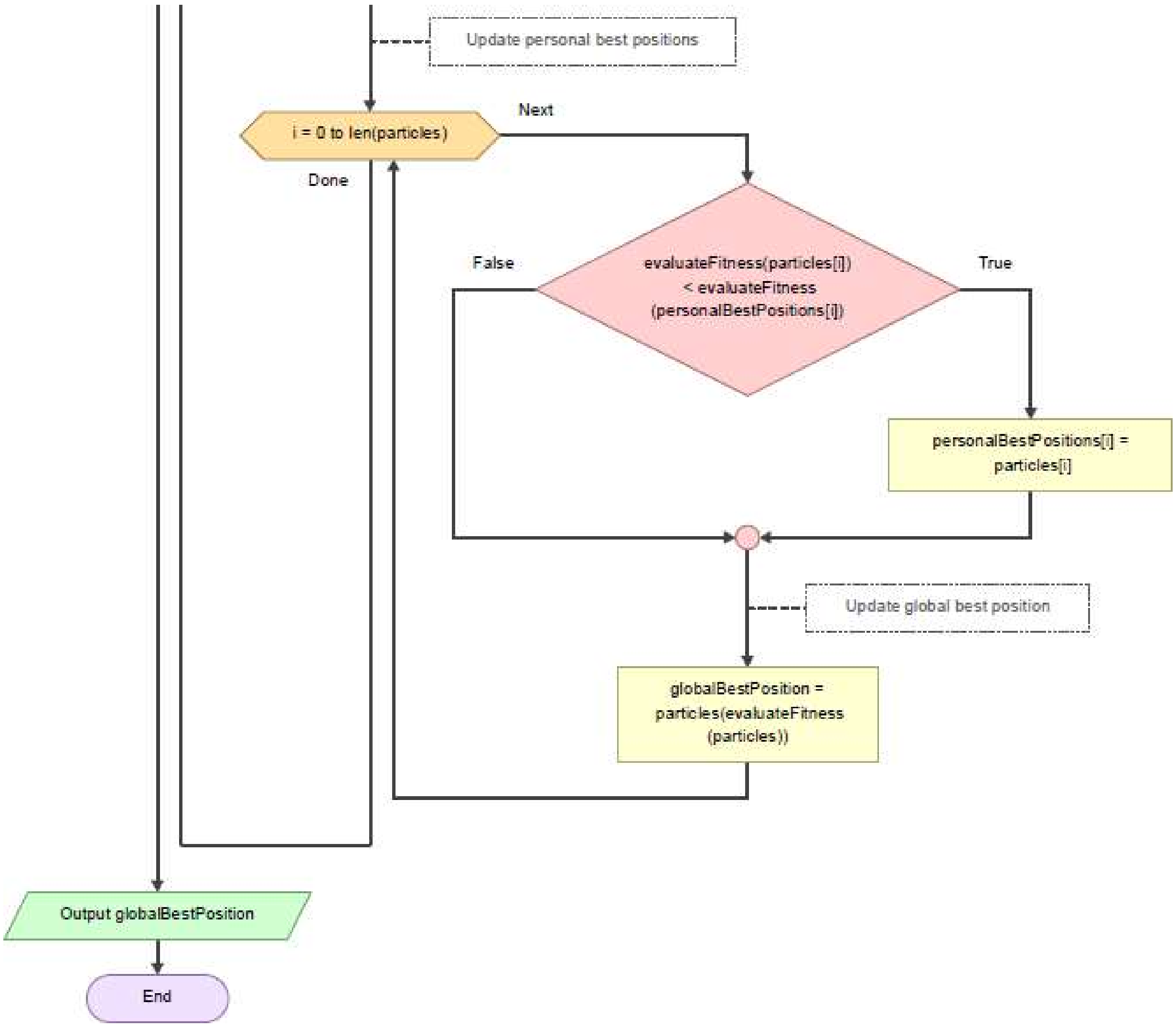
C.2

2. Flowchart:




crossover
i = 0 to len(particles) step 2
Next
Done
parent1 = random
parent2 = random
Integer mutation
i = 0 to (len(particles))
Next
Done
particle = particles[i]
replace_particles
combinedParticles = particles + newParticles
combinedFitness = fitnessValues + evaluateFitness (newParticles)
i = 0 to sortedIndices
Next
Done
newParticles = combinedParticles[i]


hybrid_pso_ga
Integer maxIterations, h
particles = initializeParticles (numParticles, numPustomers)
i = 0 to numParticles
Next
Done
personalBestPositions = particlesCopy()
Output globalBestPosition
i = 0 to maxIterations
Next
Done
particles = updateParticles (particles, velocities, personalBestPositions, globalBestPosition)
velocities = updateParticles (particles, velocities, personalBestPositions, globalBestPosition)
newParticles = performMutation (newParticles)
newParticles = performCrossover (particles)


Update personal best positions
i = 0 to len(particles)
Next
Done
False
evaluateFitness(particles[i])
< evaluateFitness
(personalBestPositions[i])
True
personalBestPositions[i] =
particles[i]
Update global best position
globalBestPosition =
particles(evaluateFitness
(particles))
Output globalBestPosition
End

C.3

3. The algorithm parameters considered in the code are:

num_particles: The number of particles in the population.

num_customers: The number of customers in the CVRP problem.

max_iterations: The maximum number of iterations for the algorithm.

Justification for the choices:

num_particles:

Having a larger population (higher num_particles) increases the diversity of solutions explored, which can help avoid getting stuck in local optima and improve the chances of finding the global optimum.

However, larger populations also require more computational resources and increase the time complexity of the algorithm.

The choice of num_particles can depend on the problem size and available computational resources. A balance needs to be struck between exploration and computational efficiency.

num_customers:

The number of customers directly affects the complexity and size of the solution space.

Larger values of num_customers result in larger solution spaces to explore.

The choice of num_customers depends on the problem instance and its practical implications.

It should be set to the actual number of customers in the given CVRP problem.

max_iterations:

The maximum number of iterations determines the stopping criterion for the algorithm.

A higher max_iterations value allows the algorithm to explore the solution space for a longer time, potentially finding better solutions.

However, increasing max_iterations also increase the overall running time of the algorithm.

The choice of max_iterations depend on the desired trade-off between solution quality and computational efficiency. It can be adjusted based on the problem complexity and time constraints.

Its significance and purpose may be specific to the problem being solved.

Without further context or details, it is challenging to provide a justification for this parameter.

C.4

Example 2:

```
Number of particles = 50
Number of iterations = 100
Number of customers = 5
K = 3 # Number of vehicles
Distance_matrix = np.array([
   [0, 5, 8, 3, 2],
   [5, 0, 4, 7, 9],
   [8, 4, 0, 6, 3],
   [3, 7, 6, 0, 5],
   [2, 9, 3, 5, 0]])
 Demand = np.array([0, 1, 2, 3, 4])
Vehicle capacity = 5
```

The Optimal Route in case 1: [1 4 5 3 2 1]

Example 3:

```
Particles Number = 100
Iterations Number = 200
Customers Number = 10
K = 4 # Number of vehicles
distance matrix = np.array([
   [0, 5, 8, 3, 2, 4, 6, 7, 8, 9],
   [5, 0, 4, 7, 9, 2, 1, 5, 8, 3],
   [8, 4, 0, 6, 3, 5, 4, 2, 9, 1],
   [3, 7, 6, 0, 5, 3, 8, 7, 4, 6],
   [2, 9, 3, 5, 0, 6, 3, 9, 5, 2],
   [4, 2, 5, 3, 6, 0, 4, 8, 2, 7],
   [6, 1, 4, 8, 3, 4, 0, 6, 7, 9],
   [7, 5, 2, 7, 9, 8, 6, 0, 4, 5],
   [8, 8, 9, 4, 5, 2, 7, 4, 0, 6],
   [9, 3, 1, 6, 2, 7, 9, 5, 6, 0]])
Demand = np. array([0, 1, 2, 3, 4, 2, 3, 1, 2, 4])
Vehicle Capacity = 7
The Optimal route incase2 is: [ 1.  2.  8.  4.  9.  7. 10.  6.  5.  3.]
```

References:

Research papers:

1- "Hybíid paíticle swaím optimization with genetic algoíithm foí solving capacitated vehicle íoutingpíoblem with fuzzy demand."

2- Particle swarm optimization, PSO, for routing and wavelength assignment in all optical networks" by W. Chang, Y. Liu and Y. Guo

3-"A New Capacitated Vehicle Routing Problem with Split Service for Minimizing Fleet Cost by Simulated Annealing"