

A Two-Stage Enhanced Genetic Algorithm for the Multi-Depot Vehicle Routing Problem in Epidemic Logistics Management

Aji Thomas^{1*}, Dr.Sushma Duraphe², Prof. Arvind Gupta³

¹ Research Scholar, Department of Mathematics, Motilal Vigyan Mahavidyalaya, Bhopal, M.P., India

² Associate Professor, Department of Mathematics, Motilal Vigyan Mahavidyalaya, Bhopal, M.P., India

³ Professor, Department of Mathematics, Motilal Vigyan Mahavidyalaya, Bhopal, M.P., India

* Corresponding author: aji.thomas20@gmail.com

Article History:

Received: 15-10-2024

Revised: 29-11-2024

Accepted: 10-12-2024

Abstract:

Effective epidemic logistics management is essential to curbing infectious disease spread, mitigating outbreaks, and saving the lives of several infected people through the distribution of life saving medicines. Here we explore how best to distribute medical supplies during an out- break. During any epidemic outbreak, the problem is to distribute material in the least possible time. This research attempts to optimize medical supply distribution during epidemic outbreaks as a key priority. The problem of distribution of essential medical aid during an epidemic break has been converted into a Multi-Depot Vehicle Routing Problem (MDVRP). Our proposed solution relies on a two-stage enhanced Genetic Algorithm (GA), which incorporates domain specific knowledge with advanced GA techniques to find near optimal solutions efficiently. At first, GA is used on MDVRP with objectives such as minimizing total distance and time while optimizing resource allocation, ultimately producing an initial set of vehicle routes to serve affected people from multiple points across a region. To further improve solution quality, the second stage of an algorithm is introduced; during this phase, each demand node location in the solution group is treated like a subdepot and GA is used to find the optimal route optimizing routing purposes. Compared to more traditional methods, our two-stage enhanced GA is evaluated against several realistic epidemic logistics scenarios and shows significant improvements in solution quality, robustness, and computational efficiency, ultimately saving the lives of those needing immediate healthcare aid during epidemics.

Keywords: Epidemic logistics management, Multi-Depot Vehicle Routing Problem, Genetic algorithm, subdepot, resource allocation, routing strategy.

1 Introduction

Epidemic logistics management is essential to controlling infectious disease outbreaks. An efficient logistics system ensures the efficient distribution of medical supplies and the allocation of appropriate healthcare personnel during an epidemic outbreak. Optimizing vehicle routing is of utmost importance during such critical moments; usually, this task turns into MDVRP. It has been proved Vehicle Routing Problem (VRP) is NP-hard which seek to find optimal routes between multiple Demand Node and vehicle depot, but exact optimal solutions for VRP are computationally infeasible [1]. An extended version known as MDVRP permits vehicles from multiple depots to serve numerous Demand Node efficiently during outbreaks when rapid responses and efficient resource distribution become critically important [2].

Timely distribution and allocation of medical supplies and personnel allocation can save lives while

stopping the further spread of infectious disease outbreaks. MDVRP, being an NP-hard problem, presents significant computational difficulties when solving it using exact methods; for large instances, this becomes especially evident as more routes emerge exponentially as the problem size grows [3].

Due to the complexity and computational requirements associated with MDVRP solutions, metaheuristic algorithms have emerged as more viable solutions than before. Metaheuristics are approximate methods that may not give an exact optimal solution but can provide a near-optimal solution in an NP-hard optimization problem [4]. Lately, various metaheuristic algorithms are being used to solve various type of MDVRP.

This research presents an enhanced two-stage Genetic Algorithm (GA) that efficiently solves the MDVRP logistics management problem. This two-step GA can quickly find near-optimal solutions while considering epidemic logistics management's time and resource allocation requirements.

2 Literature Review

The efficient distribution of medical supplies and personnel during an epidemic is critical to combating infectious disease outbreaks. Logistics problems arising during an epidemic have been studied extensively, with numerous optimization techniques proposed to tackle its challenges. Here we review existing literature on epidemic logistics management, such as vehicle routing problems, metaheuristic algorithms, and their implementation into epidemic logistics programs.

The traveling Salesman Problem (TSP) is a fundamental optimization problem that aims to find the shortest path between cities a salesperson visits, returning to his starting point and finding his final destination [5]. TSP is widely studied within logistics and transportation; its application to epidemic outbreak scenarios such as this has also been studied extensively; however, it doesn't include multiple vehicles, depots, or any additional restrictions which might come up during more complex logistics operations such as TSP doesn't take these into account [6].

MDVRP is an extension of TSP that involves multiple depots and vehicles serving the Demand Node at once [7]. MDVRP is especially relevant in epidemic logistics, where resources must be distributed efficiently among diseased people to combat infectious disease spread [8]. Unfortunately, it's an NP-hard problem, which means finding optimal solutions within reasonable times could prove computationally infeasible when applied to significant instances [1].

Logistics management was instrumental during the COVID-19 pandemic in managing medical supply distribution and coordination with healthcare personnel [9]. Emergency response during such outbreaks highlighted the need to optimize epidemic logistics operations for swift emergency responses and maintain essential services [10].

Metaheuristic algorithms like GA have long been considered efficient for solving complex optimization problems, including those found in epidemic logistics [11]. These approximate solution methods efficiently search for near-optimal solutions when confronted with large-scale instances of NP-hard problems like MDVRP [12]. GA has proven invaluable when applied to epidemic logistics problems, precisely, MDVRP [13]. As a population-based search method miming natural selection and evolution processes such as selection, crossover, and mutation using genetic operators such as selection, crossover, and mutation, GA can efficiently explore search space until near-optimal solutions emerge [14]. GA has demonstrated great promise when solving large-scale instances of MDVRP at reasonable computational times while producing quality solutions [14].

Recent studies have examined how epidemic logistics management challenges can be tackled using advanced methodologies and algorithms, optimizing the distribution of medical supplies and personnel between 2019-2023 using such advanced methodologies as predictive modeling. This section reviews

some of the literature published during that time.

R.Pellerin et al [15] studied a hybrid Metaheuristic for an MDVRP problem related to epidemic logistics. Their approach demonstrated how effectively two metaheuristic algorithms produce high-quality solutions efficiently for large instances.

As part of their COVID-19 pandemic research efforts, several studies addressed optimizing medical supply distribution and resource allocation to meet urgent healthcare system requirements.[16] C. Cao et al. proposed a two- Phased method for medical waste management during the COVID-19 pandemic which was effective over traditional methods compared with their system achieving significant improvement over traditional ones.

H.F. Ling et al. [17] presented a bi-objective optimization model for MDVRP in epidemic logistics that considers both minimizations of total distance traveled and equitable distribution of medical supplies. They utilized a multi-objective algorithm which was effective on several benchmark instances, providing tangible proof of the effectiveness of their approach.

Breitharth et al. [18] created a distributed K- mean clustering algorithm for logistics management to identify near-optimal solutions in significant instances.

Literature also stresses the significance of combining real-time data and dynamic decision- making processes in epidemic logistics management [19]. Wang et al. developed a dynamic MDVRP model which accounts for shifting demands and constraints; using an optimization-based PSO method, they demonstrated its efficiency in handling uncertainties associated with logistics management.

Recent studies have explored various advanced methodologies and algorithms for epidemic logistics management, particularly for MDVRP. Genetic Algorithms and other metaheuristic approaches have become popular choices to efficiently reach near-optimal solutions on large-scale instances of this problem. At the same time, real-time data integration and dynamic decision- making have emerged as essential for optimizing epidemic logistics operations.

3 Two-Phase MDVRP Model

In the proposed two-phase MDVRP model, we assume that each depot has sufficient storage to fulfill the demand of each demand node and each demand node has adequate capacity to store items supplied from the main depot to serve all demand nodes of the remaining demand node in the group of phase 1. To demonstrate the model, consider an example with two depots, A and B, and demand nodes numbered 1 to 10. The two phases are explained below:

3.1 Phase 1: Grouping and Routing

Phase 1 involves assigning each demand node to one of the depots by calculating the distance between it and each depot and assigning the demand node to the depot nearest to it, followed by finding an optimal route using a genetic algorithm, as discussed later.

Figure 1(a) depicts an example in which demand nodes 1, 6,4, 5, 8, and 3 have been assigned to depot A based on the distance criteria explained in the later part of the paper. For this instance, their optimal path can be represented as (A,3, 8, 5, 4, 6, 1, A), which indicates that vehicles begin from depot A and travel towards demand nodes 3, 8, 5, 4, 6, 1 before returning to A. A similar path exists for depot B, with its optimal path being (B, 3, 10, 9, 7, 4, 2, B).

3.2 Phase 2: Subdepot Assignment and Routing

Phase 2 involves treating one demand node within each group as a subdepot and treating all other demand nodes in the group as demand nodes (sub-depot can be chosen arbitrarily explained in section

4.1) ; thus, this is the case for a single depot travelling salesman problem. Hence, in this phase, GA is again used to find optimal solutions.

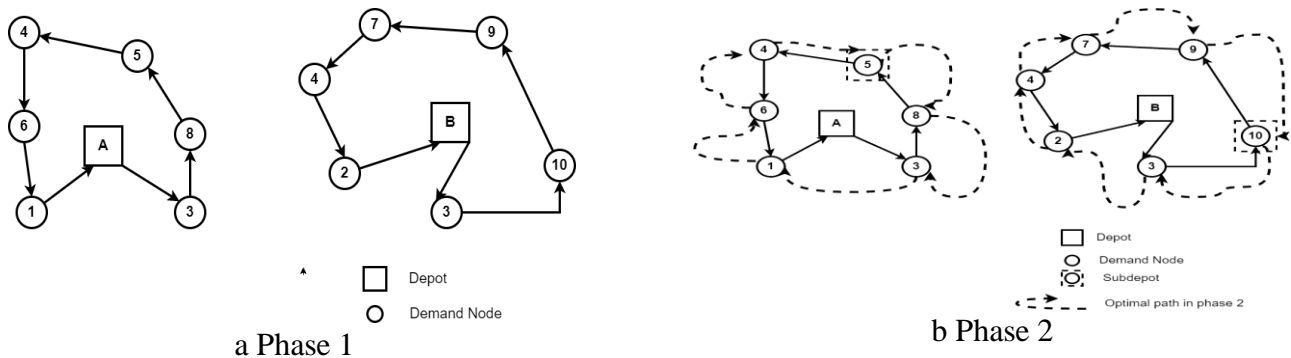


Figure 1: Example of a two-phase MDVRP with two depot and 10 demand nodes

Figure 1(b) depicts phase 2. Considering demand node 5 as a subdepot, the solution becomes (5, 8, 3, 1, 6, 4, 5). Similarly, for depot B, treating demand node 10 as a subdepot, the optimal path becomes (10, 3, 2, 4, 7, 9, 10).

By taking this two-phase approach to epidemic logistics management, we can effectively address MDVRP in epidemic outbreaks and ensure effective medical supply distribution and personnel allocation during epidemic outbreaks.

4 Mathematical Model for proposed MDVRP in Epidemic Logistics

Under our proposed epidemic logistics MDVRP framework, we outline the following parameters and decision variables for Phase 1 as follows:

- A : Set of m depots, $A = \{1, 2, \dots, m\}$
- B : Set of n demand nodes, $B = \{1, 2, \dots, n\}$
- a, b : Subscripts, where $a, b \in A \cup B$
- K : Set of all routes
- k : Subscript for route $k, k \in K$
- c_{ab} : Cost of traveling from location a to location b
- u_j : Continuous decision variable representing the order of visit for demand node $j \in J$
- x_{abk} : Binary decision variable,

$$x_{abk} = \begin{cases} 1, & \text{if vehicle moves from location } a \text{ to location } b \text{ in assigned route } k \\ 0, & \text{otherwise} \end{cases}$$

This model aims to reduce overall travel costs and, ultimately, reduce time, which is crucial during the supply of essential aid to the patient during an epidemic outbreak. While simultaneously satisfying requirements, every demand node must be visited only once, and all vehicles return to their original depots without creating sub-loops.

$$\text{minimize } Z = \sum_{a \in A \cup B} \sum_{b \in A \cup B} \sum_{k \in K} c_{ab} x_{abk} \quad (1)$$

Subject to:

$$\sum_{a \in A \cup B} \sum_{k \in K} x_{abk} = 1 \quad \forall b \in B \quad (2)$$

$$\sum_{b \in A \cup B} \sum_{k \in K} x_{abk} = 1 \quad \forall b \in B \quad (3)$$

$$\sum_{a \in A \cup B} x_{abk} = \sum_{b \in A \cup B} x_{bak} \quad \forall k \in K \quad (4)$$

$$u_a - u_b + nx_{abk} \leq n - 1 \quad (5)$$

$$\forall a \in A, b \in B, k \in K, i \neq j$$

$$1 \leq u_b \leq n - 1 \quad x_{abk} \in \{0,1\}$$

The equations (2) and (3) guarantee that each demand node is visited precisely once. The equation (4) assures that each vehicle returns to its own depot. Any Subloop is prevented by the equation (5).

In the second phase, we can adjust the model by considering a single depot (demand node is treated as a Subdepot), thus setting $m=1$. This effectively transforms the problem into a single depot Traveling Salesman Problem (TSP).

4.1 Some Definitions and Theorems

Definition 1 A Hamiltonian cycle in a graph $G = (V, E)$ is a cycle that visits each vertex in V exactly once.

This concept is crucial to the Traveling Salesman Problem (TSP), which seeks the Hamiltonian cycle with the minimum total weight in a weighted graph.

Theorem 1 In every finite weighted graph G , there exists a Hamiltonian cycle with a minimum length.

Let's consider a finite weighted graph $G = (V, E, W)$. By definition, the set of all Hamiltonian cycles in G is finite. Since every finite set of real numbers has a minimum element, there must be a Hamiltonian cycle with minimum total weight in G . Thus, a minimum length Hamiltonian cycle exists in every finite weighted graph G .

Theorem 2 In a weighted graph G , changing the vertex as the starting point will not affect the solution of the Traveling Salesman Problem (TSP).

By Theorem 1, we know that a solution to the TSP exists. Denote this solution, which is the minimum length of a Hamiltonian cycle in the graph, by t . If we change the starting point, the resulting path is a cyclic permutation of a Hamiltonian cycle in the graph. The length of any Hamiltonian cycle cannot be less than t , therefore, the length of the new Hamiltonian cycle is also at least t . However, since the weights of the edges have not changed, the length of the new cycle cannot be less than t . Thus, changing the starting point does not affect the solution to the TSP.

Based on the definition and theorems we stated, it can be concluded that the choice of a subdepot (or the starting point) in phase 2 will not affect the optimal solution of the Traveling Salesman Problem (TSP). Therefore, we can select any subdepot in phase 2 without affecting the optimality of the TSP solution.

5 The Solution Procedures

The proposed method solves an epidemic logistical problem known as MDVRP - an advanced variant of VRP. To minimize total routing costs, MDVRP involves finding routes such that every demand node is visited only once by a vehicle starting and ending at a depot. Here, it is assumed that each demand node can act as a sub-depot by receiving stock directly from its main depot and can store enough stock to fulfill the requirements of all remaining demand nodes in its group. As previously mentioned, MDVRP is an Np-hard problem, and its solution may not be possible using conventional optimization techniques. A metaheuristic is usually employed when solving such a problem. Therefore, we propose using GA, which has a global search capability and the capacity to handle complexity, parallelism, and robustness, making it particularly well-suited for this application.

5.1 Assignment of Demand node to depot

A practical and straightforward heuristic method is employed to allocate each demand node to a depot, taking distance as the basis of assignment; each demand node will be assigned to the nearest depot. For example, let's assume we have two depots: depot 1 (D1) and depot 2 (D2), and a demand node 'c'. The distance between a demand node 'c' and Depot 1 is represented as $d(c, D1)$. Similarly, the distance between the same demand node 'c' and Depot 2 is $d(c, D2)$.

Here is how demand node 'c' is assigned to depot.

- If $d(c, D1) > d(c, D2)$, in this case, Depot 2 is assigned to the Demand Node 'c' since it's closer.
- If $d(c, D2) > d(c, D1)$, In this instance, Demand Node 'c' is allocated to Depot 1, being nearer to it.
- And if $d(c, D1) = d(c, D2)$, assign 'c' to either Depot D1 or depot D2 arbitrarily.

After allocating the demand node to a depot, the next task is solving the MDVRP using GA.

5.2 Operating Procedure for GA

The Genetic Algorithm (GA) in our work operates according to the following process:

5.2.1 Chromosome Coding

GA for MDVRP requires the coding of chromosomes. Since a vehicle has to start from a depot and, after serving all demand nodes, has to return to the same depot, permutation is an effective way to represent a chromosome in MDVRP. Each in-between position in this permutation represents one demand node, and its order indicates the order in which the requirements of the demand node are fulfilled.

An example of chromosome representation with five demand nodes, viz: 1, 2, 3, 4, and 5, to be fulfilled by a depot A, may be represented as (A, 3, 1, 5, 2, 4, A). In this chromosome, the vehicle from depot A visits the Demand Node in the order $3 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 4$ and returns to depot A.

5.2.2 Initialization

The process of initialization is as follows:

1. Parameter Configuration:

- Define the number of iterations and the size of the population.
- Use a distance function to calculate the distance between each demand node and depot.

2. **Initial Population Creation:**

- Generate an initial set of chromosomes (solutions) for each depot.
- These chromosomes represent the order in which the vehicles visit the Demand Nodes.
- Form the initial population by randomly permuting the indices of the Demand Nodes.
- Each row in the population matrix corresponds to a chromosome.

The initialization process ensures the creation of an initial population of chromosomes for each selected depot as explained in section 5.1, representing potential solutions for the MDVRP. These chromosomes are randomly generated and incorporate depot and demand nodes to form complete routes.

5.2.3 **Evaluation of Fitness**

The following procedure is adopted to evaluate the fitness of a chromosome.

1. The total distance traveled by vehicles is calculated for each chromosome in the population.
2. The distance is computed by summing the distances between consecutive nodes in the chromosome, including the distance from the last demand node back to the starting depot. This accounts for the fitness score of the chromosome.
3. The lower the total distance, the better the fitness of the chromosome.

A genetic algorithm aims to minimize vehicle travel distance by finding the shortest route to serve all the demand nodes. By evaluating the fitness based on the total length, the genetic algorithm can iteratively improve the solutions by favoring shorter-distance chromosomes and selecting them for further breeding and optimization in the subsequent algorithm steps.

5.2.4 **Selection of Parents**

The tournament selection technique is utilized to select parent chromosomes for reproduction in the proposed GA. Here is an outline of this procedure: The tournament selection approach is used within genetic algorithms for parent selection. Here's how this works:

1. From the population, a random subset of chromosomes known as the tournament pool is chosen randomly and determined by algorithm parameters; these will form part of this selection process.
2. Each chromosome in the tournament pool is assessed to determine its fitness for competition. Usually, this objective function (in this instance, total vehicle travel distance) determines fitness.
3. In this step, the fitness values of each of the chromosomes in a tournament pool are compared, and one with superior fitness (i.e., shorter distance) will likely be selected as its first parent. This selection process follows the principle of survival of the fittest; more fit chromosomes have a greater likelihood of selection as first parents.
4. Steps 1-3 are repeated independently to select a second parent from the tournament pool.

Through the tournament selection method, genetic algorithms establish an environment where only suitable or "fit" chromosomes are more likely to be chosen as progenitors for subsequent generations, encouraging the proliferation of viable solutions and efficient exploration of solution spaces.

5.2.5 **The order crossover**

The Order One Crossover method is used for generating offspring from selected parents. Here's an explanation of the Order One Crossover process:

1. From the existing chromosome population, select two parent chromosomes randomly and select

a subsequence representing a subset of genes from one parent chromosome.

2. Copy this subsequence into the offspring chromosome at the same position.
3. Starting from the end of the chosen subsection, go to the other parent and take genes not already in the offspring, wrapping around to the start if necessary and inserting them into the offspring in their original order.

This process ensures that each gene from both parents appears precisely once in the offspring without any duplications. It helps preserve essential segments of the parent chromosomes while allowing for the exploration of different combinations of genes. Repetition of this procedure produces new offspring populations, forming the next generation in a genetic algorithm.

Let us take the two chromosomes: Parent 1 (A, 3, 6, 8, 9, 4, 1, 2, 5, 10, 7, A) and Parent 2 (A, 5, 9, 3, 4, 8, 10, 2, 6, 1, 7, A) as shown in table 1. In this example, we randomly select a subsequence from index 3 to 6 in Parent 1.

Parent 1											
A	3	6	8	9	4	1	2	5	10	7	A
Parent 2											
A	5	9	3	4	8	10	2	6	1	7	A
Subsequence from index 3 to 6 in Parent 1											
Offspring											
A	5	3	8	9	4	10	2	6	1	7	A

Table 1: Parents and offspring after Order One Crossover

The crossover process unfolds as follows:

1. Offspring is created by copying the subsequence of genes from Parent 1, starting from index 3 to 6 (8,9,4) excluding the first and last index for depot, into the Offspring at the same positions.
2. The remaining positions in Offspring are filled with genes from Parent 2 in the order they appear, without duplicating any existing genes. Therefore, Offspring becomes (A,5,3,8,9,4,10,2,6, 1,7, A).

The OrderOne Crossover method ensures that each gene appears precisely once in the Offspring without duplications. It combines segments of both parents, preserving essential features of the parent chromosomes while still allowing for the exploration of different combinations of genes.

5.3 Mutation

The mutation process in our genetic algorithm consists of two distinct types: Flip Mutation and Swap Mutation. Both mutations have a probability of occurrence of 0.2. A key aspect to consider here is that the Swap Mutation is applied after the Flip Mutation, which means the offspring resulting from the Flip Mutation is then subjected to Swap Mutation.

5.3.1 Flip Mutation

Flip Mutation involves an algorithm randomly selecting two positions within the chromosome - and inverting or "flipping" all genes between these positions.

Let us consider, as an example, the following chromosome: (A, 5, 9, 3,4, 8, 2, 6, 1, 10, 7, A) If we select positions 3 and 6 excluding the first and the last index we get the gene (3,4, 8, 2), the flip mutation will create the following new chromosome: (A ,5, 9, 2, 8, 4,3, 6,1, 10, 7, A).

5.3.2 Swap Mutation

In the Swap Mutation operation, the algorithm selects two random positions within the chromosome,

excluding the starting and ending points. The genes at these two positions are then swapped. Using the chromosome obtained from the Flip Mutation (A, 5, 9, 2, 8, 4,3, 6,1, 10, 7, A), if we select positions 5 and 8, final chromosome after the Swap Mutation will be (A ,5, 9, 2, 8, 1,3,6,4, 10, 7, A).

Mutation Type	Resulting Chromosome
Initial	(A, 5, 9, 3,4, 8, 2, 6, 1, 10, 7, A)
Flip	(A, 5, 9, 2, 8, 4,3, 6,1, 10, 7, A)
Swap	(A ,5, 9, 2, 8, 1,3, 6,4, 10, 7, A)

Table 2: Example of Flip and Swap Mutation

The above algorithm's code was executed on MATLAB version R2022b and ran on a 12th Gen Intel Core i7-12700H 2.30 GHz system, which is a powerful processor capable of efficiently handling the MATLAB code.

6 Analysis of result

This study utilizes a two-stage modified Genetic Algorithm (GA) to address the Multiple Depot Vehicle Routing Problem (MDVRP) within epidemic logistics. This technique was applied in three instances labelled as T01, T02, and T03 to represent different scenarios arising in India.

- **Instance T01:** This instance involves five Indian cities serving as depots and 25 cities from Madhya Pradesh State in India acting as demand nodes. These depots have been identified with letters A through E, while demand nodes are represented as numbers 1 to 25.
- **Instance T02:** This instance selects five random cities within India as depots and 50 random cities within Madhya Pradesh as demand nodes, labeling both similarly to Instance T01.
- **Instance T03:** This instance expands our scope internationally. Five randomly chosen neighboring countries of India serve as depots, with 50 randomly chosen cities scattered across India serving as demand nodes. Labeling of depots and nodes remains consistent with previous instances.

The two-phase modified GA is executed for each of these instances over an adequate number of iterations, aiming to find an efficient and effective solution to the MDVRP within this specific epidemiological framework.

The proposed Genetic Algorithm presents unique considerations due to its grouping mechanism in the initialization stage. Here, demand nodes are grouped based on proximity, each assigned to the nearest depot. Consequently, we consider two specific cases for analysis:

1. **Case 1: Unrestricted choice of demand node.** In this scenario, there are no limits or constraints on assigning demand nodes to depots. This approach maximizes flexibility and potentially minimizes total distance traveled due to the nearest-neighbor assignment strategy. A downside is that certain depots may not be assigned any demand nodes, leading to possible inefficiencies in resource utilization.
2. **Case 2: Each depot serves a minimum of two demand nodes.** For this case, the analysis incorporates an additional rule in phase 1 of the algorithm. Each depot is required to serve at least a predefined number of Demand Nodes. In our study, we have fixed it to 3 Demand Nodes. This rule ensures a more even distribution of demand nodes among depots, improving the overall balance in depot utilization. This is in addition to the original scheme of assignment that still prioritizes minimum distance. It's important to note that this rule might slightly increase the total distance traveled, as some demand nodes might not be assigned to their nearest depot.

The following tables 3,4 and 5 display the results of implementing the two-phase Genetic Algorithm (GA) on three instances (T01, T02, and T03) as previously defined. Each table compares the route

generated in Phase 1 and Phase 2 of the modified GA, the distances (in km.) covered by the standard GA and the two-phase GA, and the travel time (in hours) for both approaches. Here we have considered that the vehicle moves at a constant speed of 50 km/hour. Two specific cases discussed earlier are considered in each instance: Case 1, where demand nodes are freely assigned to depots, and Case 2, where each depot must serve a minimum of three demand nodes.

These tables encapsulate the performance of the GA, shedding light on the efficiency and effectiveness of the proposed two-phase adaptation in solving the MDVRP within an epidemic logistics framework. The routes, distances, and travel times are displayed for each instance, aiding in the comprehensive analysis of the GA’s performance and adaptability to different scenarios.

Table 3: Instance T01:25 random cities of Madhya Pradesh State of India is taken as demand nodes and 5 random cities of India as depots.

Instance	Route (Phase 1)	Route (Phase 2)	Distance (in km) (only GA)	Distance (in km) (2-Phase GA)	Time (in Hours) (only GA)	Time (in Hours) (2-Phase GA)
Case 1	A-8-10-5-11-13-12-A	10-8-12-13-11-5-10	4471	3393	89	67
	B-16-17-4-1-3 -9 -7-6-B	4-1-3-9-7-6-16-17-4				
	D-14-15-2-25-21-24-23-29-D	21-24-23-14-15-2-25-21				
	E-20-22-19-18-E	22-20-18-19-22				
Case 2	A-26-8-10-5-11-13-12-26-A	10-8-12-13-11-5-10	5598	3577	111	71
	B-27-6-17-1-3-9-7-B	1-3-9-7-6-17-1				
	C- 20-19-18-C	19-20-18-19				
	D-14-15-2-25-21-24-23-D	25-21-24-23-14-15-2-25				
	E-16-4-22-E	16-4-22-16				

The detailed analysis of the tables showcasing the results for the three instances T01, T02, and T03, conducted using the two-phase Genetic Algorithm (GA) approach, are as follows:

Instance T01: In case 1, the standard GA approach achieved a total distance of 4471 km, while the two-phase GA achieved a total distance of 3393 km, a reduction of 24.1%. Similarly, the travel time was reduced by 24.7%, from 89 hours to 67 hours. For case 2, where each depot serves a minimum of three Demand Nodes, the total distance achieved by the standard GA was 5598 km, while the two-phase GA completed a total distance of 3577 km, a reduction of 36.1%. The travel time was reduced by 36.0%, from 111 hours to 71 hours. These results highlight the increased efficiency of the two-phase GA over the standard GA.

Table 4: T02:50 random cities of Madhya Pradesh State of India is taken as demand nodes and 5 random cities of India is taken as depots.

Instance	Route (Phase 1)	Route (Phase 2)	Distance (in km) (only GA)	Distance (in km) (2-Phase GA)	Time (in Hours) (only GA)	Time (in Hours) (2-Phase GA)
Case 1	A-15-32-6-39-29-8-10-12-A	32-6-29-39-12-10-8-15-32	4932	3786	98	75

	B-4-18-17-49-24-19-44-40-30-2-31-42-B	17-18-4-19-44-40-30-2-31-42-49-24-17				
	D-25-50-3-11-20-45-16-D	50-25-3-11-20-45-16-50				
	E-41-46-21-33-5-43-7-1-14-13-28-38-22-36-23-48-27-35-34-37-47-26-9-E	7-5-33-47-26-9-41-46-21-43-14-13-28-38-36-22-23-48-34-27-37				
Case 2	A-12-10-8-39-29-6-32-15-A	10-12-39-29-6-32-15-8-10	5848	3943	116	78
	B-4-18-19-44-30-40-2-31-42-49-24-17-B	30-2-31-42-49-24-17-18-4-19-44-40-30				
	C-13-45-16-C	16-13-45-16				
	D-50-25-3-20-11-D	3-50-25-11-20-3				
	E-33-43-7-14-28-38-22-36-48-23-34-37-27-35-1-5-47-26-9-46-21-41-E	41-43-7-14-28-38-22-36-48-23-34-37-27-35-1-5-33-47-26-9-46-21-41				

Instance T02: In case 1, the total distance covered using the standard GA was 4932 km, while the two-phase GA managed to reduce the distance to 3786 km, indicating a reduction of 23.2%. The travel time also decreased from 98 hours to 75 hours, a drop of 23.5%. In case 2, where each depot serves a minimum of three Demand Nodes, the total distance achieved by the standard GA was 5848 km. In contrast, the two-phase GA reached a total distance of 3943 km, showing a reduction of 32.6%. The travel time was reduced from 116 hours to 78 hours, a decrease of 32.8%.

Instance T03: In the first case, the total distance covered by the standard GA was 12700 km, while the two-phase GA managed to reduce this distance to 11311 km, a reduction of 10.9%. The travel time also decreased from 254 hours to 226 hours, marking a decrease of 11.0%. In the second case, where each depot serves a minimum of three Demand Nodes, the total distance covered by the standard GA was 19165 km, while the two-phase GA reduced the entire distance to 12095 km, signifying a reduction of 36.9%. The travel time it also decreased significantly, from 383 hours to 241 hours, indicating a decline of 37.1%

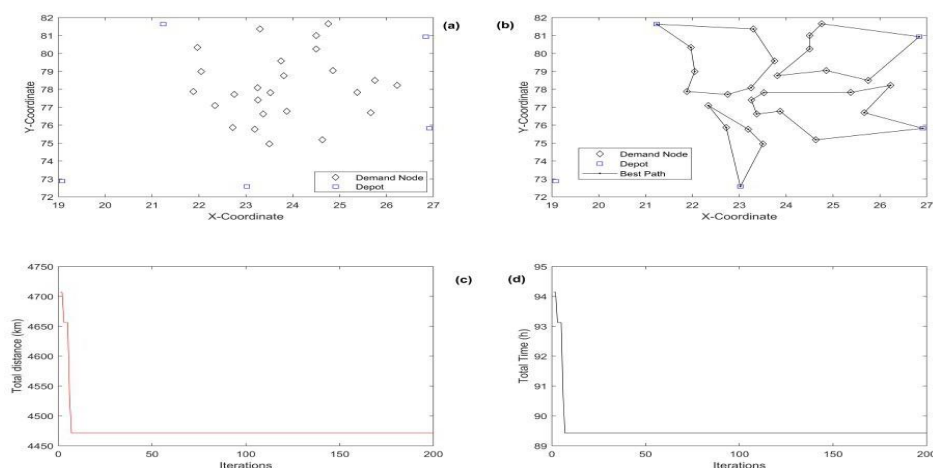
Table 5: **T03: 50 random cities of India is taken as demand nodes and 5 neighboring countries of India as depots**

Instance	Route (Phase 1)	Route (Phase 2)	Distance (in km) (only GA)	Distance (in km) (2-Phase GA)	Time (in Hours) (only GA)	Time (in Hours) (2-Phase GA)
	A- 47-48-49 -A	47-48-49-47				

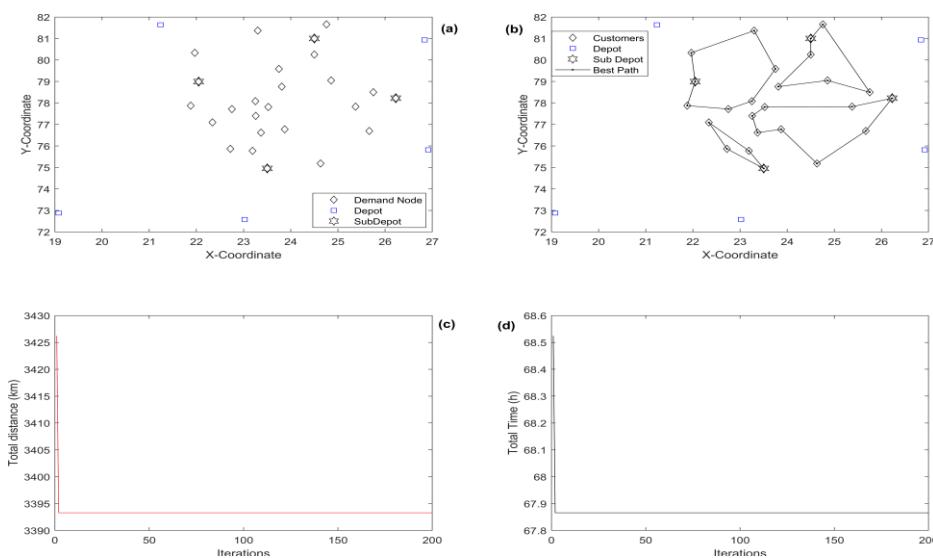
Case 1	B-16-30-7-33-25-32-35-13-29-21-8-40-24-5-17-14-15-43-38-31-28-19-23-18-2-22-10-20-36-12-11-50-B	35-32-25-33-7-30-16-50-11-12-36-20-22-2-18-23-19-28-31-10-38-43-15-14-17-5-40-24-8-21-29-13-35	12700	11311	254	226
	C-39-34-42-3-41-44-1-45-27-9-4-46-37-26-6-C	44-41-42-34-39-3-6-26-37-46-4-9-27-45-1-44				
Case 2	A-47-48-49-A	48-47-49-48	19165	12095	383	241
	B-25-32-50-11-12-20-23-19-2-18-22-10-36-15-14-43-17-5-40-24-8-21-29-13-35-33-7-30-16-B	50-32-25-16-30-7-33-35-13-29-21-8-40-24-5-17-14-15-43-10-19-23-18-2-22-20-36-12-11-50				
	C-3-41-44-1-45-27-9-4-46-37-26-6-C	27-45-1-44-41-3-6-26-37-46-4-9-27				
	D-39-34-42-D	42-39-34-42				
	E-28-31-38-E	19 -18 -20-22-19				

In all instances and cases, the two-phase GA demonstrated superior performance compared to the standard GA. The total distance traveled, and travel time reduction suggests that the two-phase GA can effectively and efficiently solve the MDVRP within an epidemic logistics context. Analyzing the two cases in detail, the most noticeable difference is in the number of demand nodes each depot must serve. In Case 1, depots can do any number of demand nodes, including none at all, while in Case 2, each depot is required to serve a minimum of three demand nodes. The comparison shows that Case 2 generally yields a slightly higher total distance traveled than Case 1. This is because Case 2, by enforcing each depot to serve at least three demand nodes, may take longer than Case 1 due to the added complexity of the constraint that each depot serves a minimum number of demand nodes. Despite this, the travel time is still within an acceptable range. Despite the slightly higher total distance traveled, the results from Case 2 may be better than those from Case 1. The reason for this lies in the more balanced utilization of depots. In Case 1, some depots may not be assigned any demand nodes, leading to potential inefficiencies in resource utilization. Case 2, conversely, ensures that every depot serves a minimum number of demand nodes, thereby improving the overall balance in depot utilization. This is a crucial factor in practical scenarios where optimal utilization of resources is vital for effective logistics management and ensuring that no depot is left unused. Therefore, despite the slight increase in distance traveled, the results from Case 2 are more optimal than Case 1 when considering real-world logistics scenarios.

In the context of the graph obtained as output by running the algorithm on MATLAB software, we now analyze the results for Instance T01 under Case 1 after both Phase 1 and Phase 2 of the two-phase Genetic Algorithm (GA) shown in figure 2. In Phase 1, the first part of the sub-figure: 2a indicates the location of the depots and the demand nodes in coordinates (based on Latitude and longitude).



a Routing assignment Phase-1



b Routing Assignment Phase-2

Figure 2: Routing assignment for T01

This provides a spatial understanding of the problem, helping visualize the distance and relationship between the various nodes. The second sub table (b) part provides the solution path after Phase 1. The solution path is the route the vehicles should take to service all the demand nodes, starting and ending at the same depot. Due to the available assignment of demand nodes to depots in Case 1, some depots might have yet to be allocated any demand nodes, thus remaining unused. Part (c) of the sub table provides the graph of the total distance (kilometers) against the number of iterations executed by the GA. The curve depicted is a decreasing converging curve, indicating that the total distance converges to a total distance of around 4471 kilometers the number of iterations increases. The convergence of the graph shows the robustness of the algorithm. The fourth sub table (d) is a graph representing the total time the vehicles take to complete their routes, considering a constant speed of 50 km/hr. Like the entire distance, the time against the number of iterations is also a decreasing curve. This curve converges to approximately 67 to 68 hours, the minimum time needed to complete all the routes.

The second sub-figure: 2b corresponds to Phase 2 of the two-phase GA for the same instance (T01) under Case 1. In Phase 2, some changes are introduced to optimize the solution further. Firstly, the concept of subdepots is introduced. These points within the existing routes can serve as temporary depot points to further optimize vehicle routing. Similar to Phase 1, the first part of this sub table (a) shows the location of the depots, subdepots, and demand nodes in coordinates. The second part (b) shows the

improved solution path after Phase 2, which now includes subdepots. Part(c) and (d) of the sub table depict the total distance and time, respectively, after the implementation of Phase 2. Here the converging graph illustrates the robustness of the algorithm.

In conclusion, the two-phase modified Genetic Algorithm significantly improves the efficiency of resource allocation in the context of epidemic logistics, as demonstrated by the notable reductions in distance and time across all cases. The algorithm balances efficiency and resource utilization by ensuring a more even distribution of demand nodes among depots. This balance is crucial in an epidemic context, where the timely and effective delivery of services can make a substantial difference in public health outcomes.

The following table 6 gives a comparison of Computation time for the instance T01.

Table 6: GPU Computing

	50 Iteration		100 Iteration		200 Iteration		500 Iteration		800 Iteration		1000 Iteration	
	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
Without GPU Computation Time (in sec)	21.1285	23.0359	187.4969	256.3670	211.6700	233.8372	567.0135	941.4195	1845.7855	2520.3263	2956.9455	3152.3435
With GPU Computation Time (in sec)	20.9413	21.4510	91.1177	106.9409	105.7417	134.4016	268.7735	340.4961	865.5438	952.6843	1060.8655	1127.5666

The GPU computing was carried out using a CUDA Device with the following properties: The GPU used is an 'NVIDIA GeForce RTX 3060 Laptop GPU', which has a compute capability of '8.6', indicating it supports a high level of precision. This GPU supports double-precision computations, denoted by 'Supports Double' being 1. The GPU driver and toolkit versions are 11.6000 and 11.2000, respectively. The total memory available on the GPU is 6.44 GB, with 5.23 GB available during data collection.

With these specifications, the NVIDIA GeForce RTX 3060 Laptop GPU was leveraged to accelerate the computation for the code's depot allocation, mutation, and crossover parts, as described in the table. The GPU significantly reduced computation times across various iterations and computational tasks. The table compares computation times in two scenarios: with and without GPU acceleration across different iterations (50, 100, 200, 500, 800, 1000) for two different cases, Case 1 and Case 2 of instance T01. The GPU computation has been applied to specific parts of the code, such as depot allocation, mutation, and crossover. We can make several observations from the table 6. For both Case 1 and Case 2, computation times with GPU are consistently lower than those without GPU, highlighting GPU acceleration's computational benefits. Comparing Case 1 and Case 2, it is evident that computation times are generally higher in Case 2 for both scenarios, suggesting that Case 2 might be more computationally demanding than Case 1. As the number of iterations increases, computation times also rise for both with and without GPU scenarios and for both cases. However, the rise is steeper in the without GPU scenario. For instance, in Case 1 without GPU, computation time grows from 21.1285 seconds at 50 iterations to 2956.9455 seconds at 1000 iterations. In contrast, with GPU, time increases from 20.9413 seconds at 50 to 1060.8655 seconds at 1000 iterations. The relative performance gain of GPU computing increases with the number of iterations, indicating that the benefit of GPU acceleration becomes more significant for more intensive computational tasks.

In conclusion, the data suggests that utilizing GPU computing, specifically in the code's depot allocation, mutation, and crossover parts, significantly reduces computation times across a range of iterations and computational tasks.

7 Conclusion

In this study, a method was proposed to solve a logistical problem relevant to epidemic scenarios, focusing specifically on the Multiple Depot Vehicle Routing Problem (MDVRP). The MDVRP, known for its complexity and NP-hard nature, extends upon the classical Vehicle Routing Problem (VRP) and necessitates efficient strategies for minimizing routing costs and ensuring each demand node is serviced only once.

By introducing the concept of sub-depots – demand nodes that transfer stock from the main depot and adequately serve the requirements of other demand nodes in their group we were able to formulate a unique and dynamic approach to the MDVRP.

Given the NP-hard nature of the MDVRP, conventional optimization methods are deemed impractical. Thus, we leveraged the Genetic Algorithm (GA) due to its inherent global search capability, robustness, ability to manage complexity, and parallelism.

The first stage of the proposed method involved assigning each Demand Node to the closest depot. This simple yet efficient heuristic was based on the relative distances between demand nodes and depots. Our method demonstrated its adaptability by assigning a demand node to an arbitrary depot in case of equal distances.

Subsequently, we employed GA for solving the MDVRP. A chromosome was represented as a permutation of demand nodes, bookended by the depot, where each in-between position represented a unique demand node. The order of these demand nodes in the permutation indicated the sequence of demand fulfillment.

Overall, the proposed method provides a flexible, efficient, and adaptive approach to solving the MDVRP in the context of epidemic logistics. By strategically combining a distance-based heuristic for depot assignment and Genetic Algorithm for demand node route optimization, the method presents a promising solution to the MDVRP. Further studies may enhance this approach by incorporating more real-world constraints and using more advanced optimization techniques to improve the solution quality and efficiency.

References

- [1] J. K. Lenstra and A. R. Kan, (1981) “Complexity of vehicle routing and scheduling problems,” *Networks*, vol. 11, no. 2, pp. 221–227. <https://doi.org/10.1002/net.3230110211>
- [2] J.-F. Cordeau and G. Laporte, (2000). “A tabu search heuristic for the periodic and multi-depot vehicle routing problems,” *Networks*, vol. 30, no. 2, pp. 105–119. [https://doi.org/10.1002/\(SICI\)1097-0037\(199709\)30:2%3C105::AID-NET5%3E3.0.CO;2-G](https://doi.org/10.1002/(SICI)1097-0037(199709)30:2%3C105::AID-NET5%3E3.0.CO;2-G)
- [3] G. Laporte (1998), “The vehicle routing problem: An overview of exact and approximate algorithms,” *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358. [https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/10.1016/0377-2217(92)90192-C)
- [4] J.-F. Cordeau, G. Laporte, and A. Mercier (2001), “A unified tabu search heuristic for vehicle routing problems with time windows,” *Journal of the Operational research society*, vol. 52, no. 8, pp. 928–936. <https://doi.org/10.1057/palgrave.jors.2601163>
- [5] E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys (1985), “The traveling salesman problem: A guided tour of combinatorial optimization,” *Estimation, Distribution and Local Search*, vol. 3.
- [6] D. Applegate, R. Bixby, V. Chvátal, and W. Cook (2007), *The traveling salesman problem: A computational study*. Princeton University Press. <https://doi.org/10.1515/9781400841103>
- [7] J. Cordeau, P. Toth, and D. Vigo, (1997) “A survey of optimization models for train routing and scheduling,” *Transportation Science*, vol. 32, no. 4, pp. 380–404. <https://doi.org/10.1287/trsc.32.4.380>
- [8] Li, C. Xie, P. Peng, X. Gao, and Q. Wan 2021, “Multi-objective location-scale optimization model and solution methods for large-scale emergency rescue resources,” *Environmental Science and Pollution Research*, pp. 1–14,

2021 <https://doi.org/10.1007/s11356-021-12753-9>

- [9] M. Queiroz, D. Ivanov, A. Dolgui, and S. Wamba(2020), “Impacts of epidemic outbreaks on supply chains: Mapping a research agenda amid the covid-19 pandemic through a structured literature review,” *Annals of Operations Research*, vol. 1. <https://doi.org/10.1007/s10479-020-03685-7>
- [10] M. U. Farooq, A. Hussain, T. Masood, and M. S. Habib(2021), “Supply chain operations management in pandemics: A state-of-the-art review inspired by covid-19,” *Sustainability*, vol. 13, no. 5. <https://doi.org/10.3390/su13052504>
- [11] J. Pasha, A. L. Nwodu, A. M. Fathollahi-Fard, G. Tian, Z. Li, H. Wang, and M. A. Dulebenets, (2022) “Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings,” *Advanced Engineering Informatics*, vol. 52, p. 101623. <https://doi.org/10.1016/j.aei.2022.101623>
- [12] F. Glover, (1994), “Genetic algorithms and scatter search: unsuspected potentials,” *Statistics and Computing*, vol. 8, no. 3, pp. 131–140. <https://doi.org/10.1007/BF00175357>
- [13] C. Prins, “A simple and effective evolutionary algorithm for the vehicle routing problem,” *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002. [https://doi.org/10.1016/S0305-0548\(03\)00158-8](https://doi.org/10.1016/S0305-0548(03)00158-8)
- [14] A. Chehouri, R. Younes, J. Khoder, J. Perron, and A. Ilinca(2004), “A selection process for genetic algorithm using clustering analysis” *Algorithms*, vol. 10, no. 4,123 , (2017). <https://doi.org/10.3390/a10040123>
- [15] R. Pellerin, N. Perrier, and F. Berthaut(2020), “A survey of hybrid metaheuristics for the resource- constrained project scheduling problem,” *European Journal of Operational Research*, vol. 280, no. 2, pp. 395–416. <https://doi.org/10.1016/j.ejor.2019.01.063>
- [16] C. Cao, Y. Xie, Y. Liu, J. Liu, and F. Zhang(2023), “Two-phase covid-19 medical waste trans- port optimisation considering sustainability and infection probability,” *Journal of Cleaner Production*, vol. 389, p. 135985. <https://doi.org/10.1016/j.jclepro.2023.135985>
- [17] H.-F. Ling, Z.-L. Su, X.-L. Jiang, and Y.-J. Zheng(2021), “Multi-objective optimization of inte- grated civilian-military scheduling of medical supplies for epidemic prevention and control,” *Healthcare*, vol. 9, no. 2,126 <https://doi.org/10.3390/healthcare9020126>.
- [18] E. Breitbarth, W. Groß, and A. Zienau, (2021) “Protecting vulnerable people during pandemics through home delivery of essential supplies: a distribution logistics model,” *Journal of Humanitarian Logistics and Supply Chain Management*. 10.1108/JHLSCM-07-2020-0062
- [19] Y. Wang, J. Zhe, X. Wang, Y. Sun, and H. Wang(2022), “Collaborative multidepot vehicle routing problem with dynamic customer demands and time windows,” *Sustainability*, vol. 14, no. 11. <https://doi.org/10.3390/su14116709>