Enhanced Genetic Algorithm Optimization Models for Vehicular Routing Problems

K. Premkumar, R. Baskaran, M. Shanmugam

Abstract: ***The Vehicle Routing Problem (VRP) is one of the most studied combinatorial optimization problems because of its practical relevance and complexity. Though there are several techniques have been proposed to solve the VRPs and its variants effectively, each technique has its own tradeoff values in terms of the performance factors. From this perspective, the work presented in this paper proposed an intelligent routing strategy for VRP based on distance values between the cities. The proposed strategy uses an enhanced model of Genetic Algorithm to find the optimal tour paths among the cities under distance based optimized tour path estimation scenarios. For distance-based optimization approach, experiments were performed on the standard benchmark TSP instances obtained from TSPLIB. A set of fine-grained result analyses demonstrated that the proposed model of routing strategies performed comparatively better w.r.t. the existing relevant approaches. By considering this problem as the base, a distinct model was developed as a set of assistive modules for Genetic Algorithms (GA), which are aimed at improving the overall efficiency of the typical GA, particularly for optimization problems. The capability of the proposed optimization models for VRP is demonstrated at various levels, particularly at the population initialization stage, using a set of well-defined experiments.***

Keywords : Genetic Algorithm, Vehicle Routing, TSP

I. INTRODUCTION

The Vehicular Routing Problem (VRP) is to deliver the goods from the central depot to the number of customers who had placed orders. The main objective is to find the routes and minimize the cost of distributing the goods [1]. Vehicle Routing Problem can be solved by two approaches, namely exact approach and heuristic approach. A Genetic Algorithm is a search heuristic, which belongs to the category of Evolutionary Algorithm (EA). To find the feasible route in VRP, we are moving to the optimization algorithms. Many parameters are there to find the optimal solution in VRP, but the proposals are concentrating on the time traveled by the vehicle and eliminating the worst routes. It is possible to get an optimal solution or feasible solution from those routes. In [2], they focused on those things and eliminated the routes and proposed a dynamic programming method to that problem-based On-Demand Transportation (ODT). In [3] a Variable Neighborhood Search (VNS) is proposed for solving the Multi-Depot Vehicle Routing Problem with Loading Cost (MDVRPLC). The variance of vehicle routing problem is combined in this method; that is the Multi-Depot Vehicle Routing Problem (MDVRP) and Vehicle Routing Problem with Loading Cost (VRPLC). The VNS consist of three phases, one is for generating the initial solution, the second one is for selecting the search neighborhood solution and the third one is used for neighborhood solution acceptance. The search neighborhood solution performs 4 operations they are select, insertion, crossover, mutation operations, the main flaw in this operations are; cities are selected randomly in all the operation. Without considering the cost or any other parameters in the operation. In VRP using optimization algorithm initially, we will generate all the possible routes, from that we will find the feasible route. In real time that is not always possible, the vehicle may get the request form a new customer, Dynamic Vehicular Routing Problem (DVRP)[4]. We have to find a dynamic approach to solve this problem; in [5] they solved this problem using Particle Swarm Optimization (PSO) and Variable Neighborhood Search (VNS) paradigms. In that, they analyzed that the PSO performance is well then the VNS. [6] The ALNS heuristic framework was further preceded by the influence of [7] [8] [9] to find an optimal route and least distance in different types of VRP. Instead of using a single large neighborhood as in LNS, this executes many removals and insertion operators in the solution. Depend on the previous performance these operators are chosen during the execution and set a score to the operators, which will boost the solution. In our survey, we discussed how the VRP is solved in using different methods, with different parameters and different constraints. We proposed a socially inspired transportation problem, in this based on the distance in the path, we are routing the vehicle. The experiments are done using the slandered TSP benchmark instances and then analyzed the performance with different initialization techniques. Our paper is organized as follows: Section 2 will discuss the background information; ODV Based Initialization Technique and TSP to understand the basic concepts. In Section 3, we will discuss the proposed system and algorithm, description for a different scheme. In section 4 we will discuss the Experimentation Methodologies and Section 5 describes the experimental analysis of different scheme with different initialization techniques and their results. Finally, Section 6 concluded of the paper.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author

K. Premkumar*, Computer Science and Engineering, Manonmaniam Sundaranar University, Tamil Nadu, INDIA. Email: premkvpt@gmail.com

R. Baskaran, Department of Computer Science and Engineering, Anna University, Chennai, INDIA. Email: baaski@cs.annauniv.edu

M. Shanmugam, Computer Science & Engineering, Vignan's Foundation for Science, Technology, and Research, Guntur, INDIA.

II. BACKGROUND WORK

A. ODV Based Initialization Technique for GA

To set the opening population we are using an operative population initialization technique since the starting city is fixed in our proposal, we are moving to The EV (Equi-begin with Variable diversity) based ODV (Ordered Distance Vector) population seeding technique based on the ODV matrix [11].

- ***Ordered Distance Vector (ODV)***

In ODV the cities are sorted based on the distance that is computed by the permutation of $City_1, City_2, City_3 . . City_n$ cities. The ODV of a $City_x$ is,

$ODV(City_x) = City_y, City_{y+1}, City_{y+2} . . City_{n-1}$ (1)

Then,

$D(City_{x,y}) \leq D(City_{x,y+1}) \leq D(City_{x,y+2}) \ldots \leq D(City_{x,n-1})$ (2)

$Where\ D(City_{x,y}) is\ the\ distance\ between\ the\ City_x and\ City_y$

For each city, the ODV generates corresponding least distance cities in sorted order and rank the cities based on the distance, then it will move to the ODM (Ordered Division Matrix) that is given by$n(n-1)$matrix,

$$ODM = \begin{vmatrix} \mathrm{ODV}_{City_1} \\ \mathrm{ODV}_{City_2} \\ \mathrm{ODV}_{City_3} \\ \cdots \\ \mathrm{ODV}_{City_n} \end{vmatrix} = \begin{vmatrix} City_{1(y)} & City_{1(y+1)} & City_{1(y+2)} & \cdots & City_{1(n-1)} \\ City_{2(y)} & City_{2(y+1)} & City_{2(y+2)} & \cdots & City_{2(n-1)} \\ City_{3(y)} & City_{3(y+1)} & City_{3(y+2)} & \cdots & City_{3(n-1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ City_{n(y)} & City_{n(y+1)} & City_{n(y+2)} & \cdots & City_{n(n-1)} \end{vmatrix}$$

- **Best adjacent value (ba):**

This method is efficient because of the ba value; it is used to select the next city in population generation. The other techniques a random city is added as nethe xt city. Based on the size of the population, size of the ba value has been changed. The size for best adjacent (ba) value is assigned as follows:

$If\ 1 \leq n \leq 10, then\ 2 \leq ba \leq 3.$

$If\ 11 \leq n \leq 100, then\ 2 \leq ba \leq 4.$

$If\ 101 \leq n \leq 1000, then\ 2 \leq ba \leq 5.$

$If\ n \geq 1001, then\ 2 \leq ba \leq 6.$

- **Equi-begin (Eb):**

The starting city of each individual is always same (i.e.) $City_1$ is fixed for all the individuals in the population. In our proposal, the starting city of the individuals is fixed so we are applying this method.

- Variable diversity (Vd) :

The Next city in the individual is added based on the bax value, bax is an integer that selected within the range bavalue. The city in the position of bax value is moved to the next city location of the individual.

- $ODV - EV$ Method:

As we discussed above, Using EV method we generating the population, in that the starting city of each individual is fixed and based on thebax value the next city of the individual is chosen and added. The individuals in the population have high permutation of cities and the time complexity can be reduced. Number of maximum individuals in the population during initialization is,

$\max(tot(Pop_{ODM})) = ba^{n-1}$ (4)

Where $tot(Pop)_{ODM}$ the total number of individuals in the population is, ba is the best adjacent value and n is the number of cities.

The initial population is generated EV method is given by,

$$Pop_{ODM} = \begin{vmatrix} \delta_1(1) & \delta_1(City_2) & \delta_1(City_3) & \cdots & \delta_1(City_n) \\ \delta_2(1) & \delta_2(City_2) & \delta_2(City_3) & \cdots & \delta_2(City_n) \\ \delta_3(1) & \delta_3(City_2) & 3(City_3) & \cdots & \delta_3(City_n) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \delta_n(1) & \delta_n(City_2) & \delta_n(City_3) & \cdots & \delta_n(City_n) \end{vmatrix} \quad (5)$$

Since the initial city, all the individuals in the population are fixed, $\delta_1(1) \equiv \delta_2(1) \equiv \delta_3(1) \ldots, \delta_n(1)$

B. Traveling salesman Problem

The TSP is an NP-hard problem in combinatorial optimization, the objective is to find the least possible route to visit each city exactly once and returns to the starting city. It organized into symmetric traveling salesman problem, asymmetric traveling salesman problem, and multi traveling salesman problem. In our paper we are using symmetric traveling salesman problem, Let$G = \{City_n, A_n\}$ be a complete undirected graph such that the cities $Cit(3)\in \{City_1, City_2, City_3, \ldots City_n\}$ and the arcs $A \in \{(City_1, City_2), (City_1, City_3), \ldots (City_1, City_n) \ldots$ $(City_2, City_3), (City_2, City_4), \ldots (City_2, City_n) \ldots$ $(City_{n-1}, City_n,),\}$. In Graph$G, City_n$ represent the cities and A_n represents the arcs (i.e.) the path between the cities. The distance between the $City_i$ and the$City_j$ is represented as $DM(City_i, City_j)$, here the$City_i \neq City_j$. The TSP is formulated as follows,

$$Optimal\ Distance = \sum_{k=1}^{n} DM(City_k, City_{k+1}), \quad i+1 \equiv 1 \quad (6)$$

Since it is symmetric TSP, it gives $n(n-1)!/2$ possible solutions for any permutation of $n$cities. In our proposed system, the Starting city is samthe e for the entire problem, so it gives the possible solutions $n(n-2)!/2$

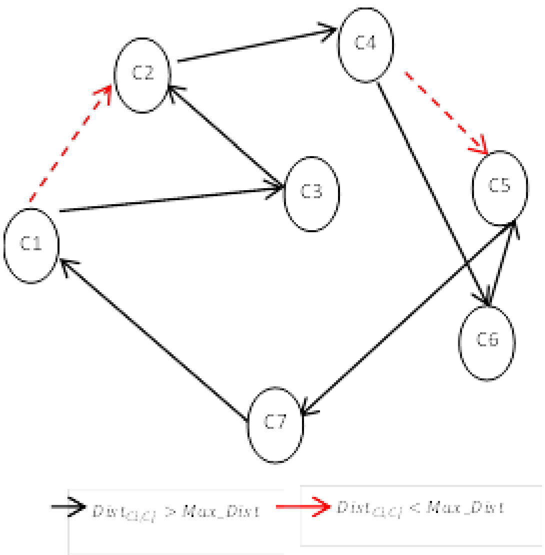
III. PROPOSED SYSTEM

A. Problem Formulation

The main objective of vehicle routing problem is to find the route with the least distance, as we discussed in the section [1] many proposals are there to solve this problem. We are contemplating this problem in different perspective and used various initialization methods to this transportation problem effectively. A vehicle wants to reach a destination; it is possible there may be any number of paths in between the starting point and the destination. Before going to enter into a path, the vehicle has to check the possible least distance. If the distance is more, it has to take the alternative path otherwise it can proceed through the path. We applied this in the symmetric Travelling salesman Problem, the objective is to find a least possible route to visit each city exactly once and returns to the starting city with respect to the distance. (i.e.) our main objective is to minimize the distance in the path. We will discuss our proposed system with a diagrammatic example.

Retrieval Number: B4410129219/2019©BEIESP
DOI: 10.35940/ijeat.B4410.129219
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication



Explanation: In this example, we have 7 cities$(C1, C2, C3 \ldots . C7)$, assume that the cities are arranged based on the distance$(d(C1) \leq d(C2) \leq C3 \ldots \leq C7)$. The objective of TSP is to visit all the cities once and then return to the stating city. Our objective is to visit the all the city with the least distance. The starting city is $c1$ and its target is to visit the next city $c2$ which is having the minimum distance. The distance is less than the maximum distance that is next adjacent path we can visit that city otherwise, we have to move to the next city with least distance. This will continue until we reach the starting city. Note that the distance between the path $c7$ and $c1$, the vehicle have to wait and then it can take that path.

To improve the efficiency of the optimal tour we are going for the genetic algorithm. In genetic algorithm many methods are used to generate the initial population, in this proposal we are using the ODV based population seeding technique, the initialization of the set population is done through Equi-begin with Variable diversity (EV) method discussed in section 2. In our proposal, we are dealing this problem with Optimal Distance based routing scenario. Now we will see the algorithm and its explanation.

- **Algorithm Explanation**
  - ***Population Initialization****:* Set the parameters, number of cities as n ($City_1, City_2, City_3, \ldots City_n$), maximum number of generation G , population size $PopSize$ and maximum distance in the path between the two cities $dist_{x,y}$. The cities are arranged in the increasing order using the Distance Matrix (DM) and then rank the cities move to the Ordered Division Matrix (ODM) matrix. Bubble sort is used for arranging the cities. Depending on the ODM, the initial population is generated using equi-begin with the variable diversity (EV) Method. In that first get the initial city and then make it as current city in the individual. Then generate the next city randomly from the ODM with in the limit of *'ba'* value (i.e.) $'R'\ where\ 1 \leq R \leq ba$ (Depending on the population size the 'ba' value will change). Check whether the next city is already present in the Individual and also the distance between the current city and the next city should be minimum (i.e. the distance between the cities should be less than the next adjacent path) $Next\_City \notin Indiv\ and\ DIST_{Cur\_City,Next\_City} < Max\_Dist$ and then add the city in the Individual. Do these procedures until we get a complete individual and then add the individual to the population $POP_{Size} \leftarrow Indiv$ do this procedure till we reach the maximum number of population, this process is population initialization. After initialized the Population, evaluate the total cost $\forall[1, \leq i \leq PopSize],\ TC_i \leftarrow \sum_{j=1}^{n} DM(Indiv_i(j), Indiv_i(j+1)),\ \ j+1 \equiv 1$ of all the individuals from the whole population. Now we are going for Genetic Algorithm, to improve the effectiveness of the population. In this we are analyzing cost scenario.

B. Optimal Distance Based Routing in TSP

In Optimal Distance Based Routing, we are moving towards a single objectthe ive, distance. From the generated initial population we achieved the optimal distance based routing based on the distance. Depending on the minimum distance the selection and the crossover operation has been performed. In the selection process, the elitist individuals have been selected from the population.

Here $CPOP_{PopSize \times n}$ is the current population, which consists of possible solutions.

$$TC_j = \begin{vmatrix} TC_1 \\ TC_2 \\ TC_3 \\ \cdots \\ TC_{PopSize} \end{vmatrix} = \begin{vmatrix} DM(Indiv_1) \\ DM(Indiv_2) \\ DM(Indiv_3) \\ \cdots \\ DM(Indiv_{PopSize}) \end{vmatrix} = \begin{vmatrix} Indiv_1 \\ Indiv_2 \\ Indiv_3 \\ \cdots \\ Indiv_{PopSize} \end{vmatrix} =$$

$$\begin{vmatrix} City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \\ City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \\ City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \end{vmatrix} =$$

$$CPOP_{PopSize\ Xn} \qquad (7)$$

$PopSize$is the population size and n is the number of cities. Each tour or possible solution in the population is represented as individual $Indiv_j$ and TC_j is the total cost of each individual in the population. The individuals having best fitness or minimum tour costs are selected and send to the next generation.

$$POP_{ERXn} =$$

$$\begin{vmatrix} City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \\ City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \\ City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ City_1 & City_i & City_i & \cdots & City_{n-1} & City_1 \end{vmatrix} = \begin{vmatrix} Indiv_1 \\ Indiv_2 \\ Indiv_3 \\ \cdots \\ Indiv_{ER} \end{vmatrix} = \begin{vmatrix} TC_1 \\ TC_2 \\ TC_3 \\ \cdots \\ TC_{ER} \end{vmatrix} = min(TC_{ER}) \quad (8)$$

- ***Algorithm***

Variables Used

n ←No. of cities
G←Maximum No. of Generations
$PopSize$ $Dist_{ci,cj} < Max\_Dist$
lutions
Max_Dist←Maximum Distance
$Pop_{Popsize\ xn}$←Possible solutions / Search space
$Indiv_{mxn}$←Single solution in the possible solutions / Search space
$City_n$←Subset of each individual
$DM_{n\times n}$← Distance Matrix
$DM(x,y)$←Cost between x^{th} City and y^{th}city
ba← Best Adjacent value
bax← Random value based on ba value $bax = RAND(1,ba)$
$ODM_{n\times ba}$←Ranked Matrix of Distance
ER← Elitist Rate for selecting best Individual from the Population
P_Indiv_k←Parent Individuals, Randomly generated from the Population based on crossover rate $where\ 0 < k \le 4$
TC_i←Total cost of the Population,$where\ 0 < i \le Popsize$
Ω_k ← Normalized values of different cities, $where\ 0 < k \le 4$

Assumptions

The present distance will be inform intermittently, among the city routes $City_i\ to\ City_j$, where $0 < i, j \le n$
Edge of Maximum Distance Max_Dist for all the cities are Persistent.
The first city is static in population generation.
TR Trade off Rate is set to 0.5.
ER Elitist Rate is set to 0.4.
Crossover Rate is set to 0.7
Mutation Rate is set to 0.1

$Fig.2\ List\ if\ varibales\ used\ in\ the\ Algorithm$

```
Population Initialization Algorithm
Step 1: Set the constraints $n$ ,$Max\_Gen$,$PopSize$ ,$Max\_Dist$
Step 2: Generate the First Population by ODV-EV method
            Step 2.1: Set the first city  $Cur\_city,\ Len \leftarrow 1, Size \leftarrow 1$
            Step 2.2: Enhance the first city in the Individual,  $Indiv[Len] = Init\_city$
            Step 2.2: Repeat through Step 2.8 Until  $Size \le PopSize$, else goto Step 3
            Step 2.3: Repeat through Step 2.4 Until  $Len \le n$, else goto Step 2.6
                        Produce a random number $'R' where\ 1 \le R \le ba$
                        $Cur\_City \leftarrow Indiv[Len]$// current city in the individual is assigned as current city
                        $Next\_city \leftarrow ODM(Cur\_city,R)$//extort the next city from the ODM
            Step 2.5:    IF$Next\_City \notin Indiv\ and\ POLL_{Cur\_City,Next\_City} < Max\_Poli$, else goto Step 2.3
            Step 2.6     $Len \leftarrow Len + 1$,Update$Indiv[Len] \leftarrow next\_City$
            Step 2.7:  Update  $POP_{Size} \leftarrow Indiv$//transfer the concluded individual to the population
            Step 2.8: $Size \leftarrow Size + 1$   //  goto the next individual
                              Step 3: Calculate the cost of each Individual in the Population
            $\forall[1, \le i \le PopSize], TC_i \leftarrow \sum_{j=1}^{n} DM(Indiv_i(j), Indiv_i(j+1)),\ \ j+1 \equiv 1$
Step 4: If $scenario = 1$ then,  $Pop \leftarrow Distance\_Algorithm(Pop,TC,G,n)$
Else If $scenario = 2\ then$,  $Pop \leftarrow Pollution\_Algorithm(Pop,TP,G,n)$
Else   $Pop \leftarrow Tradeoff\_Algorithm(Pop,TC,TP,G,n)$
End If
Step 5: Return$Pop$
```

$Fig.3\ Algorithmfor\ Population\ Initialization\ using\ EV\ method$

Where ER is the Elitism Rate, depending on that the number of elitist individuals are selected and passed to next generation populationPOP . $TC_1, TC_2, TC_3, \dots TC_{ER}$ are the minimum cost of each individual arranged in the increasing order and $Indiv_1, Indiv_2, Indiv_3, \dots Indiv_{ER}$ are the corresponding individuals to the minimum cost.

The crossover is done through the greedy crossover. Selected two random individuals from the total population as a parent individual. Since the starting city is the same for all individuals, the starting city is the initial city of the offspring. Find the position of the current city in both the parent individuals $Pos1 \leftarrow find\left(P_{indiv1}\left(Cur_{city}\right)\right),\ Pos2 \leftarrow find\left(P_{indiv2}\left(Cur_{city}\right)\right)$ and then locate the right side and left side city to the current city in both the parent individuals. If the position of current city is the starting city then lthe ocation of left city to the current city is last city $IF\ Pos1 = 1\ then\ LLoc1 \leftarrow n$ and if the position of current city is last city then location of Right city to the current city is last city $IF\ Pos1 = n\ then\ RLoc1 \leftarrow 1$. Check the cities in the locations are less than the maximum distance $DIST_{Cur\_City\ ,Next\_City} < Max_{dist}$. Based on the minimum cost the cities are added in offspring individual, after generated the complete individual move to the Mutation process, two random locations are generated and then swap the offspring individual cities in the locations and vice versa. The offspring is added into the population. Stop this process until we reach the total population.

$GeneC1 \leftarrow RAND\,(1,n,)$ (9)

$GeneC2 \leftarrow RAND\,(1,n,)$ (10)

$Swap\ Indiv(GeneC1) \leftarrow Indiv(GeneC2)$ (11)

- ***Algorithm Explanation***

In optimal distance based routing, our goal is to reduce the Total Cost(TC) , based on the cost of the individuals we are applying the genetic Algorithm to improve the effectiveness of the Distance based population. Get the Population, evaluated the cost of each individual in the population, number of generation, number of cities, maximum population and Assign the population into a temporary population.

Selection: In selection, our main idea is to pass the best individuals to the next generation. Select ER number of best individuals from the temporary population, whose cost should be minimum than the other individuals $position \leftarrow \min(TC)$. The best individuals are moved to the population $Pop_i \leftarrow TPop_{position}$, we are following elitism in the selection process.

Crossover: In this choose any two random Parent individuals within the total population. The first city in the parent individual is moved to the offspring individual first city. Assign the current city as current city in the offspring. Now find the position of the current city in both the parent individuals $Pos1 \leftarrow find\left(P_{indiv1}\left(Cur_{city}\right)\right),\ Pos2 \leftarrow find\left(P_{indiv2}\left(Cur_{city}\right)\right)$ and then locate the right side and left side city to the current city in both the parent individuals. If the position of current city is the starting city then location of left city to the current city is last city $IF\ Pos1 = 1\ then\ LLoc1 \leftarrow n$ and if the position of current city is last city then location of Right city to the current city is last city $IF\ Pos1 = n\ then\ RLoc1 \leftarrow 1$. Evaluate the cost of all the locations from the current city and choose the next city which is having the minimum cost. Check whether the next city is already present in the offspring Individual and also the distance between the current city and the next city should be minimum $Next\_City \notin Indiv\ and\ DIST_{Cur\_City\ ,Next\_City} < Max\_DIST$ and then add the city in the offspring Individual. Do these procedures until we are getting a complete offspring individual.

Mutation: In Mutation, Generate two random locations $GeneC1 \leftarrow RAND\,(1,n,), GeneC2 \leftarrow RAND\,(1,n)$and then swap the offspring individual cities in the locations$Swap\ Indiv(GeneC1) \leftarrow Indiv(GeneC2)$ and vice versa. Now add the offspring individual to the population, do this procedure till we reach the maximum number of temporary population. Evaluate the total cost of all the individuals from the whole population.

- *Algorithm*

```
Optimal Distance based routing Algorithm ($Pop, TC, G$, n)
Step 1: Initialize $Gen \leftarrow 1, i \leftarrow 0, k \leftarrow 0$
Step 2: Store the Population into a temporary variable, $CPop_{nxn} \leftarrow Pop_{nxn}$,
Step 2: Repeat through Step 10 Until $Gen \leq G$, else go to Step 10
Step 3: Select ER number of elitist individual which is having minimum cost
    Step 3.1: Repeat through Step 3.3Until $i < ER$, else goto Step 11
    Step 3.2: $position \leftarrow \min(TC)$ // Position of the Individual with Minimum cost value will be acquired.
    Step 3.3 $Pop_i \leftarrow CPop_{position}$ // the individual in the position in temporary population is moved to the population
Step 4: Repeat through Step 5.6 Until $k \leq Popsize$, else goto Step 6 $where\ ER < k \leq PopSize$
Step 5: Choose the random parents Individuals, $P\_Indiv1\ and\ P\_Indiv2$
    Step 5.1: Select the initial City $Init\_City,\ Length \leftarrow 1,\ Size \leftarrow 1$
    Step 5.2: $Indiv[Length] \leftarrow Init\_City$ //the first city of parent individual is the initial city in the offspring individual
    Step 5.3: Repeat through Step 5.5 Until $Length \leq n$ , else goto Step 5
        $Cur\_City \leftarrow Indiv[Length]$ //the present city in the offspring individual gave as present city
        Find the Position $Pos1\ and\ Pos2$ of the Current City in the Parent Individuals
        $Pos1 \leftarrow find(P\_indiv1\ (Cur\_city)),\ Pos2 \leftarrow find(P\_indiv2\ (Cur\_city))$
            IF $Pos1 = 1,\ LLoc1 \leftarrow n$
            Else IF $Pos1 = n,\ RLoc1 \leftarrow 1,$
            IF $Pos2 = 1,\ LLoc2 \leftarrow n$
            Else IF $Pos2 = n,\ RLoc2 \leftarrow 1,$
        Calculate the Distances $d1, d2, d3, d4$ from Preceding City to Present City and Present City to Subsequent City from the Parent Individuals
            $d_1 \leftarrow DM(P_{Indiv1(LLoc1)}, P_{Indiv1(Pos1)})$
            $d_2 \leftarrow DM(P_{Indiv1(Pos1)}, P_{Indiv1(RLoc1)})$
            $d_3 \leftarrow DM(P_{Indiv1(LLoc2)}, P_{Indiv1(Pos2)})$
            $d_4 \leftarrow DM(P\_Indiv1(Pos2), P\_Indiv1(RLoc2))$
        Step 5.4: Repeat through Step 5.6 Until $k < 4,\ else\ goto\ Step\ 6\ \ where\ 0 < k \leq 4$
            $LNext\_City \leftarrow min(d)$ //the position of the city with minimum cost will be acquired
            $Next\_City \leftarrow Indiv(LNext\_city)$
        Step 5.5: IF $Next\_City \notin Indiv\ and\ POLL_{Cur\_City,Next\_City} < Max\_Poll$, else goto Step 4.2
            $Length \leftarrow Length + 1$ , Update $Indiv[Length] \leftarrow next\_City$
        Step 5.6: $k \leftarrow k + 1$ // goto the subsequent individual
Step 6: Generate $GeneC1 \leftarrow RAND\ (1,n,), GeneC2 \leftarrow RAND\ (1,n)$
    $Swap\ Indiv(GeneC1) \leftarrow Indiv(GeneC2),\ Swap\ Indiv(GeneC2) \leftarrow Indiv(GeneC1)$
Step 7:, $Pop_{size} \leftarrow Indiv$ , $Size \leftarrow Size + 1$ // goto the subsequent individual
Step 8: Calculate the cost of each Individual in the Population
    $\forall[1, \leq i \leq PopSize], TC_i \leftarrow \sum_{j=1}^{n} DM(Indiv_i(j), Indiv_i(j+1)),\ \ j+1 \equiv 1$
Step 9: $Gen \leftarrow Gen + 1$ //Present generation is concluded, raise the Gen for subsequent generation
Step 10: Return $Pop$
```

IV. EXPERIMENTATION METHODOLOGY

A. Experimental Set-up

We have performed the experimental analysis for optimal distance based scenario with the different instances. Our method is coded in Matlab, version7.12 and tested on Windows 7, i7 CPU @ 3.4 GHz processor. All implementation are done using MATLAB with TSP benchmark datasets obtained from TSPLIB. The result was analyzed with TSP benchmarks, for the instances ulysses16, ulysses22, bays29, att48, eil56, eil76, and kroA100. We fixed the number generation, the total population in each generation as 100. We had done the experiment with three initialization technique and analyzed with the average of 50 runs of each technique with a different scenario.

B. Experimental Scenarios

- ***Random Technique:***

This is a simple technique to initialize the initial population, in this technique, there is no order which city will be added as next city in the individual (i.e.) a randomly generated city has been added. In our experiment, we have been included one criteria, $DIST_{Cur\_City,Next\_City} < Max\_DIST$ the distance between the current city and the next city to be added should be less than the maximum solution while generation the initial solution. As we discussed in section 3, we implemented the algorithms in this technique and analyzed.

1) Nearest Neighbor (NN) technique:

Nearest neighbor is a constructive heuristic, this is the one of the techniques used to initialize the initial population, it is easy to implement and the quickly yields an effectively short route. First, it will start from the random city and then choose the nearest city as the next city. In our Experimental implementation, it will choose the next nearest city and it should satisfy our criteria that, the distance between the current city and nearest city should be minimum. We implemented the three algorithms that we discussed in section 3 with this technique.

- **ODV –EV Technique:**

Already we discussed this technique in detail in section 2, it is an effective population initialization technique. Based on the ‘ba’ value it will select the next city from the distance matrix and then added to the initial population. But the next city should satisfy the condition that the distance between the current city and the next city should be less than the maximum distance.

C. Validation Criteria

In our proposal, we have analyzed different performance factors, based on the optimal solution

Convergence rate: convergence rate of the solution can be calculated based on the percentage of fitness and the optimal fitness of the problem. Using this equation we derived convergence rate for Cost equation (1).

$$Convergence\ Rate(\%) = \left(1 - \frac{Fitness\ cost - Optimal\ cost}{Optimal\ cost}\right) * 100 \quad (12)$$

Average Convergence rate: It is defined by the difference between the average fitness and the optimal fitness of the problem. It is given by,

$$Average\ Convergence(\%) = \left(1 - \frac{Average\ fitness\ + Optimal\ fitness}{Optimal\ fitness}\right) * 100 \quad (13)$$

Error rate: It is the percentage of difference in fitness and optimal fitness of the solution in the population. From that we derived two equations (27) and (28) which is given by,

$$Error\ Rate\ for\ Cost = \left(\frac{Fitness\ - Optimal\ fitness}{Optimal\ fitness}\right) * 100 \quad (14)$$

$$Error\ Rate\ for\ Cost = \left(\frac{Fitness\ cost - Optimal\ cost}{Optimal\ cost}\right) * 100 \quad (15)$$

$$Error\ Rate\ for\ Pollution = \left(\frac{Fitness\ Pollution - Optimal\ Pollution}{Optimal\ Pollution}\right) * 100 \quad (16)$$

Convergence diversity: convergence diversity is referred by the variation of the best individual in the population and worst individual in the population. From that, we derived one equation (18) for our problem.

$$Convergence\ diversity\ (\%) = CR_{Highest} - CR_{Least} \quad (17)$$

$$Convergence\ diversity\ for\ Cost\ (\%) = CRC_{Highest} - CRC_{Least} \quad (18)$$

V. EXPERIMENTAL ANALYSES

A. Optimal Distance Based Routing in TSP

As we discussed in section 3, In Optimal Distance Based Routing we are implementing a distance based routing. We are moving this problem towards a single objective, distance. From the generated initial population we achieved the optimal distance base routing based on the distance. Here we are analyzing that with different population initialization technique Random initialization technique, NN Technique and ODV-EV technique with different TSP instances.

Random initialization technique: we implemented our algorithms in this technique Table I illustrates that how this technique performing in three scenarios with different TSP instances.

Nearest Neighbor (NN) technique: in this technique we carried out our proposed system. We analyzed how the three routing algorithm is performing in this technique with different TSP instances, it is described in Table I.

ODV-EV technique: ODV –EV is an efficient technique to initialize the initial population, from this we can obtain the optimal solution. We executed the proposed algorithms in this technique and the Table I illustrates, how this technique working in three algorithms with different TSP instances.

B. Result Analyses

Remark 1: From Fig. 6 and 8 we analyzed that, The best convergence rate and the worst convergence rate are high in ODV-EV technique when compare to the other techniques and the error rate is also less in this technique showed in Fig.7 and 9. Because the distance in the path between the current city and the next minimum distance city is high. It will move to the next minimum distance city. So the convergence rate is high and the error rate is less.

Remark 2: the best convergence rate of NN technique is less than random technique and the worst convergence rate of the NN technique is better than random technique.

Remark 3: During the initial population the convergence diversity should be high, in the final generation the convergence diversity should be less. The convergence diversity is low in the ODV-EV method than the other two techniques shown in the Fig.12.

Remark 4: The average convergence is giving good results in ODV –EV technique is better than the other two techniques. The average convergence of random technique is high than NN technique is showed in the Fig.10.

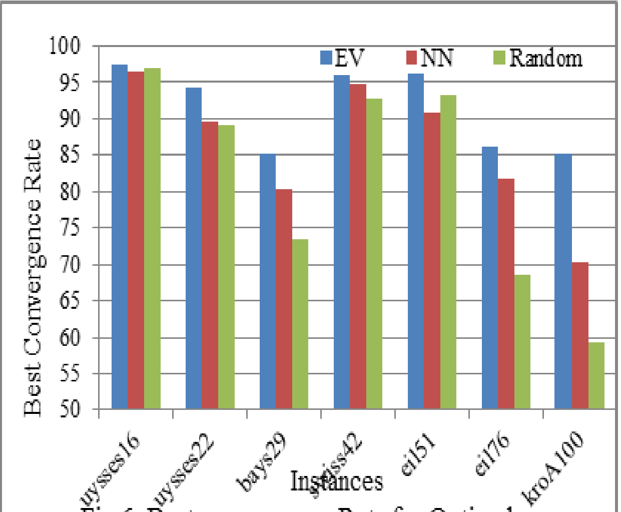


Fig.6. Best convergence Rate for Optimal Distance Based Routing

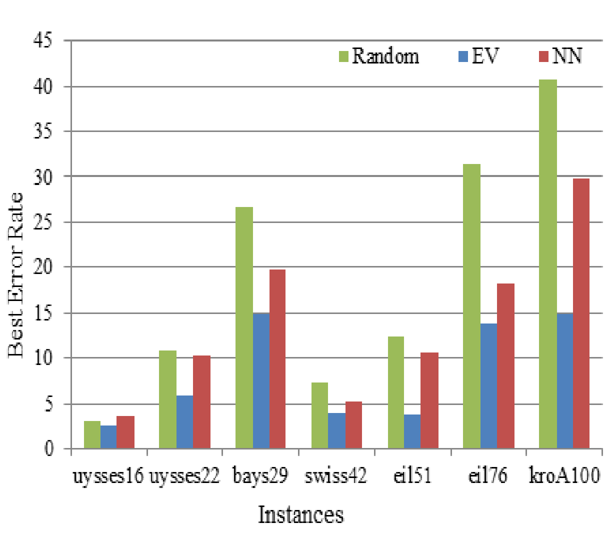


Fig.7. Best Error Rate for Optimal Distance Based Routing

Table I Result Analysis of Optimal Distance Based Routing

Instance	Seeding Technique	Optimal Solution (Distance)	Computation Time	Quality of the Solution			Convergence Rate (%)		Error Rate (%)		Convergence Diversity	Average Convergence
				Best	Worst	Average	Best	Worst	Best	Worst		
uysses16	**EV**	74.1087	10.8	76.048	124.436	110.259	97.383	32.091	2.617	67.909	65.293	51.219
	NN	74.1087	10.8	76.766	133.315	115.263	96.415	20.109	3.585	79.891	76.306	44.468
	Random	74.1087	10.66	76.394	125.972	108.641	96.916	30.017	3.084	69.983	66.899	53.403
uysses22	**EV**	75.6615	16.42	80.084	144.88	127.428	94.154	8.515	5.846	91.485	85.639	31.581
	NN	75.6615	16.67	83.463	151.628	126.768	89.689	-0.403	10.311	100.403	90.092	32.454
	Random	75.6615	16.55	83.814	151.434	128.305	89.225	-0.147	10.775	100.147	89.372	30.423
bays29	**EV**	2020	20.65	2319	4196.6	3791.736	85.198	-7.752	14.802	107.752	92.95	12.29
	NN	2020	22.07	2420	4329.2	3799.374	80.198	-14.317	19.802	114.317	94.515	11.912
	Random	2020	21.91	2558.2	4719.4	3933.75	73.356	-33.634	26.644	133.634	106.99	5.26
swiss42	**EV**	1273	22.25	1324.222	1933.556	1694.034	95.976	48.11	4.024	51.89	47.866	66.926
	NN	1273	22.5	1339.6	2048.8	1702.51	94.768	39.057	5.232	60.943	55.711	66.26
	Random	1273	22.82	1365	2021.8	1758.182	92.773	41.178	7.227	58.822	51.595	61.887
EIL51	**EV**	426	28.75	442.062	833.087	603.879	96.23	4.44	3.77	95.56	91.79	58.244
	NN	426	29.98	464.658	699.113	579.155	90.925	35.889	9.075	64.111	55.037	64.048
	Random	426	28.86	454.848	667.031	585.117	93.228	43.42	6.772	56.58	49.808	62.649
eil76	**EV**	538	42.98	612.614	908.158	769.134	86.131	31.197	13.869	68.803	54.934	57.038
	NN	538	43.94	636.223	928.683	800.029	81.743	27.382	18.257	72.618	54.361	51.296
	Random	538	46.1	707.045	1025.886	904.237	68.579	9.315	31.421	90.685	59.264	31.926
kroA100	**EV**	21285	56.21	24432.547	37271.678	33334.141	85.212	24.892	14.788	75.108	60.32	43.391
	NN	21285	57.68	27629.297	45023.254	35956.651	70.194	-11.526	29.806	111.526	81.719	31.07
	Random	21285	60.82	29955.978	50561.596	40854.846	59.262	-37.546	40.738	137.546	96.808	8.058

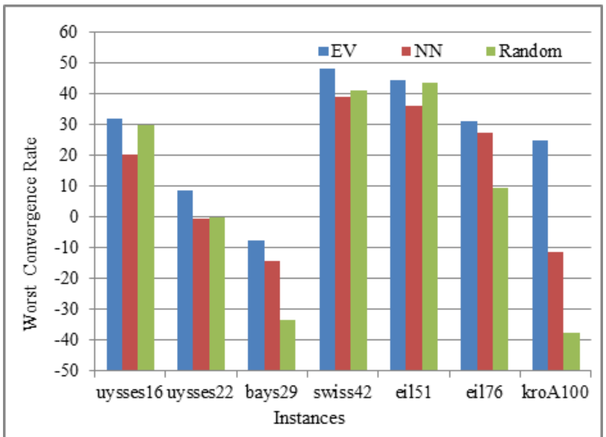


Fig.8. Worst convergence Rate for Optimal Distance Based Routing

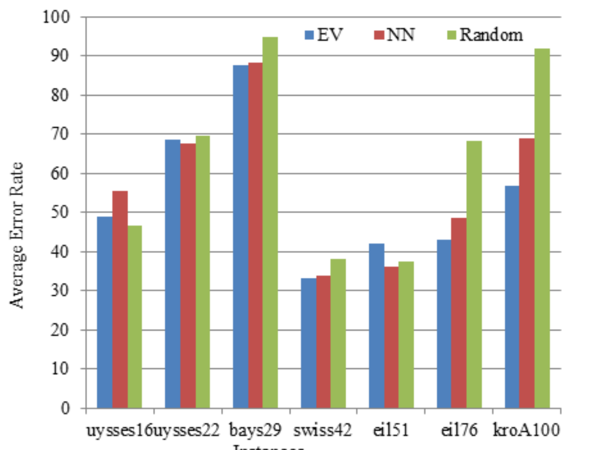


Fig.11. Average Error Rate for Optimal Distance Based Routing

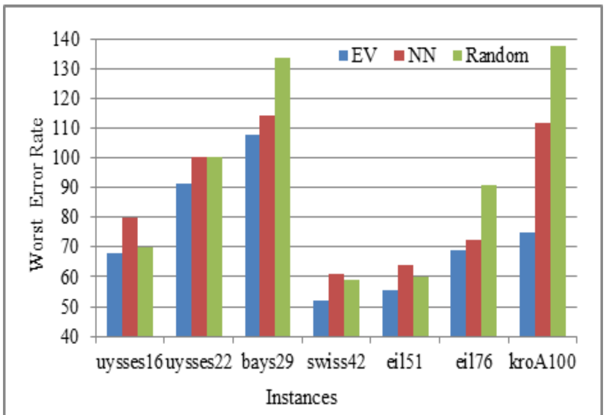


Fig.9. Worst Error Rate for Optimal Distance Based Routing

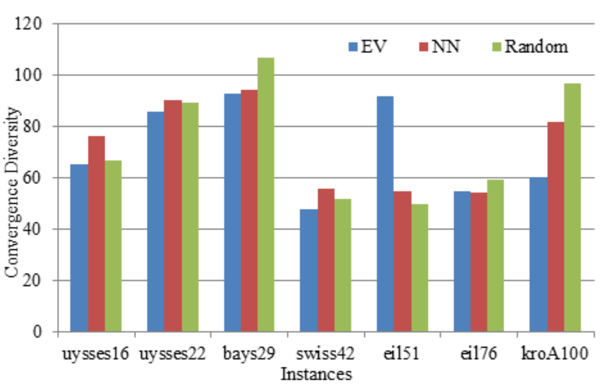


Fig.12. Convergence Diversity for Optimal Distance Based Routing

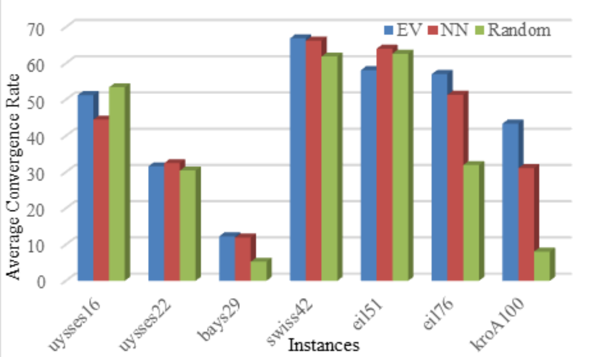


Fig.10. Average convergence Rate for Optimal Distance Based Routing

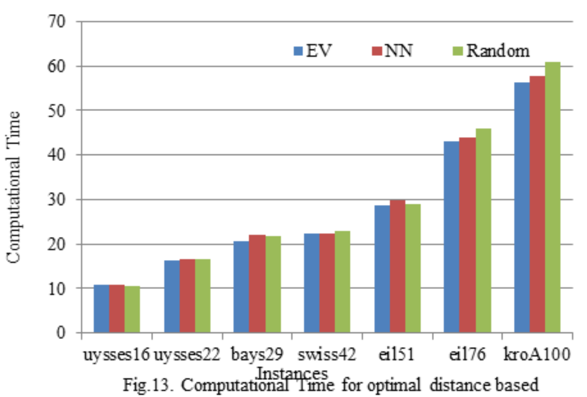


Fig.13. Computational Time for optimal distance based routing

VI. CONCLUSION

In this research, an intelligent routing strategy for VRP has been proposed based on the values of the distance between the corresponding cities. The proposed routing strategy uses a recent enhanced model of Genetic Algorithm to find the optimal route under optimal distance based routing scenario.

Experiments were performed on the benchmark TSP instances obtained from the TSPLIB. The performance of the intelligent routing strategy for VRP is investigated using three population seeding techniques namely random, nearest neighbor and ODV based EV method. In the viewpoint of effective intelligent routing strategy for VRP, the maximum combined convergence rate obtained for the route is 97.383% for the instance uysses using ODV-EV population seeding technique with a computation time of 10.8. The results of the final model show that we achieved optimal distance route path for an effective Integrated Routing Strategy in Transportation Systems.

REFERENCES

1. Ren Y, Dessouky M, & Ordóñez F, (2010)The multi-shift vehicle routing problem with overtime. Computers and Operations Research 37: 1987–1998.
2. Thierry Garaix, Christian Artigues, Dominique Feillet, Didier Josselin, (2010) Vehicle routing problems with alternative paths: An application to on-demand transportation, European Journal of Operational Research 204: 62–75.
3. YiyoKuo, Chi-Chang Wang, (2012) A variable neighborhood search for the multi-depot vehicle routing problem with loading cost, Expert Systems with Applications 39: 6949–6954
4. Ghiani G, Guerriero F, Laporte G, Musmanno R, (2003) Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies, European Journal of Operational Research 151: 1–11.
5. Mostepha, Khouadjia R, Briseida Sarasola, Enrique Alba, LaetitiaJourdan, El GhazaliTalbi, (2012) A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests, Applied Soft Computing 12: 1426–1439.
6. Milthers N.P.M., (2009) Solving VRP using Voronoi Diagrams and Adaptive Large Neighborhood Search.Master's thesis, University of Copenhagen,
7. Pisinger D, Ropke S, (2005) A General Heuristic for Vehicle Routing Problems. Technical Report, DIKU-Department of Computer Science, University of Copenhagen. <http://www.diku.dk/hjemmesider/ansatte/sropke/Papers/GeneralVRP_TechRep.pdf(15.09.13)>.
8. Pisinger D, Ropke S, (2007) A general heuristic for vehicle routing problems. Computers & Operations Research 34 (8): 2403–2435.
9. Ropke S, Pisinger D, (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40 (4): 455–472.
10. Shanmugam M and Amudhavel J, (2017) Revenant of the Ecosystem: An Environmental based Green Computing Models for Vehicular Routing Problems using Genetic Algorithm Optimization Approach", IIOABJ - Special issue on computer Science, 8(2):262-273.
11. Shanmugam M, Saleem Basha MS, Dhavachelvan P, Baskaran R (2013b) Performance Assessment over Heuristic Population Seeding Techniques of Genetic Algorithm: Benchmark Analyses on Traveling Salesman Problems. International Journal of Applied Engineering Research, Research India Publications, 8(10): 1171-1183.
12. Shanmugam M, Jayakumar L, Anand T, Rajaguru D, Chandramohan D, Amudhavel J, (2018) Air Pollution Based Vehicular Routing Problems: Using Genetic Algorithm Optimization Approach, 27(106): 1575-1587

AUTHORS PROFILE

Mr. K. Premkumar, pursed Bachelor degree from Adhipara Sakthi Engineering college and Master degree in Computer Science and Engineering from Sathyabama Deemed University, Chennai. He is pursuing his P.hd in the field of Vanet at Manonmaniam Sundaranar University, Tirunelveli. He has 16 years of teaching experience.

Dr R. Baskaran, pursed Bachelor degree from Pondicherry University Puducherry and Master degree from Vellore Engineering College. He completed his Reserch work in the field of Inforamtion Retrieval from Anna University, Chennai. He is currently working as Professor in Computer Science and Engineering depatment and he has 5 years of Research Experience and 12 years of teaching experience.

Dr. M. Shanmugam, working as an Associate Professor in the department of Computer Science & Engineering, Vignan's Foundation for Science, Technology, and Research, Guntur, INDIA.