Hindusthan International Journal of Multidisciplinary Research (HIJOMR)
Volume 1 Issue 1, September 2024, pp.47-67

ESTIMATING SHORTEST ROUTE FOR NAVA DIVYA DESAMS

Dr. S.Anuradha[1], S.Renuga Devi[2], Dr.D.Mallika[3], P.Rekha[4]

[1]Professor & Head, Department of Mathematics, Data Science & Analytics
Hindusthan College of Arts & Science, Coimbatore.
[1]*anuradha.s@hicas.ac.in*

[2,4]Assistant Professor, [3]Professor,
Department of Mathematics,
Hindusthan College of Arts & Science,
Behind Nava India,Coimbatore
[2]*srenugasri@gmail.com*, [3]*mallika.d@hicas.ac.in*, [4]*rekha.p@hicas.ac.in*

ABSTRACT

In this paper Diagonal Completion technique is applied to solve the Travelling Salesman Problem (TSP) whose objective is to minimize the cost. Given a set of cities and the distance between every pair of cities, the problem of finding the shortest route between a set of points and locations that must be visited. When engaging on a spiritual journey, one prefers shortest route (minal distance) to travel Nava Divya Desams namely Srirangam, Thirupathi, Srivaikuntam, Azhagar Kovil (Madurai), Thiruvananthapuram (Kerala), Kanchipuram, Thiruvellarai, Thirukoshtiyur, Srivilliputhur. Our problem is not possible to solve by penalty method because five cities are not covered in the route. So we solve the problem by Diagonal Completion technique. In this method we found the route to cover Nava Divya Desams in two sub tours.

Keywords: *Divya Desams, Interval numbers, Branch and Bound Technique, Diagonal Completion Techniques.*

1. DIVYA DESAMS

Divya Desams are 108 Vishnu temples that are mentioned in the works of the Alvars, the Tamil poet-saints of the 6^{th}-9^{th} centuries AD. These temples are of great significance in the Vaishnavism tradition of Hinduism and are spread across India, predominantly in the Tamil Nadu region, with others in Andhra Pradesh, Kerala, Gujarat, and Uttar Pradesh, and one in Nepal. There are 84 Divya Desams in Tamil Nadu, and 11 in the northern and southern areas of Kerala. 105 of the 108 Divya Desams are dispersed over the Indian subcontinent's states, including Tamil Nadu, Kerala, Andhra Pradesh, Gujarat, Uttar Pradesh, and Uttarakhand. While one is in Nepal, the other two are beyond earthly worlds, namely Thirupalkadal (the ocean of milk) and Paramapadam (Vaikuntha, where Lord Vishnu resides).

1.1 History of Divya Desams:

- **Alvars and their Hymns**: The Alvars, twelve in number, were ardent devotees of Lord Vishnu. They composed hymns in praise of Vishnu, which were later compiled into a collection known as the "NalayiraDivyaPrabandham." The hymns dedicated to the temples are considered to mark them as Divya Desams.

- **Geographical Spread**: The majority of Divya Desams are located in Tamil Nadu, with significant clusters in and around the temple towns of Kanchipuram, Tiruchirapalli, and Madurai. The rest are scattered across various states in India and one in Nepal, emphasizing the widespread influence of Vaishnavism.
- **Architectural Significance**: The temples are renowned for their architectural beauty and historical significance. They feature Dravidian architecture, characterized by towering gopurams (gateway towers), intricately carved pillars, and expansive courtyards.
- **Religious Practices**: The Divya Desams follow specific rituals and traditions. The daily worship and festivals in these temples are conducted according to the Pancharatra Agama and Vaikhanasa Agama scriptures.
- **Pilgrimage**: Visiting the 108 Divya Desams is considered a sacred pilgrimage for devotees of Lord Vishnu. It is believed that visiting these temples and offering prayers can lead to salvation (moksha).
- **Prominent Divya Desams**:

  Srirangam (Sri Ranganathaswamy Temple): Located in Tiruchirapalli, Tamil Nadu, it is the largest functioning Hindu temple in the world.

  Tirupati (Sri Venkateswara Temple): Located in Andhra Pradesh, it is one of the most visited pilgrimage sites in the world.

  Srivilliputhur (SrivilliputhurAndal Temple): Known for its connection to Andal, the only female Alvar.

  Thiruvananthapuram (Padmanabhaswamy Temple): Located in Kerala, it is famous for its wealth and the reclining form of Vishnu.

  The Divya Desams hold a cherished place in the spiritual landscape of India, representing the rich cultural and devotional heritage of Vaishnavism.

2. TRAVELING SALESMAN PROBLEM

The Travelling Salesman Problem (TSP) is one of the most researched problems in optimization and computer science. It is concerned with determining the shortest possible route for a traveling salesman to visit each city on a given list exactly once before returning to the starting point. Despite its basic concept, TSP is well-known for its computational complexity and diverse variety of practical applications. The TSP has remained a benchmark problem in optimization and computer research. Solving it efficiently has far-reaching consequences for industries ranging from logistics and transportation to network design and bioinformatics. Its research has also resulted in advances in algorithm design and computational complexity theory.

2.1 Problem Definition:

The Travelling Salesman Problem (TSP) is a well-known problem in optimization and computer science.

Given a list of cities and the distances between each pair of cities, the task is to find the shortest possible route that visits each city exactly once and returns to the original city.

Key Components:

Cities: A set of locations that need to be visited.

Distances: The distance (or cost) between each pair of cities.

Objective: The objective is to minimize the total travel distance (or cost) of the tour that visits each city exactly once and returns to the starting city.

2.2 Mathematical Formulation:

Decision Variables: Let x_{ij} be a binary variable that equals 1 if the tour goes directly from city i to city j, and 0 otherwise.

Objective Function: Minimize the total distance:

$$Minimize\ Z = \sum_{i,j=1}^{n} d_{ij}\, x_{ij}$$

Where d_{ij} is the distance between cities i and j and n is the number of cities.

2.3 Constraints:

Each City is visited Exactly Once: For each city,there must be exactly one incoming and one outgoing edge.

$x_{ij} = 1$ for all i ,j ≠ i $x_{ij} = 1$ for all i ,i ≠ j

2.4. Solution Methods:

- **Exact Algorithms**:

Brute Force: Check all possible permutations of cities. This is feasible only for very small numbers of cities due to its factorial time complexity.

Dynamic Programming: The Held-Karp algorithm is a dynamic programming approach that reduces the time complexity to $O(n^2\, 2^n)$ but it is still exponential in nature.

Integer Linear Programming (ILP): Use linear programming formulations with integer constraints and solve using solvers like CPLEX or Gurobi.

- **Heuristic Methods**:

Nearest Neighbour: Start from a city and repeatedly visit the nearest unvisited city. This is simple but does not guarantee the optimal solution.

Greedy Algorithm: Build a tour incrementally by choosing the shortest available edge at each step.

Genetic Algorithms: Use evolutionary strategies to find good solutions through operations like mutation and crossover.

Simulated Annealing: Search for better solutions by probabilistically accepting worse solutions with the hope of escaping local optima.

- **Approximation Algorithms**:

Christofides' Algorithm: Provides a solution that is within 1.5 times the optimal tour length for metric TSP, where distances satisfy the triangle inequality.

2.5 Applications:

Logistics and Route Planning: Optimizing delivery routes for salespeople, delivery trucks, or other vehicles.

Manufacturing: Optimizing the path of a drill or other tools on a circuit board.

Computer Networking: Designing efficient network layouts.

3. BRANCH AND BOUND

Branch and Bound (B&B) is a popular algorithmic technique for solving combinatorial optimization problems like the Travelling Salesman Problem (TSP). It is a general method for finding optimal solutions by systematically exploring and pruning the solution space.

Branch and Bound works by breaking the problem into smaller subproblems (branching) and using bounds to determine which subproblems need to be explored further (bounding). It is particularly effective for problems with a large but finite solution space, where a brute-force approach would be computationally infeasible.

3.1 Steps in Branch and Bound for TSP:

Initialization:

Start with an initial upper bound on the solution (often obtained through a heuristic or a simple greedy algorithm).Initialize the best known solution as the initial upper bound.

Branching:

Divide the problem into smaller sub problems. For the TSP, this involves creating subsets of cities and constructing partial tours. Each branch represents a decision point, such as including a specific edge in the tour or not.This can be done by considering all possible edges that can be added to the tour and creating sub problems based on these choices.

Bounding:

For each sub problem, compute a lower bound on the cost of the tour that can be achieved from that sub problem.Various bounding techniques can be used, such as the Minimum Spanning Tree (MST) or the Linear Programming relaxation of TSP.If the lower bound of a sub problem is greater than or equal to the current best known solution, discard that sub problem (pruning).

Pruning:

Discard branches where the lower bound exceeds the current best solution or where constraints cannot be met.This helps in reducing the number of sub problems that need to be explored.

Updating:

Update the best known solution if a better solution is found in any sub problem.

Termination:

The algorithm terminates when all branches have been either explored or pruned, and the best known solution is the optimal solution.

4. DIAGONAL COMPLETION METHOD:

The diagonal completion method is a technique used in the field of operations research, specifically in the context of optimization problems, such as the assignment problem. This method helps in finding an optimal solution by systematically improving a feasible solution until optimality is achieved.The Travelling Salesman Problem (TSP) is a classic optimization problem in operations research where the goal is to find the shortest possible route that visits a set of cities and returns to the origin city. Although the diagonal completion method is typically used for assignment problems, it can be adapted for solving TSP in certain contexts.

Key Concepts

Travelling Salesman Problem (TSP): A problem where a salesman must visit each city exactly once and return to the starting city, minimizing the total travel cost or distance.

Cost Matrix: A matrix where the entry C_{ij} represents the cost or distance of traveling from city i to city j .

4.1 Steps in Applying the Diagonal Completion Method to TSP

Construct the Cost Matrix: Create a cost matrix where each entry represents the travel cost between two cities.

Initial Feasible Solution: Start with an initial feasible solution, which can be generated using heuristics such as the Nearest Neighbor, Minimum Spanning Tree, or any other TSP heuristic.

Cost Adjustment: Similar to the assignment problem, adjust the cost matrix to facilitate finding an optimal tour. This may involve reducing the cost matrix by subtracting the smallest value in each row and column.

Optimization by Diagonal Completion: Improve the current feasible solution by examining potential swaps and reassignments to reduce the total tour cost. In the context of TSP, this may involve completing cycles in the cost matrix that correspond to potential improvements in the tour.

Iterative Improvement: Repeatedly adjust the cost matrix and reassign paths until no further improvements can be made, indicating an optimal or near-optimal solution has been found.

5. DISTANCE BETWEEN NAVA DIVYA DESAMS

Table 1: Collected Data Details-The distance between Nava divya desams

Distance / Place	Srirangam	Tirupathi	Sri Vaikuntam	Azhagarkovil-Madurai	Thiruvananthapuram-Kerala	Kanchipuram	Thiruvellarai	Thirukoshtiyur	Srivilliputhur
Srirangam	∞	370	312	124	460	274	16	112	223
Tirupathi	370	∞	679	492	826	108	400	478	589
Sri Vaikuntam	312	679	∞	195	177	590	397	237	141
Azhagarkovil - Madurai	124	492	195	∞	344	403	138	54	106
Thiruvananthapuram-Kerala	460	826	177	344	∞	730	474	385	184
Kanchipuram	274	108	590	403	730	∞	282	388	500
Thiruvellarai	16	400	397	138	474	282	∞	124	235
Thirukoshtiyur	112	478	237	54	385	388	124	∞	147
Srivilliputhur	223	589	141	106	184	500	235	147	∞

Consider: Srirangam (A), Tirupathi (B), Sri Vaikuntam (C), Azhagarkovil- Madurai (D), Thiruvananthapuram-Kerala (E), Kanchipuram (F), Thiruvellarai (G), Thirukoshtiyur (H), Srivilliputhur (I).

Find Solution of Travelling salesman problem using branch and bound (penalty) method (MIN case)

Place \ Distance	A	B	C	D	E	F	G	H	I
A	x	370.	312	124	460	274	16	112	223
B	370	x	679	492	826	108	400	478	589
C	312	679	x	195	177	590	397	237	141
D	124	492	195	x	344	403	138	54	106
E	460	826	177	344	x	730	474	385	184
F	274	108	590	403	730	x	282	388	500
G	16	400	397	138	474	282	x	124	235
H	112	478	237	54	385	388	124	x	147
I	223	589	141	106	184	500	235	147	x

Solution:

The number of rows = 9 and columns = 9

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	
A	M	370	312	124	460	274	16	112	223
B	370	M	679	492	826	108	400	478	589
C	312	679	M	195	177	590	397	237	141
D	124	492	195	M	344	403	138	54	106
E	460	826	177	344	M	730	474	385	184
F	274	108	590	403	730	M	282	388	500
G	16	400	397	138	474	282	M	124	235
H	112	478	237	54	385	388	124	M	147
I	223	589	141	106	184	500	235	147	M

Steps of penalty methods, which is not to solve the problem.

We know that the sum of row minimum gives us the lower bound.

Step-1: Find out the each row minimum element and subtract it from that row

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	
A	M	354	296	108	444	258	0	96	207	(-16)
B	262	M	571	384	718	0	292	370	481	(-108)
C	171	538	M	54	36	449	256	96	0	(-141)
D	70	438	141	M	290	349	84	0	52	(-54)
E	283	649	0	167	M	553	297	208	7	(-177)
F	166	0	482	295	622	M	174	280	392	(-108)
G	0	384	381	122	458	266	M	108	219	(-16)
H	58	424	183	0	331	334	70	M	93	(-54)
I	117	483	35	0	78	394	129	41	M	(-106)

So, row minimum will be 780. (16+108+141+54+177+108+16+54+106=780)

Step-2: Find out the each column minimum element and subtract it from that column.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
A	M	354	296	108	408	258	0	96	207
B	262	M	571	384	682	0	292	370	481
C	171	538	M	54	0	449	256	96	0
D	70	438	141	M	254	349	84	0	52
E	283	649	0	167	M	553	297	208	7
F	166	0	482	295	586	M	174	280	392
G	0	384	381	122	422	266	M	108	219
H	58	424	183	0	295	334	70	M	93
I	117	483	35	0	42	394	129	41	M
	(-0)	(-0)	(-0)	(-0)	(-36)	(-0)	(-0)	(-0)	(-0)

So, column minimum will be 36. (0+0+0+0+36+0+0+0+0=36)
we get the lower bound = 780+36=816

Hindusthan International Journal of Multidisciplinary Research (HIJOMR)
Volume 1 Issue 1, September 2024, pp.47-67

Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
A	*M*	354	296	108	408	258	0(166)	96	207
B	262	*M*	571	384	682	0(520)	292	370	481
C	171	538	*M*	54	0(42)	449	256	96	0(7)
D	70	438	141	*M*	254	349	84	0(93)	52
E	283	649	0(42)	167	*M*	553	297	208	7
F	166	0(520)	482	295	586	*M*	174	280	392
G	0(166)	384	381	122	422	266	*M*	108	219
H	58	424	183	0(58)	295	334	70	*M*	93
I	117	483	35	0(35)	42	394	129	41	*M*

Here maximum penalty is 520, occur at $X_{B,F}$ or $X_{F,B}$, so we choose $X_{F,B}$ to begin branch
There are two branches.
1. If $X_{F,B}=0$, then we have an additional cost of 520 and the lower bound becomes 816+520=1336
2. If $X_{F,B}=1$, we can go $F\rightarrow B$
So we can't go $B\rightarrow F$, so set it to M.
Now we leave row *F* and column *B*, so reduced matrix is

	A	C	D	E	F	G	H	I
A	M	296	108	408	258	0	96	207
B	262	571	384	682	M	292	370	481
C	171	M	54	0	449	256	96	0
D	70	141	M	254	349	84	0	52
E	283	0	167	M	553	297	208	7
G	0	381	122	422	266	M	108	219
H	58	183	0	295	334	70	M	93
I	117	35	0	42	394	129	41	M

Hindusthan International Journal of Multidisciplinary Research (HIJOMR)
Volume 1 Issue 1, September 2024, pp.47-67

Step-1: Find out the each row minimum element and subtract it from that row

	A	*C*	*D*	*E*	*F*	*G*	*H*	*I*	
A	M	296	108	408	258	0	96	207	(-0)
B	0	309	122	420	M	30	108	219	(-262)
C	171	M	54	0	449	256	96	0	(-0)
D	70	141	M	254	349	84	0	52	(-0)
E	283	0	167	M	553	297	208	7	(-0)
G	0	381	122	422	266	M	108	219	(-0)
H	58	183	0	295	334	70	M	93	(-0)
I	117	35	0	42	394	129	41	M	(-0)

So, row minimum will be 262. (0+262+0+0+0+0+0+0=262).

Step-2: Find out the each column minimum element and subtract it from that column.

	A	*C*	*D*	*E*	*F*	*G*	*H*	*I*
A	M	296	108	408	0	0	96	207
B	0	309	122	420	M	30	108	219
C	171	M	54	0	191	256	96	0
D	70	141	M	254	91	84	0	52
E	283	0	167	M	295	297	208	7
G	0	381	122	422	8	M	108	219
H	58	183	0	295	76	70	M	93
I	117	35	0	42	136	129	41	M
	(-0)	(-0)	(-0)	(-0)	(-258)	(-0)	(-0)	(-0)

So, column minimum will be 258. (0+0+0+0+258+0+0+0=258)
We get the lower bound = 816+262+258=1336
Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	A	*C*	*D*	*E*	*F*	*G*	*H*	*I*
A	*M*	296	108	408	0(8)	0(30)	96	207
B	0(30)	309	122	420	*M*	30	108	219
C	171	*M*	54	0(42)	191	256	96	0(7)
D	70	141	*M*	254	91	84	0(93)	52
E	283	0(42)	167	*M*	295	297	208	7
G	0(8)	381	122	422	8	*M*	108	219
H	58	183	0(58)	295	76	70	*M*	93
I	117	35	0(35)	42	136	129	41	*M*

Here maximum penalty is 93, occur at *XD,H* , so we choose *XD,H* to begin branch
There are two branches.
1. If *XD,H*=0, then we have an additional cost of 93 and the lower bound becomes 1336+93=1429
2. If *XD,H*=1,
we can go *D*→*H*
So we can't go *H*→*D*, so set it to M.
Now we leave row *D* and column *H*, so reduced matrix is

	A	*C*	*D*	*E*	*F*	*G*	*I*
A	M	296	108	408	0	0	207
B	0	309	122	420	M	30	219
C	171	M	54	0	191	256	0
E	283	0	167	M	295	297	7
G	0	381	122	422	8	M	219
H	58	183	M	295	76	70	93
I	117	35	0	42	136	129	M

Step-1: Find out the each row minimum element and subtract it from that row

	A	*C*	*D*	*E*	*F*	*G*	*I*	
A	M	296	108	408	0	0	207	(-0)
B	0	309	122	420	M	30	219	(-0)
C	171	M	54	0	191	256	0	(-0)
E	283	0	167	M	295	297	7	(-0)
G	0	381	122	422	8	M	219	(-0)
H	0	125	M	237	18	12	35	(-58)
I	117	35	0	42	136	129	M	(-0)

So, row minimum will be 58. (0+0+0+0+0+58+0=58)
we get the lower bound = 1336+58+0=1394
Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	A	*C*	*D*	*E*	*F*	*G*	*I*
A	*M*	296	108	408	0(8)	0(12)	207
B	0(30)	309	122	420	*M*	30	219
C	171	*M*	54	0(42)	191	256	0(7)
E	283	0(42)	167	*M*	295	297	7
G	0(8)	381	122	422	8	*M*	219
H	0(12)	125	*M*	237	18	12	35
I	117	35	0(89)	42	136	129	*M*

There are two branches.
1. If *XI,D*=0, then we have an additional cost of 89 and the lower bound becomes 1394+89=1483
2. If *XI,D*=1,
we can go *I*→*D*
Now we leave row *I* and column *D*, so reduced matrix is

	A	*C*	*E*	*F*	*G*	*I*
A	M	296	408	0	0	207
B	0	309	420	M	30	219
C	171	M	0	191	256	0
E	283	0	M	295	297	7
G	0	381	422	8	M	219
H	0	125	237	18	12	35

Here we have 0 in every row and column. So, the lower bound remains the same
i.e, 1394+0=1394
Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	A	*C*	*E*	*F*	*G*	*I*
A	*M*	296	408	0(8)	0(12)	207
B	0(30)	309	420	*M*	30	219
C	171	*M*	0(237)	191	256	0(7)
E	283	0(132)	*M*	295	297	7
G	0(8)	381	422	8	*M*	219
H	0(12)	125	237	18	12	35

Here maximum penalty is 237, occur at $X_{C,E}$, so we choose $X_{C,E}$ to begin branch
There are two branches.
1. If $X_{C,E}=0$, then we have an additional cost of 237 and the lower bound becomes 1394+237=1631
2. If $X_{C,E}=1$,
we can go $C \rightarrow E$
So we can't go $E \rightarrow C$, so set it to M.
Now we leave row *C* and column *E*, so reduced matrix is

	A	*C*	*F*	*G*	*I*
A	M	296	0	0	207
B	0	309	M	30	219
E	283	M	295	297	7
G	0	381	8	M	219
H	0	125	18	12	35

Step-1: Find out the each row minimum element and subtract it from that row

	A	*C*	*F*	*G*	*I*	
A	M	296	0	0	207	(-0)
B	0	309	M	30	219	(-0)
E	276	M	288	290	0	(-7)
G	0	381	8	M	219	(-0)
H	0	125	18	12	35	(-0)

So, row minimum will be 7. (0+0+7+0+0=7)

Step-2: Find out the each column minimum element and subtract it from that column.

	A	*C*	*F*	*G*	*I*
A	M	171	0	0	207
B	0	184	M	30	219
E	276	M	288	290	0
G	0	256	8	M	219
H	0	0	18	12	35
	(-0)	(-125)	(-0)	(-0)	(-0)

So, column minimum will be 125. (0+125+0+0+0=125)
we get the lower bound = 1394+7+125=1526.
Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column).

	A	*C*	*F*	*G*	*I*
A	*M*	171	0(8)	0(12)	207
B	0(30)	184	*M*	30	219
E	276	*M*	288	290	0(311)
G	0(8)	256	8	*M*	219
H	0(0)	0(171)	18	12	35

Here maximum penalty is 311, occur at *XE,I* , so we choose *XE,I* to begin branch
There are two branches.
1. If *XE,I*=0, then we have an additional cost of 311 and the lower bound

becomes 1526+311=1837
2. If *XE*,*I*=1,
we can go *E*→*I*
Now we leave row *E* and column *I*, so reduced matrix is

	A	*C*	*F*	*G*
A	M	171	0	0
B	0	184	M	30
G	0	256	8	M
H	0	0	18	12

Here we have 0 in every row and column. So, the lower bound remains the same
i.e, 1526+0=1526
Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column).

	A	*C*	*F*	*G*
A	*M*	171	0(8)	0(12)
B	0(30)	184	*M*	30
G	0(8)	256	8	*M*
H	0(0)	0(171)	18	12

Here maximum penalty is 171, occur at *XH*,*C* , so we choose *XH*,*C* to begin branch
There are two branches.
1. If *XH*,*C*=0, then we have an additional cost of 171 and the lower bound becomes 1526+171=1697
2. If *XH*,*C*=1,
we can go *H*→*C*
Now we leave row *H* and column *C*, so reduced matrix is

	A	*F*	*G*
A	M	0	0
B	0	M	30
G	0	8	M

Here we have 0 in every row and column. So, the lower bound remains the same
i.e, 1526+0=1526

Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	A	*F*	*G*
A	*M*	0(8)	0(30)
B	0(30)	*M*	30
G	0(8)	8	*M*

Here maximum penalty is 30, occur at *XA*,*G* or *XB*,*A* , so we choose *XA*,*G* to begin branch
There are two branches.
1. If *XA*,*G*=0, then we have an additional cost of 30 and the lower bound becomes 1526+30=1556
2. If *XA*,*G*=1,

we can go *A*→*G*
So we can't go *G*→*A*, so set it to M.
Now we leave row *A* and column *G*, so reduced matrix is

	A	*F*
B	0	M
G	M	8

Step-1: Find out the each row minimum element and subtract it from that row

	A	*F*	
B	0	M	(-0)
G	M	0	(-8)

So, row minimum will be 8. (0+8=8)
we get the lower bound = 1526+8+0=1534

Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

A	*F*	
B	0(0)	*M*
G	*M*	0(0)

we can go $B \rightarrow A$and $G \rightarrow F$
So our final path is $F \rightarrow B \rightarrow A \rightarrow G \rightarrow F \rightarrow B \rightarrow A \rightarrow G \rightarrow F \rightarrow B$
and total distance is 108+370+16+282+108+370+16+282+108=1660

Our program is not able to solve this problem by penalty method, so we are trying to solve using diagonal completion method to solve this

Step-1: Find out the each row minimum element and subtract it from that row

	1	2	3	4	5	6	7	8	9	
1	M	354	296	108	444	258	0	96	207	(-16)
2	262	M	571	384	718	0	292	370	481	(-108)
3	171	538	M	54	36	449	256	96	0	(-141)
4	70	438	141	M	290	349	84	0	52	(-54)
5	283	649	0	167	M	553	297	208	7	(-177)
6	166	0	482	295	622	M	174	280	392	(-108)
7	0	384	381	122	458	266	M	108	219	(-16)
8	58	424	183	0	331	334	70	M	93	(-54)
9	117	483	35	0	78	394	129	41	M	(-106

Step-2: Find out the each column minimum element and subtract it from that column.

	1	2	3	4	5	6	7	8	9
1	M	354	296	108	408	258	0	96	207
2	262	M	571	384	682	0	292	370	481
3	171	538	M	54	0	449	256	96	0
4	70	438	141	M	254	349	84	0	52
5	283	649	0	167	M	553	297	208	7
6	166	0	482	295	586	M	174	280	392
7	0	384	381	122	422	266	M	108	219
8	58	424	183	0	295	334	70	M	93
9	117	483	35	0	42	394	129	41	M
	(-0)	(-0)	(-0)	(-0)	(-36)	(-0)	(-0)	(-0)	(-0

Step-3: Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	1	2	3	4	5	6	7	8	9
1	*M*	354	296	108	408	258	0(166)	96	207
2	262	*M*	571	384	682	0(520)	292	370	481
3	171	538	*M*	54	0(42)	449	256	96	0(7)
4	70	438	141	*M*	254	349	84	0(93)	52
5	283	649	0(42)	167	*M*	553	297	208	7
6	166	0(520)	482	295	586	*M*	174	280	392
7	0(166)	384	381	122	422	266	*M*	108	219
8	58	424	183	0(58)	295	334	70	*M*	93
9	117	483	35	0(35)	42	394	129	41	*M*

Step-4: List the penalties *P*(*i*,*j*) in descending order by value.
P(2,6)=520
P(6,2)=520
P(1,7)=166
P(7,1)=166
P(4,8)=93
P(8,4)=58
P(3,5)=42
P(5,3)=42
P(9,4)=35
P(3,9)=7
Step-5: The links (2,6),(1,7),(4,8),(3,5),(9,4) are selected for inclusion in the feasible partial tour.
Step-6: Feasible partial tour contains the following chains
2→6,1→7,9→4→8,3→5
(6,2),(7,1),(8,9),(5,3) can not be selected, because they are the closing links and create prohibited subtours.
Step-7: The new sub matrix is :

	2	1	9	3
6	M	274	500	590
7	400	M	235	397
8	478	112	M	237
5	826	460	184	M

Step-1: Find out the each row minimum element and subtract it from that row

	2	1	9	3	
6	M	0	226	316	(-274)
7	165	M	0	162	(-235)
8	366	0	M	125	(-112)
5	642	276	0	M	(-184)

Step-2: Find out the each column minimum element and subtract it from that column.

	2	1	9	3
6	M	0	226	191
7	0	M	0	37
8	201	0	M	0
5	477	276	0	M
	(-165)	(-0)	(-0)	(-125)

Step-3: Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	2	1	9	3
6	*M*	0(191)	226	191
7	0(201)	*M*	0(0)	37
8	201	0(0)	*M*	0(37)
5	477	276	0(276)	*M*

Step-4: List the penalties *P*(*i*,*j*) in descending order by value.
P(5,9)=276
P(7,2)=201
P(6,1)=191
P(8,3)=37
P(7,9)=0
P(8,1)=0

Step-5: The links (5,9),(7,2),(6,1),(8,3) are selected for inclusion in the feasible partial tour.

Step-6: Feasible partial tour contains the following chains
2→6→1→7→2,9→4→8→3→5→9
(2,2),(9,9) can not be selected, because they are the closing links and create prohibited subtours.
Step-7: The new sub matrix is :

	2	9
2	M	589
9	589	M

Repeat from step-1 to step-7
Step-1: Find out the each row minimum element and subtract it from that row

	2	9	
2	M	0	(-589)
9	0	M	(-589)

Step-3: Calculate the penalty of all 0's (penalty = minimum element of that row + minimum element of that column.)

	2	9
2	*M*	0(0)
9	0(0)	*M*

Step-4: List the penalties *P*(*i*,*j*) in descending order by value.
P(2,9)=0
P(9,2)=0
Step-5: The links (2,9) are selected for inclusion in the feasible partial tour.
Step-6: Feasible partial tour contains the following chains
2→6→1→7→2→9→4→8→3→5→9
So our final path is *B→F→A→G→B→I→D→H→C→E→I→B*

and total distance is **108+274+16+400+589+106+54+237+177+184+589=2734**.

5.1 Result

Feasible partial tour contains the following chains.
So our final path is ***B→F→A→G→B→I→D→H→C→E→I→B***
(i.e) Tirupathi→ Kanchipuram→ Srirangam→ Thiruvellarai→ Tirupathi→ Srivilliputhur→ Azhagarkovil (Madurai)→ Thirukoshtiyur→ Sri Vaikuntam→ Thiruvananthapuram (Kerala)→ Srivilliputhur→ Tirupathi.
and total distance is 108+274+16+400+589+106+54+237+177+184+589=2734.

6. CONCLUSION

The Travelling salesman problem can be solved using a variety of ways. In this work, the Diagonal Completion technique is used to discover the best path. Each step calculates the minimum distance (cost matrix). In the tests undertaken, the combined diagonal completion and 2-optimal approach revealed that it is a relatively fast and highly effective Branch and Bound method for delivering good solutions to travelling salesman problems, even those of enormous size.

REFERENCES

[1] GerardReinelt. The Traveling Salesman: Computational Solutions for TSP Applications. Springer-Verlag, 1994.

[2] D. B. Fogel, "An Evolutionary Approach to the Traveling Salesman Problem", Biol. Cybern. 60,139- 144 (1988) .

[3] Kylie Bryant ,Arthur Benjamin, Advisor, "Genetic Algorithms and the Traveling Salesman Problem", Department of Mathematics, December 2000.

AUTHOR PROFILE:

Author – 1

Dr.S.Anuradha, Professor, Head Department of Mathematics, Data Science and Analytics is having 29 Years Teaching and Research Experience. She received Best Teacher Award 2013 in Akshaya Institute of Management studies Coimbatore , Grade A award for Team Leader in Task Based Training (TBT-2019) conducted by e-Yantra Lab Setup Initiative(eLSI), IIT BOMBAY, Vocational Service Award 2021 from Rotary Club of Coimbatore DownTown in 2021, Most Accomplished Spoken Tutorial Educator Resource Award in 2023 by SPOKEN TUTORIAL, IIT BOMBAY. She is a Teacher invitee in 34th Mid Year Meeting of Indian Academy of Sciences in the year 2023. She published 73 International Journal, 12 Books, 9 Book Chapter and 1 patent. She is Editorial Board Member in International Research Conference Series-Scientific and Technical Committee and Editorial board Member on mathematical and Computational Sciences and Reviewer in International Journals of American Journal of Applied Scientific Research, Science Publishing Group, Advances in Applied Research, www.IndianJournals.com,Latin American Applied Research, Applied Mathematics & Information Sciences, Nature Publishing Group. 7 PhD Scholars and 18 MPhil Scholars awarded Degree under her guidance.

Hindusthan International Journal of Multidisciplinary Research (HIJOMR)
Volume 1 Issue 1, September 2024, pp.47-67

Author –2

Mrs. S Renuga Devi , designated as Assistant Professor, Department of Mathematics, Hindusthan College of Arts & Science, Coimbatore. She has 15 years of teaching experience. She has Published 5 books and 1 Paper in International Journal and 2 Papers in Conference Proceedings. Her field of interest is Graph Theory.

.

Author – 3

Dr.D.Mallika, designated as Professor, Department of Mathematics, Hindusthan College of Arts & Science, Coimbatore. She has 17 years of teaching experience. She has Published 6 Books, 6 Papers under International, Scopus, SCI and Web of Science Journals and 2 Papers in Conference Proceedings. Her Field of field of Interest is Differential equations.

Author – 4

Mrs.P.Rekha, designated as Assistant Professor, Department of Mathematics, Hindusthan College of Arts & Science, Coimbatore. She has 12 years of teaching experience. She has Published 6 books and 2 Papers in Conference Proceedings. Her field of interest is Operator Theory.