Vol. XVII & Issue No. 04 April - 2024
INDUSTRIAL ENGINEERING JOURNAL

TOURIST TRAVELLING PROBLEM SOLVER FOR BERHAMPUR CITY USING TRAVELLING SALESMAN PROBLEM

Chittaranjan Mallick
Department of Basic Science (Mathematics), Parala Maharaja Engineering College (Govt.), Berhampur 761003, Odisha, India
Email: chittaranjan.bs@pmec.ac.in

Sourav Kumar Bhoi
Department of Computer Science and Engineering, Parala Maharaja Engineering College (Govt.), Berhampur 761003, Odisha, India
Email sourav.cse@pmec.ac.in

Chitta Ranjan Mohanty
Department of Civil Engineering, Parala Maharaja Engineering College (Govt.), Berhampur 761003, Odisha, India
Email chitta123@yahoo.com

Abstract:

In this work, we applied the Travelling Salesman Problem Technique is to optimize the shortest route for visiting different tourist places in Berhampur city of Odisha in India with minimum transportation cost. The tourist knows the distance (cost) between every pair of tourist places of a city. The problem is to select a possible route that begins from the starting point (tourist place) of a city and passes through each place only once and return to the starting point by using the shortest possible distance. Suppose n tourist places are there in a city then (n-1)! routes are possible. At present, the best option to solve such types of problems is by using assignment techniques. Here, we have to focus optimum route (path) where the total distance or cost is minimum. In this problem, the tourist started from Gopalpur Beach and ended at Rambha Chilika point by passing through all tourist places once. Therefore, the tourist uses different techniques of Travelling Salesman Problem (TSP) solving methods to solve this problem. The methods used are Branch & Bound (Penalty) Method, Diagonal Completion Method and Nearest Neighbor method, to find the initial feasible solution (IFS). From this case study, it is found that the minimum transportation cost achieved is 166 Km for Branch & Bound (Penalty) Method as compared to other methods.

Keywords: Tourist Service, Optimum route, Assignment problem, Branch & Bound Method, Diagonal Completion Method, Nearest Neighbor Method.

1. INTRODUCTION

Odisha in India is known for its famous and beautiful tourist places. So, one such city is Berhampur in Odisha which has huge attractions for tourists (Brahmapur, (2021)). The city is known as the silk city in Odisha because of the famous Pata Sarees. It is nearly 165 Km from Bhubaneswar, the capital city of Odisha. It is well known for its long coastal line which attracts tourists much. So, many such beaches are there such as Gopalpur, Arjiapalli, and Sonepur beach. It has also many beautiful lakes such as Tampara lake and Chilika lake. It has also famous temples such as Tara Tarini and Siddha Bhairavi. It is also famous for tourist parks such as Ramalingeswar park and Biju Patnaik Park. Further, it has many fort attractions, one such fort is Potagarh fort. As per our knowledge, these are some tourist places mentioned in our problem as a case study. However, many such other places are also there which has much more tourist attractions. Lastly, it's may need to say Berhampur is the best among all. These are the main reasons tourists may attract to the city to travel all the places of Berhampur with a minimum travelling cost. If the tourist is to visit only two places (P&Q), so the number of possible routes is (P→Q→P), and there is only one choice i.e. (2-1)! If the number of places is 3 (P, Q, and R) of which its starting point is P, there are two possible routes i.e.(3-1)! (P→Q→R and P→R→Q). Similarly, if there are four places, the number of possible routes is 6 i.e. (4-1)! (P→Q→R→S, P→Q→S→R, P→R→Q→S, P→R→Q→S, P→R→S→Q, P→S→Q→R and P→S→R→Q). If this process continues in this manner the tourist will have a total of (n-1)! possibly round trips starting from a given nth place. Suppose a tourist has to visit all the places, the initial feasible solution remains different for the selection of the starting point. Here, we have to focus to find the best route without trying each one, but unfortunately, there is no analytical method, which can give the best result. However, a few computational methods are suggested to solve this type of problem.

The major areas of research done in this paper are mentioned as follows:

1. In this paper, we have taken a case study of how a tourist visits the number of Tourist Places in Berhampur city with a minimum cost of transportation using different travelling salesman problem-solving methods.
2. In this problem, the tourist started from Gopalpur Beach and ended at Rambha Chilika point by passing through all tourist places once only. The methods used to solve this problem are Branch & Bound (Penalty) Method, Diagonal Completion Method and Nearest Neighbor method, to find an IFS.
3. To solve the above problem, 10 tourist places of Berhampur city are considered. From this case study, it is found that the minimum transportation cost achieved is 166 Km for Branch & Bound (Penalty) Method as compared to other methods.

The paper is managed as follows. In Section 2, Travelling Salesman related works are presented. In Section 3, the methodology is presented. Section 4 presents the Results and Discussion. In Section 5, we concluded the work.

2. RELATED WORKS

Travelling Salesman Problem is essentially a vehicle routing problem. In such a problem salesman wants to determine the route (tour) that yields minimum distance travelled subject to the constraints that of the n given cities, each city is visited once and this way the tourist return to the city he started from. Travelling salesman problem is a special kind of Assignment problem. Basically, the tourist service problem aims at maximizing profit by minimizing cost. The author in a work describes the TSP problem in ex-urban mass transit (Gaffi et al. (1999)). In another work, the authors solved the problem by linear programming by using TSP with more than several hundred cities; over the years (Applegate et al. (2011)). In another work, the authors studied the application of Genetic Algorithms, Neural networks etc. with the application of evolutionary algorithms to solve the TSP problem (Davendra (2010)). In another work, the authors formulate a method for employees for doing high-quality work for high job satisfaction who are desirous of higher-order need satisfaction tends to have high motivation (E. Lawler et al. (2011)). In another work, the authors studied the results indicating that the power of a particular hierarchical clustering procedure is a function of the type of partitions (Hubert L. J. et al. (1978)). In another work, the authors describe the public transport crew costs using TSP (Patrikalakis et al. (1992)). In another work, the authors describe the use of route planning methods including TSP (Sharma (2016)). In another work, the authors classify the satisfaction of the branching and the calculation of the lower bounds in solving assignment problems (J. Little et al. (2016)). In another work, the authors studied optimally designed routes through integrating vehicle scheduling problems sequentially (Jelokhani-Niaraki et al. (2021)). In another work, the authors proposed an algorithm for vehicle and crew scheduling (Friberg et al. (1999)). In another work, the authors describe the basic methodology of arc routing by discussing generic arc routing models and their solution techniques are discussed (L. Golden et al. (1986)). In another work, the authors studied the vehicle routing and scheduling problem (M. M. Solomon (1987)). In another work, the authors studied dynamic programming (M. Hayes et al. (1984)). In another work, the authors studied new heuristics in TSP (Nuriyeva, F et al. (2012)). In another work, the authors studied vehicle capacity and travel time of a city such as travelled limitations and a penalty for delivering passengers (Ntakolia et al. (2021)). In another work, the authors solve the technique using point to point new direction in the field of man-machine interaction and the field of Artificial Intelligence (P. Krolak et al. (1971)). R. In another work, the authors considered a problem of the vehicle and crew scheduling as compared within a sequential manner (Freling et al. (1994 & 2001)).In another work, the authors describe for every TSP problem a short description is given along with known lower and upper bounds (Reinelt (1994)). In another work, the authors make a decision problem in management science (Srinivas, B. et al. (2015)). In another work, the authors provides a branch-and-bound for getting optimum solutions to all travelling-salesman problems, ranging in size up to sixty-four cities (Lin and Kernighan (1973)). In another work, the authors developed a flexible ridesharing system to minimize travel costs while visiting some paths to satisfy a pre-established quota (Silva et al. (2020)). In another work, the authors applied the timetableing problem to an Educational Institution adopted in a tertiary institution (Mallick et al. (2021)). In another work, the authors derive a generalized model for crew members to solve the crew assignment problems, which determines coverage of all buses at a minimal cost (Mallick et al. (2021)). In a work, the authors maximizes the net profit for TSP using metaheristic algorithms (Isik et al. (2023)). In another work the authors optimizes the TSP for tour package routes in Langkat using cheapest insertion heuristic algorithm (Syahputra et al. (2023)). These are some of the works in TSP for optimization of routes and maximizing the profit.

3. METHODOLOGY

We use different methods to solve Tourist Travelling Problem using different Travelling Salesman Problem-solving methods to find the minimum distance from a given starting point are as follows (Devendra (2010), Lawer et al. (2011), Hubert at al. (1978), Patrikalakis et al. (1992), Sharma et al. (2016), & Little et al. (2016)):

1. Diagonal Completion Method
2. Branch and Bound (Penalty method)
3. Nearest Neighbor Method

3.1 Mathematical Form of Travelling Salesman Problem

If C_{ij} is the cost of moving from place i to j and X_{ij} be the minimum travelling time i.e. X_{ij}=1, if the tourist goes directly from place i to place j, and 0 otherwise, then the problem is to find X_{ij} which minimizes.

X_{ij} which minimizes.

$$Z=\sum_{i=1}^{n} \sum_{j=1}^{n} CijXij \quad (1)$$

Subject to $\sum_{j=1}^{n} Xij = 1$ (2)

$$\sum_{i=1}^{n} Xij = 1 \quad (3)$$

and X_{ij}=0 or 1 where i=1,2,……..,n; j=1,2,-----------n

With two additional constraints as shown in eq. (2) and eq.

(3), that no places will be visited again before the tour of all places. It is necessary to complete the route by reaching a place once, and not possible to move to the same places again, which means $C_{ii}=\infty$. Thus the travelling salesman problem or the multiple product production problems can be put in the form of an assignment problem, which can be shown below in Table 1.

Table 1: Mathematical formulation of Travelling Salesman problem.

∞	C_{12}	C_{13}	.	.	C_{1n}
C_{21}	∞	C_{12}	.	.	C_{12}
C_{12}	C_{22}	∞	.	.	C_{2n}
.	.	.	.	.	.
.	.	.	.	.	.
Cn_1	Cn_2	C_{n3}	.	.	∞

Given the above, the assignment problem can be solved and one may hope that the solution satisfies the additional restriction also.

3.2 Diagonal Completion Method

To find IFS for tourist problems using the diagonal completion method is an integral part of the travelling salesman problem. By using this method by applying matrix reduction getting zero elements by assigning zeros that fall on the diagonal of a reduced matrix; and a link selected by choosing the starting point of the initial tour. The method is presented below:

Step 1. Consider the cost matrix, given in Table 2. For getting at least one zero in each row and column, by using reduction of a matrix

Step 2. Subtract the minimum element in every row for row reduction from each element in a row and repeat the same procedure for each row (i=l,..., 10), and similarly in column j (j = 1, . . . , 10). By using the Hungarian method of assignment problem for getting matrix reduction, that is the first level of the reduced matrix as per diagonal completion algorithm.

Step 3. Calculate the penalty of all 0's from the above-reduced matrix, after row& column reduction by addition of smallest element in each row & column of corresponding 0's. For the fulfilment of minimum criteria, the smallest element is chosen in rows and columns.

Step 4. A single link or links can take care of a partial tour.

Step 5. The largest penalty is calculated accordingly in order so that the largest value in the link is considered as the starting link of the modified tour. After selecting starting link, every other link is taken in such a way that considers all M elements to be present on the diagonal. Ignore the cost elements in the matrix that are not on the diagonal and the submatrix is calculated like this.

Step 6. Every elements on the diagonal become zero by using starting elements then the initial feasible tour is achieved. One none zero elements is only present in diagonal.

3.3.1 Branch and Bound (penalty) Method

The Branch & Bound Technique is a recently developed technique to deal with combinatorial problems. This technique involves a systematic search of all feasible solutions. In this method, by using assignment problem-solving techniques, it becomes an additional restriction of choosing starting point from a particular place, visiting each place once then coming back to the starting point. Because of this fact, the travelling salesman problem so obtained provides a lower bound. If at least one sub tour (if the tourist visits a certain place and returns to that place later) exists in the solution then we have to adopt the procedure stated in the below steps.

If the solution to the travelling salesman problem have no constraints (i.e. there happens to sub tour(s) in the solution) then select a sub tour and let k be the number of arcs (links) in the selected sub tour W.L. Eastman selects the sub tour with the smallest number of arcs]. Then make a branch into k-subproblems. For instance, if the subtour is P→Q→R→P then subproblem-1, let D*(P→Q)=∞, for subprobiem-2, D*(Q→R)=∞ and for subprobiem-3, D*(C→A)=∞ and so on.

3.4 Nearest Neighbour Method

The nearest neighbour (NN) method for solving a travelling salesman tour is as follows:

The algorithm generates the optimal path to visit all the routes exactly once and return to the starting route. The procedure is the same for all the routes.

Step-1: Calculate a route randomly as starting route in which a tourist wants to visit.

Step-2: Determine the nearest route that connects the current route and mark the route which is not visited.

Step-3: Set the new route as the current route.

Step-4: Find out the previous current route which is already been visited.

Step-5: If once all the routes are visited, then stop.

Step-6: Go to step-2.

RESULTS AND DISCUSSION

As already discussed earlier, the assignment problem relates to concerning the distribution of various tourist places in Berhampur, a tourist wants to visit each place only once with minimal cost. To study the effectiveness of this cost-minimizing problem we have solved this using a case study by taking 10 tourist places. The problem is discussed below and the results are computed with the Atozmath calculator (Atozmath, (2021)).

Problem: A tourist wants to visit 10 different tourist places in Berhampur such as *Gopalpur Beach, Tara Tarini Temple, Tampara Lake, Bhairavi Temple, Biju Patnaik Park, Ramalingeswar Park, Aryapalli Beach, Potagarh Fort, Sonepur Beach, and Rambha Chilika* and we denote these tourist places in this problem as *A, B, C, D, E, F, G, H, I, and J* respectively. The distance in Kms between the 10 places are taken from Google map (Google Map, (2021)) and the distance matrix is represented in Table 2. If the tourist starts from place A and returns to the same place A, then the tourist should select which route so that the total distance travelled is minimum.

Table 2: Distance Matrix of Tourist place in Berhampur (in Km).

	Gopalpur Beach	Tara Tarini Temple	Tampara Lake	Bhairavi Temple	Biju Patnaik Park	Ramalingeswar Park	Aryapalli Beach	Potagarh Fort	Sonepur Beach	Rambha Chilika
Gopalpur Beach	∞	38	26	29	17	18	14	26	41	40
Tara Tarini Temple	38	∞	37	48	31	32	44	42	59	36
Tampara Lake	26	37	∞	37	25	26	9	11	49	25
Bhairavi Temple	29	48	37	∞	18	19	41	45	11	59
Biju Patnaik Park	18	31	25	18	∞	1	25	33	30	47
Ramalingeswar Park	17	31	25	18	1	∞	26	34	31	48
Aryapalli Beach	14	44	9	41	30	26	∞	17	50	31
Potagarh Fort	26	42	11	45	33	34	17	∞	58	16
Sonepur Beach	41	59	49	11	30	31	50	58	∞	71
Rambha Chilika	40	36	25	59	47	48	31	16	71	∞

4.1 Solution using Diagonal Completion Method

In the given problem (Table 2) represents a balanced assignment problem.is shown in Table 3.

Table 3: Modifying Tables.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	Row minimum
A	M	38	26	29	17	18	14	26	41	40	(14)
B	38	M	37	48	31	32	44	42	59	36	(31)
C	26	37	M	37	25	26	9	11	49	25	(9)
D	29	48	37	M	18	19	41	45	11	59	(11)
E	18	31	25	18	M	1	25	33	30	47	(1)
F	17	31	25	18	1	M	26	34	31	48	(1)
G	14	44	9	41	30	26	M	17	50	31	(9)
H	26	42	11	45	33	34	17	M	58	16	(11)
I	41	59	49	11	30	31	50	58	M	17	(11)
J	40	36	25	59	47	48	31	16	71	M	(16)

Step1: In Table 3 row minimum is calculated (Subtract the minimum element in each row from all the elements of that row).

Table 4: Calculating Row Minimum.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	M	24	12	15	3	4	0	12	27	26
B	7	M	6	17	0	1	13	11	28	5
C	17	28	M	28	16	17	0	2	40	16
D	18	37	26	M	7	8	30	34	0	48
E	17	30	24	17	M	0	24	32	29	46
F	16	30	24	17	0	M	25	33	30	47
G	5	35	0	32	21	17	M	8	41	22
H	15	31	0	34	22	23	6	M	47	5
I	30	48	38	0	19	20	39	47	M	6
J	24	20	9	43	31	32	15	0	55	M
Column minimum	5	20	0	0	0	0	0	0	0	5

Step-2: In Table 4, column minimum is calculated (Subtract the minimum element in each column from all the elements of that column).

Table 5: Calculating Column Minimum.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	M	4	12	15	3	4	0	12	27	21
B	2	M	6	17	0	1	13	11	28	0
C	12	8	M	28	16	17	0	2	40	11
D	13	17	26	M	7	8	30	34	0	43
E	12	10	24	17	M	0	24	32	29	41
F	11	10	24	17	0	M	25	33	30	42
G	0	15	0	32	21	17	M	8	41	17
H	10	11	0	34	22	23	6	M	47	0
I	25	28	38	0	19	20	39	47	M	1
J	19	0	9	43	31	32	15	0	55	M

Step-3: Calculate the penalty of all 0's from the above-reduced matrix, after row& column reduction by addition of smallest element in each row & column of corresponding 0's. For example, in row-1 the smallest elements of row and column of 0 are 3 and 0 respectively. So we have 3+0=3, in row-4 by calculating penalty by choosing minimum element in that row and column are 7 and 27 respectively i.e. the penalty, in that case, is (27+7=34) and all other penalties of all 0's are calculated similarly as shown in Table 6.

Table 6: Penalty of all 0's.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	M	4	12	15	3	4	0(3)	12	27	21
B	2	M	6	17	0(0)	1	13	11	28	0(0)
C	12	8	M	28	16	17	0(2)	2	40	11
D	13	17	26	M	7	8	30	34	0(34)	43
E	12	10	24	17	M	0(11)	24	32	29	41
F	11	10	24	17	0(10)	M	25	33	30	42
G	0(2)	15	0(0)	32	21	17	M	8	41	17
H	10	11	0(0)	34	22	23	6	M	47	0(0)
I	25	28	38	0(16)	19	20	39	47	M	1
J	19	0(4)	9	43	31	32	15	0(2)	55	M

Step-4: List the penalties of all cell values P(i,j) in descending order by value. *P*(D,I)=34,*P*(I,D)=16,*P*(E,F)=11,*P*(F,E)=10,*P*(J,B)=4,*P*(A,G) =3,*P*(C,G)=2,*P*(G,A)=2,*P*(J,H)=2,*P*(B,E)=0,*P*(B,J)=0,*P*(G,C) =0,*P*(H,C)=0,*P*(H,J)=0

Step-5: The largest penalty is calculated accordingly in order so that the largest value in the link is considered as the starting link of the modified tour [i.e. link (D, I) in the problem]. Once the starting link is chosen, every other link is considered for inclusion in the partial tour. Thus, link (D, I) is to be deducted because it would create a sub tour: D-I. The links (D,I),(E,F),(J,B),(A,G),(B,E),(G,C),(H,J) are selected for the partial tour. Step-6: Feasible partial tour contains the following chains, D→I, H→J→B→E→F, A→G→C (I,D),(F,E),(C,G) ,(G,A),(J,H),(B,J),(H,C) cannot be selected.

Table 7: The new submatrix of the partial tour (Row minimum).

	D	H	A	Row minimum
I	M	58	41	41
F	18	M	17	17
C	37	11	M	11

Step-7: In Table 7, find the minimum element in every row and then subtract that element from every row. Hence, the first reduced cost matrix is generated where each row has exactly one zero. Repeat from step-1 to step-7.

Table 8: Calculating Column Minimum.

	D	H	A
I	M	17	0
F	1	M	0
C	26	0	M
Column minimum	1	0	0

Step-1: Find the smallest element in every column and then subtract that element from every column. Hence, the first reduced cost matrix is generated where each column has exactly one zero. After Column Reduction Calculation of penalty of all 0's

Step-2: Calculate the penalty of all 0's from the above-reduced matrix, after row& column reduction by addition of smallest element in each row & column of corresponding 0's.

Step-3: List the penalties *for all cells P(i,j)* in descending order by value. *P*(C,H)=42, *P*(F,D)=25, *P*(I,A)=17, *P*(F,A)=0

Step-4: The links (C,H),(F,D),(I,A) are chosen for entry into feasible partial tour. (F, A) cannot be selected, because arising of prohibited sub tours due to the closing links. Step-5: Feasible Partial tour contains the following chains.

D → I → A → G → C → H → J → B → E → F → D .

So our final path is D→I→A→G→C→H→J→B→E→F→D, and total distance travelled is 11 + 41 + 14 + 9 + 11 + 16 + 36 + 31 + 1 + 18 = 188 kms.

Figure 1. Distance of 188 Km traversed by the tourist through different tourist points of Berhampur city using the diagonal method.

4.2 Solution of Branch and bound (penalty) method

In the given problem (Table 2) is a balanced assignment problem. The number of rows=10 and Columns=10. In Table-3. So, the row minimum will be 114 (14+31+9+11+1+1+9+11+11+16=114) i.e. Sum of row minimum gives us a lower bound. Step-1: Find the smallest element in every row and then subtract that element from each row. Hence, the first reduced cost matrix is generated where each row has exactly one zero (Table-3).

Step-2: In Table-4, Calculating Column Minimum So, column minimum will be 30 (5 + 20 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 5 = 30).

We get the lower bound=114+30=144. After that, the column minimum is calculated (Subtract the smallest element in each column from all the elements of that column). Then Column Reduction is done and presented in Table 9.

Table 9: After Column Reduction.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	M	4	12	15	3	4	0	12	27	21
B	2	M	6	17	0	1	13	11	28	0
C	12	8	M	28	16	17	0	2	40	11
D	13	17	26	M	7	8	30	34	0	43
E	12	10	24	17	M	0	24	32	29	41
F	11	10	24	17	0	M	25	33	30	42
G	0	15	0	32	21	17	M	8	41	17
H	10	11	0	34	22	23	6	M	47	0
I	25	28	38	0	19	20	39	47	M	1
J	19	0	9	43	31	32	15	0	55	M

Table 10: Penalty of all 0's.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	M	4	12	15	3	4	0(3)	12	27	21
B	2	M	6	17	0(0)	1	13	11	28	0(0)
C	12	8	M	28	16	17	0(2)	2	40	11
D	13	17	26	M	7	8	30	34	0(34)	43
E	12	10	24	17	M	0(11)	24	32	29	41
F	11	10	24	17	0(10)	M	25	33	30	42
G	0(2)	15	0(0)	32	21	17	M	8	41	17
H	10	11	0(0)	34	22	23	6	M	47	0(0)
I	25	28	38	0(16)	19	20	39	47	M	1
J	19	0(4)	9	43	31	32	15	0(2)	55	M

Step- 3: Calculate the penalty of all 0's from the above-reduced matrix, after row& column reduction by addition of smallest element in each row & column of corresponding 0's. For example, in row-1 the smallest elements of row and column of 0 are 3 and 0 respectively. so we have 3+0=3, In row-4 by calculating penalty by choosing minimum element in that row and column are 7 and 27 respectively i.e. the penalty, in that case, is (27+7=34) and similarly, all 0 penalties can be calculated. Here the maximum penalty is 34, which occurs at Row -D and column-I (i.e. D, I) to begin branch, there are two ways for selecting branches.

1. If D, I=0, then we have chosen the highest penalty 34 that can be added in the lower bound and it becomes 144+34=178.

2. If D, I=1, then move from D→I.

So we cannot move now from I→D, so set it to M. Now we eliminate row D and column I, so the reduced matrix is shown in Table-11.

Table- 11: Calculating Row Minimum.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*J*	Row Minimum
A	M	4	12	15	3	4	0	12	21	0
B	2	M	6	17	0	1	13	11	0	0
C	12	8	M	28	16	17	0	2	11	0
E	12	10	24	17	M	0	24	32	41	0
F	11	10	24	17	0	M	25	33	42	0
G	0	15	0	32	21	17	M	8	17	0
H	10	11	0	34	22	23	6	M	0	0
I	25	28	38	M	19	20	39	47	1	1
J	19	0	9	43	31	32	15	0	M	0

So, the row minimum will be 1 (0+0+0+0+0+0+0+1+0=1).

Step-4: Again, find the minimum element in every row and then subtract that element from each row. Hence, the first reduced cost matrix is generated where each row has exactly one zero from Table-11.

Table 12: Calculating Column Minimum.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*J*
A	M	4	12	15	3	4	0	12	21
B	2	M	6	17	0	1	13	11	0
C	12	8	M	28	16	17	0	2	11
E	12	10	24	17	M	0	24	32	41
F	11	10	24	17	0	M	25	33	42
G	0	15	0	32	21	17	M	8	17
H	10	11	0	34	22	23	6	M	0
I	24	27	37	M	18	19	38	46	0
J	19	0	9	43	31	32	15	0	M
Column Minimum	0	0	0	15	0	0	0	0	0

So, column minimum will be 15 (0+0+0+15+0+0+0+0+0=15) and we get lower bound=144+1+15=160.

Step-5: In Table-12 column minimum calculated (Subtract the minimum element in each column from all the elements of that column).

Table 13: After Column Reduction.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*J*
A	M	4	12	0	3	4	0	12	21
B	2	M	6	2	0	1	13	11	0
C	12	8	M	13	16	17	0	2	11
E	12	10	24	2	M	0	24	32	41
F	11	10	24	2	0	M	25	33	42
G	0	15	0	17	21	17	M	8	17
H	10	11	0	19	22	23	6	M	0
I	24	27	37	M	18	19	38	46	0
J	19	0	9	28	31	32	15	0	M

Table-14: Penalty of all 0's.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*J*
A	*M*	4	12	(2)0	3	4	(0)0	12	21
B	2	*M*	6	2	(0)0	1	13	11	(0)0
C	12	8	*M*	13	16	17	(2)0	2	11
E	12	10	24	2	*M*	(3)0	24	32	41
F	11	10	24	2	(2)0	*M*	25	33	42
G	(2)0	15	(0)0	17	21	17	*M*	8	17
H	10	11	(0)0	19	22	23	6	*M*	(0)0
I	24	27	37	*M*	18	19	38	46	(18)0
J	19	(4)0	9	28	31	32	15	(2)0	*M*

Here the maximum penalty is 18, which occurs at Row -I and column-J (i.e. I, J) to begin branch, there are two ways for selecting branches.

1. If I, J=0, then we have chosen the highest penalty 18 that can be added in the lower bound and it becomes 160+18=178.

2. If I, J=1, then we can move from I→J.

Here, till now we traversed D→I →J, so we cannot move from J→D, so set it to M. Now we eliminate row-I and column-J, and the reduced matrix is shown in Table-15.

Table 15: After deletion of row-I and column- J.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
A	M	4	12	0	3	4	0	12
B	2	M	6	2	0	1	13	11
C	12	8	M	13	16	17	0	2
E	12	10	24	2	M	0	24	32
F	11	10	24	2	0	M	25	33
G	0	15	0	17	21	17	M	8
H	10	11	0	19	22	23	6	M
J	19	0	9	M	31	32	15	0

From Table-15, zero (0) element occurs in every row and column., the lower bound remains the same i.e.160+0=160.

Table 16: Penalty of all 0's.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
A	M	4	12	0(2)	3	4	0(0)	12
B	2	M	6	2	0(1)	1	13	11
C	12	8	M	13	16	17	0(2)	2
E	12	10	24	2	M	0(3)	24	32
F	11	10	24	2	0(2)	M	25	33
G	0(2)	15	0(0)	17	21	17	M	8
H	10	11	0(6)	19	22	23	6	M
J	19	0(4)	9	M	31	32	15	0(2)

Here the maximum penalty is 6, which occurs at Row-H and column-C (i.e. H, C) to begin branch, there are two ways for selecting branches.

1. If H, C=0, then we have chosen the highest penalty as 6 that can be added in the lower bound and it becomes 160+6=166.

2. If H, C=1, then can move H→C, but we cannot move from C→H, so set it to M.

Similarly Proceeding in this manner, we get

Step-6: After deletion of row-H and column-C, every row and column occurs with a zero (0) element, and the lower bound is the same i.e.160+0=160. Here the maximum penalty is 10, which occurs at Row-G and Column-A (i.e. G, A) to begin the branch. There are two ways for selecting the branches:

1. If G, A=0, then we have chosen the highest penalty as 10 that can be added in the lower bound and it becomes 160+10=170

2. If G, A=1, then we can move from G→A, but cannot move from A→G, so set it to M.

Step-7: After deletion of row-G and column-A, zero (0) element occurs in every row and column. The lower bound remains the same i.e.160+0=160. Here the maximum penalty is 21, which occurs at Row-C and Column-G (i.e. C, G) to begin the branch. There are two ways for selecting branches:

1. If C, G=0, then we have chosen the highest penalty as 21 that can be added in the lower bound and it becomes 160+21=181.

2. If C, G=1, then we can move from C→G, but cannot move from G→C, so set it to M.

Step-8: After deletion of row -C and column -G, zero (0) elements occur in every row and column., the lower bound remains the same i.e.160+0=160. Here the maximum penalty is 11, which occurs at Row-J and Column-H (i.e. J, H) to begin the branch. There are two ways for selecting branches

1. If J, H=0, then we have chosen the highest penalty as 6 that can be added in the lower bound and it becomes 160+6=166.

2. If J, H=1, then we can move from J→H.

Here till now we traversed as *D→I→J→H→C→G→A,* So, we cannot move from A→D, so set it to M. Now, eliminate row-J and column-H.

Step-9: After deletion of row-J and column-H we calculate the Row Minimum. So, row minimum will be 3 (3 + 0 + 0 + 0 = 3). Again, find the minimum element in each row and then subtract that element from each row. Hence, the first reduced cost matrix is generated where each row has exactly one zero. Then we calculated the Column Minimum.

Step-10: After Column Reduction, 3 be the required column minimum i,e. (1+2+0+0=3). Then, we get the lower bound =160 + 3 + 3 = 166.

By calculating the penalty of all 0's, the maximum penalty is 9, which occurs at Row-A and Column-B (i.e. A, B) to begin the branch. There are two ways for selecting the branches:

1. If A, B=0, then we have chosen the highest penalty as 9 that can be added in the lower bound and it becomes 166+9=175.

2. If A, B=1, then we can move from A→B.

Here till now we traversed as *D→I→J→H→C→G→A→B.* So, we cannot move from B→D, so set it to M.

Step-11: After deletion of row-A and column-B, we have every row and column with zero (0) elements, and the lower bound becomes the same i.e.160+0=160.

By calculation of penalty of all 0's, the maximum penalty is 1, which occurs at Row-B and Column-E (i.e. B, E) or Row-E and Column-F, so we choose row-E and column-F to the beginning branch. There are two ways for selecting branches:

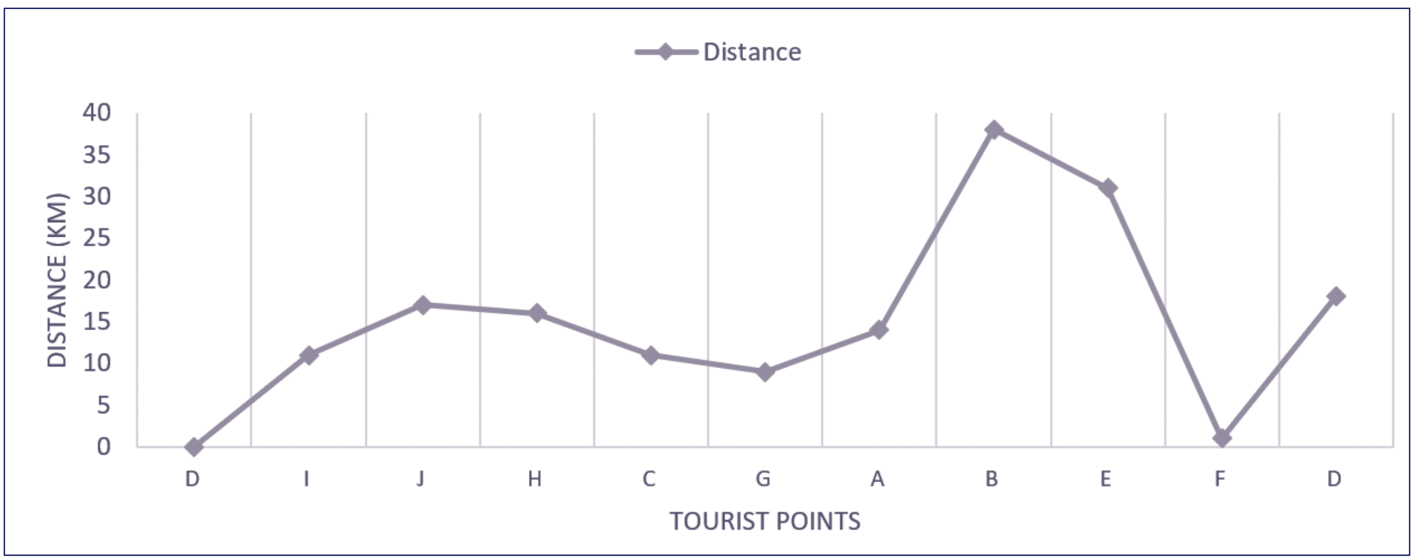
1. If E, F=0, then we have chosen the highest penalty as 1 that can be added in the lower bound and it becomes 166+1=167.

2. If E, F=1, then we can move from E→F, but not from E→F, so set it to M.

Step-12: After deletion of row-E and column-F, we have every row and column with zero (0) elements, the lower bound same i.e.160+0=160.

For calculation of Penalty of all 0's, we can go from B→E and F→D. So the final path is: D → I → J → H → C → G → A → B →E → F → D and the total distance is: 11+17+16+11+9+14+38+31+1+18=166 Kms.

Figure 2. Distance of 166 Km traversed by the tourist through different tourist points of Berhampur city using branch and bound (penalty) method.



4.3 Solution using the nearest neighbor method

As per Table 2, in this section, we have given a solution for the nearest neighbor approach. The steps solved are shown below:

1. If we start from A, then the path is
A→G=14,*G→C*=9,*C→H*=11,*H→J*=16,*J→B*=36,*B→E*=31,*E→F*=1,*F→D*=18,*D→I*=11,*I→A*=41
And total distance=188

2. If we start from B, then the path is
B→E=31,*E→F*=1,*F→A*=17,*A→G*=14,*G→C*=9,*C→H*=11,*H→J*=16,*J→D*=59,*D→I*=11,*I→B*=59
And total distance=288

3. If we start from C, then the path is
C→G=9,*G→A*=14,*A→E*=17,*E→F*=1,*F→D*=18,*D→I*=11,*I→J*=17,*J→H*=16,*H→B*=42,*B→C*=37
And total distance=182

If we start from D, then the path is:
D→I=11,*I→J*=17,*J→H*=16,*H→C*=11,*C→G*=9,*G→A*=14,*A→E*=17,*E→F*=1,*F→B*=31,*B→D*=48
and total distance = 175

5. If we start from E then the path is
E→F=1,*F→A*=17,*A→G*=14,*G→C*=9,*C→H*=11,*H→J*=16,*J→*

B=36,$B\rightarrow D$=48,$D\rightarrow I$=11,$I\rightarrow E$=30
And total distance=193

6. If we start from F then the path is
$F\rightarrow E$=1,$E\rightarrow A$=18,$A\rightarrow G$=14,$G\rightarrow C$=9,$C\rightarrow H$=11,$H\rightarrow J$=16,$J\rightarrow B$=36,$B\rightarrow D$=48,$D\rightarrow I$=11,$I\rightarrow F$=31
And total distance=195

7. If we start from G then the path is
$G\rightarrow C$=9,$C\rightarrow H$=11,$H\rightarrow J$=16,$J\rightarrow B$=36,$B\rightarrow E$=31,$E\rightarrow F$=1,$F\rightarrow A$=17,$A\rightarrow D$=29,$D\rightarrow I$=11,$I\rightarrow G$=50
And total distance=211

8. If we start from H then the path is
$H\rightarrow C$=11,$C\rightarrow G$=9,$G\rightarrow A$=14,$A\rightarrow E$=17,$E\rightarrow F$=1,$F\rightarrow D$=18,$D\rightarrow I$=11,$I\rightarrow J$=17,$J\rightarrow B$=36,$B\rightarrow H$=42
And total distance=176

9. If we start from I then the path is
$I\rightarrow D$=11,$D\rightarrow E$=18,$E\rightarrow F$=1,$F\rightarrow A$=17,$A\rightarrow G$=14,$G\rightarrow C$=9,$C\rightarrow H$=11,$H\rightarrow J$=16,$J\rightarrow B$=36,$B\rightarrow I$=59
And total distance=192

10. If we start from J then the path is
$J\rightarrow H$=16,$H\rightarrow C$=11,$C\rightarrow G$=9,$G\rightarrow A$=14,$A\rightarrow E$=17,$E\rightarrow F$=1,$F\rightarrow D$=18,$D\rightarrow I$=11,$I\rightarrow B$=59,$B\rightarrow J$=36
And total distance=192

In this study, the NN method has calculated a total shortest distance of 175 Kms for solving this problem with path D→I→A→G→C→H→J→B→E→F→D.

Figure 3. Distance of 175 Km traversed by the tourist through different tourist points of Berhampur city using the nearest neighbor method.

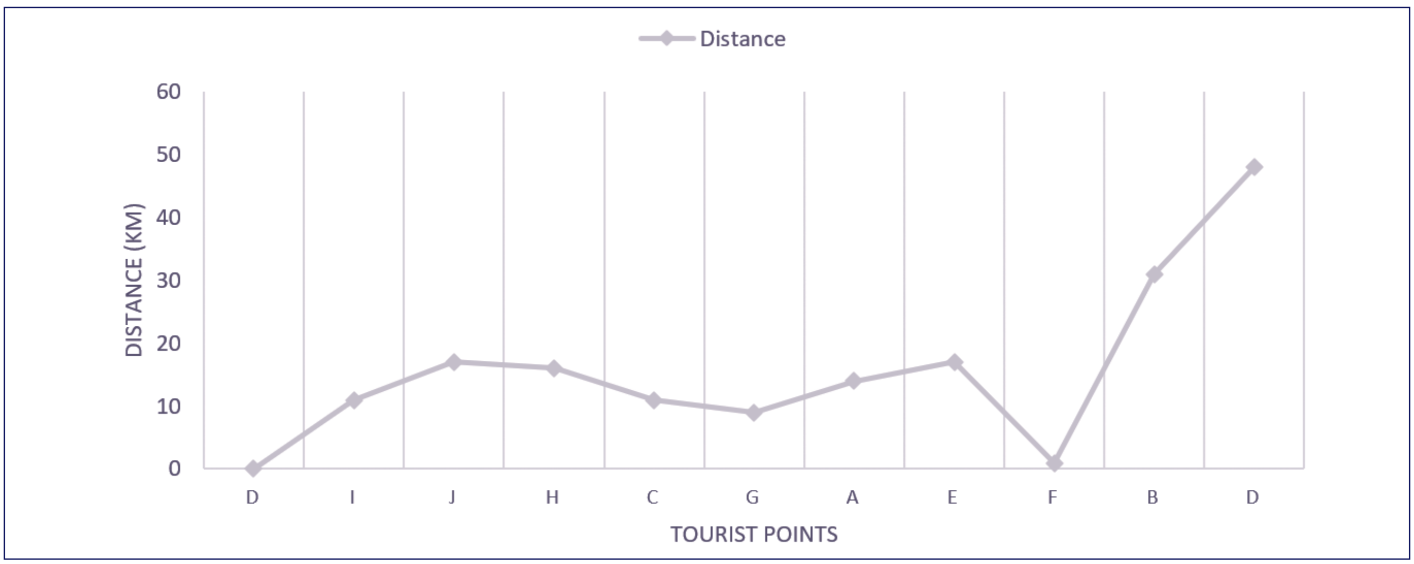
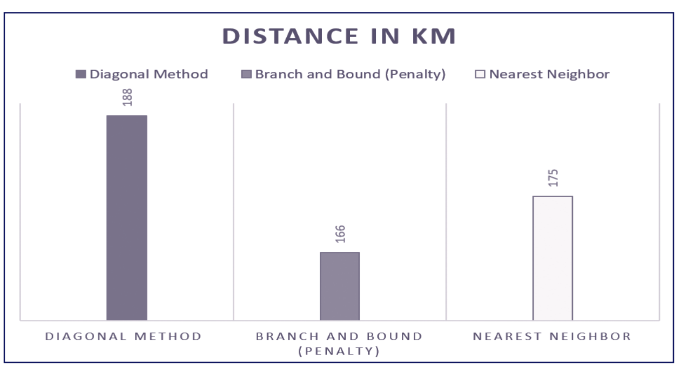


Figure 4. Comparison of diagonal method, branch and bound (penalty method), and nearest neighbor method.



From Figures 1 to 4, it is observed that branch and bound (penalty) method shows less distance than 166 Km if a tourist moves to each tourist place once and return to the starting point. However, the diagonal method and nearest neighbor method shows a distance of 188 Km and 175 Km respectively. Therefore, a tourist uses the final path of D→I→A→G→C→H→J→B→E→F→D.

5. CONCLUSION

In this work, we use Branch & Bound (Penalty Method), Diagonal Completion method and Nearest Neighbour method to solve a Travelling Salesman Problem. The problem is to find the shortest route for the tourist in Berhampur city so that the total distance travelled is minimum. In this work, a case study is taken where a tourist wants to visit 10 different tourist places in Berhampur city. The distance in Kms between the 10 places is considered with different sources and destinations. From this case study, it is found that the minimum transportation cost is 166 Km for Branch & Bound (Penalty method) technique. So, it is clear from the above discussion that Branch & Bound (Penalty) method gives better results for solving this tourist travelling problem in Berhampur city. In future, this problem can be extended by solving it using some more Travelling Salesman Problem-solving approaches. Also, this can be further used to solve the tourist travelling problems in other cities by minimizing the shortest possible routes.

Declarations

Funding: There is no funding information.

Conflicts of Interest: There is no conflict of interest.

Availability of Data and Material: Data is available on request.

Authors Contribution: All authors contributed equally.

REFERENCES

1. *Applegate, David L., Robert E. Bixby, Vašek Chvátal, and William J. Cook (2011). The travelling salesman problem. Princeton university press.*
2. *Atozmath, https://cbom.atozmath.com/Menu/CBomMenu.aspx, Accessed on July 2021.*
3. *Brahmapur, https://odishatourism.gov.in/content/tourism/en/discover/major-cities/brahmapur.html, Accessed on July 2021.*
4. *Davendra, D. (Ed.). (2010). Travelling Salesman Problem: Theory and Applications. BoD–Books on Demand.*
5. *E. Lawler, J. Lenstra and A. H. G. Rinnooy Kan (2011). The Traveling Salesman Problem. Wiley, London.*
6. *Freling, R., Boender, C. G. E., & Paixão, J. M. P. (1993). An integrated approach to vehicle and crew scheduling. Econometric Institute, Erasmus University Rotterdam.*
7. *Freling, R., Huisman, D., & Wagelmans, A. P. (2001). Applying an integrated approach to vehicle and crew scheduling in practice. In Computer-aided scheduling of public transport (pp. 90-73). Springer, Berlin, Heidelberg.*
8. *Friberg, C., & Haase, K. (1999). An exact branch and cut algorithm for the vehicle and crew scheduling problem. In Computer-aided transit scheduling (pp. 80-63). Springer, Berlin, Heidelberg.*
9. *Google Map, https://www.google.com/maps/place/Brahmapur,+Odisha/@19.3082418,84.7732373,13z/data=!3m1!4b1!4m5!3m4!1s0x3a3d500ef1cb60ad:0x5b75778874294ff!8m2!3d19.3149618!4d84.7940911, accessed on August 2021.*
10. *Hayes, M., & Norman, J. M. (1984). Dynamic programming in orienteering: route choice and the siting of controls. Journal of the Operational Research Society, 35(9), 791-796.*
11. *Hubert, L. J., & Baker, F. B. (1978). Applications of combinatorial programming to data analysis: the travelling salesman and related problems. Psychometrika, 43(1), 81-91.*
12. *Işık, E. E., & Şimşir, M. (2023). Maximizing Total Net Profit for Traveling Salesman Problem with Profits Using Metaheuristic Algorithms. The European Journal of Research and Development, 3(1), 46-59.*
13. *Jelokhani-Niaraki, M., Samany, N. N., Mohammadi, M., & Toomanian, A. (2021). A hybrid ridesharing algorithm based on GIS and ant colony optimization through geosocial networks. Journal of Ambient Intelligence and Humanized Computing, 12(2), 2387-2407.*
14. *Krolak, P., Felts, W., & Marble, G. (1971). A man-machine approach toward solving the travelling salesman problem. Communications of the ACM, 14(5), 327-334.*
15. *L. Golden and A. A. Assad (1986). Perspectives in vehicle routing: exciting new developments. Ops Res, 34(5),803-810.*
16. *Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the travelling-salesman problem. Operations research, 21(2), 498-516.*
17. *Little, J. D., Murty, K. G., Sweeney, D. W., & Karel, C. (1963). An algorithm for the travelling salesman problem. Operations research, 11(6), 972-989.*
18. *Mallick, C. (2021). APS-CR: Assignment Problem Solver for Cost Minimization of Crew Routes. Turkish Online Journal of Qualitative Inquiry, 3250-3237 ,(5)12.*
19. *Mallicka, C. (2021). CLAPS: Course and Lecture Assignment Problem Solver for Educational Institution Using Hungarian Method. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 3085-3092.*
20. *Nonato, M., & Gaffi, A. (1999). An integrated approach to extra-urban crew and vehicle scheduling. In Computer-Aided Transit Scheduling (Vol. 471, pp. 128-103). Springer-Verlag.*
21. *Ntakolia, C., & Iakovidis, D. K. (2021). A swarm intelligence graph-based pathfinding algorithm (SIGPA) for multi-objective route planning. Computers & Operations Research, 133, 105358.*
22. *Nuriyeva, F., Kizilates, G., & Berberler, M. E. (2012, September). Experimental analysis of new heuristics for the TSP. In 2012 IV International Conference" Problems of Cybernetics and Informatics"(PCI) (pp. 5-1). IEEE.*
23. *Patrikalakis, I., & Xerocostas, D. (1992). A new decomposition scheme of the urban public transport scheduling problem. In Computer-Aided Transit Scheduling (pp. 425-407). Springer, Berlin, Heidelberg.*
24. *Reinelt, G. (1994). The Traveling Salesman: Computational Solutions for TSP Applications. Springer, Germany.*
25. *Sharma, J. K. (2016). Operations research: theory and applications. Trinity Press, an imprint of Laxmi Publications Pvt. Limited.*
26. *Silva, B. C., Fernandes, I. F., Goldbarg, M. C., & Goldbarg, E. F. (2020). Multi-Strategy MAX-MIN Ant System for Solving Quota Traveling Salesman Problem with Passengers, Incomplete Ride and Collection Time. In Operations Management-Emerging Trend in the Digital Era. IntechOpen.*
27. *Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research, 35(2), 254-265.*
28. *Srinivas, B., & Ganesan, G. (2015). A method for solving branch-and-bound techniques for assignment problems using triangular and trapezoidal fuzzy. International Journal of Management and Social Science, 3, 7-10.*
29. *Syahputra, E., Sitorus, Z., Prabowo, I. R., & Aprijal, R. (2023, July). Optimation of Traveling Salesman Problem (TSP) For Tour Package Routes In Langkat By Algorithm Cheapest Insertion Heuristic (CIH). In Procedding International Conference of Media and Communication in Southeast Asia (Vol. 3, pp. 82-77).*