Integra: Journal of Integrated Mathematics and Computer Science

Journal Homepage: https://integrajimcs.com/index.php/integra/index

integra

Journal of Integrated Mathematics and Computer Science

Research Paper

Comparative Analysis of CIH and Christofides Algorithms for Optimal Tourist Route Planning in West Java

Nur Wafiqoh Hadi[1]*, Rehsya Nurfabella[1], Wamiliana[1], Mira Mustika[2]

[1] Department of Mathematics, Faculty of Mathematics and Natural Sciences, University Lampung, Lampung, 35145, Indonesia

[2] Department of Mathematics, Faculty of Sciences, Institut Teknologi Sumatera, Lampung, 35365, Indonesia

***Corresponding author**: nurwafiqoh@gmail.com

Keywords

TSP, Cheapest Insertion Heuristic, Christofides, Recreational Spot, West Java

Abstract

Efficient route planning plays a crucial role in supporting tourism development, particularly in regions with numerous scattered attractions such as West Java, Indonesia. This study addresses the Traveling Salesman Problem (TSP) by comparing two algorithmic approaches: the Cheapest Insertion Heuristic (CIH) and the Christofides algorithm, to determine the shortest tour among 20 selected tourist sites. Using travel time data obtained from Google Maps, both algorithms were implemented manually and using Phython language programming. The manual application of the CIH algorithm resulted in a total travel time of 813 minutes, which was later optimized to 764 minutes after adjustments to eliminate intersecting paths. Meanwhile, the CIH algorithm implemented in python provided a final route of 717 minutes. In contrast, the Christofides algorithm yielded consistent results for both manual and Python-based calculations, producing a tour with a total travel time of 746 minutes. The findings suggest that the CIH algorithm using Python language offers the most efficient route in this case study. This research contributes to the development of intelligent tour planning systems and can be a valuable reference for optimizing regional tourism logistics.

Received: 26 May 2025, Accepted: 23 July 2025

https://doi.org/10.26554/integra.20252231

1. INTRODUCTION

Tourism is one of the sectors that plays an important role in supporting regional economic growth [1]. West Java, especially in District Bandung as one of the leading tourist destinations in Indonesia has many tourist attractions spread in various locations, both natural, cultural, culinary and educational tourism. The large number of tourist locations requires efficient travel route planning. Efforts to minimize time and also make travel efficient are called optimization. One of the problems involving the optimization process is the Travelling Salesman Problem (TSP).

TSP is the process of determining the travel route of a traveler starting from one starting point, visiting several predetermined points, and returning to the starting point with the aim of minimizing the total distance traveled, where each city is visited only once [2]. The Traveling Salesman Problem (TSP) is one of the most widely studied combinatorial optimization problems, partly due to its wide area of application in logistics, manufacturing, telecommunications, and others [3]. By visiting the destination location points only once per point, the Traveling Salesman Problem (TSP) approach can be utilized to identify the most efficient travel points [4].

TSP has been widely researched and many methods have been developed, both exact and heuristic methods. Among the methods that have been developed are Brute Force [5], Nearest Neighbour [6], Genetic Algorithm [7], Particle Swarm Optimization [8], Branch and Bound [9], Combinatorial Bees [10], and many more. The Christofides algorithm, also known as the Christofides-Serdyukov algorithm, can be used to find an approximate solution to the traveling salesman problem [11]. The

Christofides algorithm was implemented to solve the tour for conventional markets by Aswin et al., [12].

Other algorithms commonly used in finding TSP are Best First Search [13], Genetics [7], Improved Ant Colony [14], Nearest Neighbor, Cheapest Insertion Heuristic [15, 16, 17], and Djikstra algorithms [18]. The Cheapest Insertion Heuristics method builds a tour from small cycles with minimal weight and successively adds new points [15]. At each step, this algorithm selects the new point and edge that gives the smallest insertion value. The advantage of the Cheapest Insertion Heuristic algorithm is its ability to select the node to be inserted, be it a node that is outside the tour being built or an edge that already exists in the tour [19]. Based on research [20], the Traveling Salesman Problem (TSP) problem can be solved using the Cheapest Insertion Heuristic (CIH) algorithm.

In this research, two algorithmic approaches are used to solve TSP, namely Christofides Algorithm and Cheapest Insertion Heuristic (CIH). By using data on the distance between tourist sites in West Java, this study aims to compare the effectiveness of the two algorithms in producing efficient tourist routes. It is hoped that the results of this research can be a reference in planning tourist trips and developing tour information systems.

Figure 1. Recreation Spots Area in West Java

2. METHODS

2.1 The Cheapest Insertion Heuristic Algorithm

CIH starts by forming a tour from the two closest points or nodes. However, in the method used in this research, a modification is made by starting the tour from the three closest points. The steps are as follows:

1. The search begins by determining the 3 points that have the closest distance, then summing the distance between the three points (total initial weight).
2. Search for candidate edges or lines that will be selected to be included in the new.
3. Calculate each candidate edge that has been selected, using the formula: total weight = weight of the removed edge + weight of the added edge + weight of the edge connecting the removed point and the added point.
4. Determine the smallest weight value of all candidate edges that have been calculated.
5. Form a new tour based on the selected edge.

Repeat the above steps until all points are included in the tour.

```python
def cheapest_insertion(dist_matrix, start=0):
    n = len(dist_matrix)
    visited = [start]
    nearest = np.argmin([dist_matrix[start][j] if j
    visited.append(nearest)
    visited.append(start)
    unvisited = set(range(n)) - set(visited)

    while unvisited:
        best_increase = float('inf')
        best_city = None
        best_position = None

        for city in unvisited:
            for i in range(1, len(visited)):
                prev = visited[i - 1]
                next_ = visited[i]
                increase = dist_matrix[prev][city] +
                if increase < best_increase:
                    best_increase = increase
                    best_city = city
                    best_position = i
```

Figure 2. The Part of the Source Code for CIH

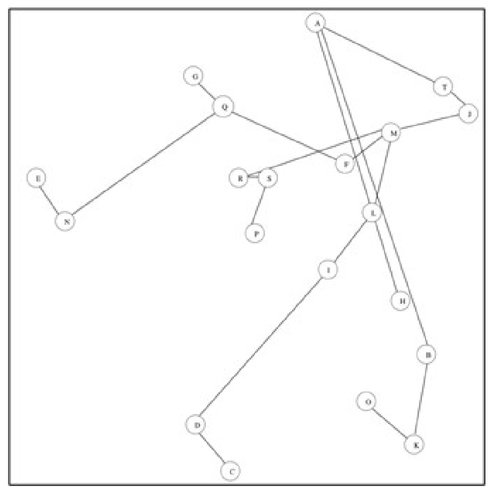

Figure 3. The Result of MST using Kruskal Algorithm

2.2 The Christofides Algorithm

The following is the implementation procedure of the Christofides algorithm:

1. Find the Minimum Spanning Tree (MST) that connects all vertices in the graph (can use Kruskal [21], Prim [22], and Boruvka (Solin) Algorithm [23], in this research, Kruskal is used).
2. Determine the vertices that have odd degree and connect them with other odd degree vertices so that all vertices become even degree.

Table 1. The Information of Recreational Areas in West Java

Node	Name	Address
A	Orchid Forest Cikole	Orchid Forest Cikole, Genteng, Cikole, Lembang, West Bandung, West Java 40391
B	Jelekong Galery	Jelekong 03 / 01, Jelekong, Baleendah, Bandung, West Java 40375
C	Kawah Putih Ciwidey	St. Kawah Putih KM 11, Patengan, Ciwidey, Bandung Regency, West Java 40973
D	Ranca Upas Camp	Ranca Upas Ciwidey, St. Camp Ranca Upas, Patengan, Rancabali, Bandung, West Java 40973
E	Sanghyang Heuleut	Sanghyang Heuleut, Cipanas Village, Rajamandala Kulon, Cipatat, West Bandung, West Java 40554
F	Puncak Punclut	Punclut Puncak, Pagerwangi, St. Punclut, Ciumbuleuit, Cidadap, Bandung City, West Java 40142
G	Dusun Bambu	Bambu Village, St. Kolonel Masturi KM 11, Kertawangi, Cisarua, West Bandung Regency, West Java 40551
H	Kiara Artha Park	Kiara Artha Park, St. Banten, Kebonwaru, Batununggal, Bandung City, West Java 40272
I	Braga Street	St. Braga, Sumur Bandung, Bandung City, West Java
J	Mycelia Cikole Forest	Mycelia Forest Educational Park, Cikole, Lembang, Bandung Barat, West Java 40391
K	Sunrise Point Cukul	Sunrise Point Cukul, St. Cukul, Sukaluyu, Pangalengan, Bandung Regency, West Java 40378
L	Dago Dream Park	Dago Dreampark, Mekarwangi, St. Dago Giri KM 2.2, Pagerwangi, Lembang, West Bandung, West Java 40135
M	Kebun Begonia Lembang	Kebun Begonia, Jl. Maribaya No.120 A, Langensari, Lembang, West Bandung Regency, West Java 40391
N	Cikahuripan Green Canyon Saguling	Cikahuripan Green Canyon Saguling, Cianjur Saguling, Saguling, Batujajar, West Bandung, West Java 40561
O	Taman Langit Pangalengan	Taman Langit Pangalengan (Sunrise Point & Camping Ground), Puncak Mulya, St. Cukul, Sukaluyu, Pangalengan, Bandung Regency, West Java 40378
P	Amazing Art & Games	Amazing Artgames, St. Dr. Setiabudi No.293-295, Isola, Sukasari, Bandung, West Java 40154
Q	Lembang Park & Zoo	Lembang Park & Zoo, St. Kolonel Masturi No.171, Sukajaya, Lembang, West Bandung, West Java 40391
R	The Great Asia Africa Lembang	The Great Asia Africa, St. Raya Lembang - Bandung No.71, Gudangkahuripan, Lembang, West Bandung, West Java 40391
S	Farmhouse Lembang	Farm House Susu Lembang, St. Raya Lembang - Bandung No.71, Gudangkahuripan, Lembang, West Bandung, West Java 40391
T	Grafika Cikole	St. Raya Tangkuban Parahu No. KM.8, Cikole, Lembang, West Bandung, West Java 40391

3. Reduce the vertices that have more than two degrees until all vertices have two degrees, thus forming a Hamiltonian tour.

2.3 Data

The data used in this research was collected from Google Maps on May 16, 2025, at 10:00 AM until 02.00 PM (WIB). Figure 1 shows the map of the recreation spots area in West Java, while Table 1 shows the information about those locations. Table 2 displays the travel time in minutes between recreation spots area in West Java Regency using a four-wheeled vehicle.

Table 1 shows the information information about the locations of recreational areas in West Java Regency

3. RESULTS AND DISCUSSION

3.1 The CIH Algorithm

Based on the algorithm steps previously described, manual calculations will be performed using the data in Table 2. For space efficiency, we use table. Table 3 displays the iteration number, the subtour formed, the total weight in every iteration, added and removed edges, total weight after added/removed weight, and the new subtour.

Table 2. The Distance Between Locations (in Minutes)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	0	90	126	127	129	30	44	124	59	32	163	35	21	129	167	36	34	29	28	6
B	90	0	102	90	103	72	77	43	50	89	107	72	83	103	112	63	77	71	69	89
C	126	102	0	6	135	106	111	91	87	132	117	109	122	129	123	103	109	101	110	133
D	127	90	6	0	137	106	127	106	81	128	113	104	116	125	117	96	113	105	106	128
E	129	103	135	137	0	120	104	115	112	137	185	140	127	13	190	124	107	118	108	136
F	30	72	106	106	120	0	43	41	38	31	148	15	15	109	152	33	30	25	24	31
G	44	77	111	127	104	43	0	70	62	44	146	51	37	99	151	32	14	30	30	47
H	124	43	91	106	115	41	70	0	21	68	141	38	51	97	146	42	59	51	65	71
I	59	50	87	81	112	38	62	21	0	61	135	29	43	123	137	38	50	53	53	74
J	32	89	132	128	137	31	44	68	61	0	174	31	17	150	179	33	31	24	25	1
K	163	107	117	113	185	148	146	141	135	174	0	145	157	189	8	148	150	155	155	178
L	35	72	109	104	140	15	51	38	29	31	145	0	13	132	155	39	38	29	30	17
M	21	83	122	116	127	15	37	51	43	17	157	13	0	167	167	25	24	16	17	31
N	129	103	129	125	13	109	99	97	123	150	189	132	167	0	183	110	74	119	127	136
O	167	112	123	117	190	152	151	146	137	179	8	155	167	183	0	152	155	160	159	184
P	36	63	103	96	124	33	32	42	38	33	148	39	25	110	152	0	21	12	13	43
Q	34	77	109	113	107	30	14	59	50	31	150	38	24	74	155	21	0	18	18	35
R	29	71	101	105	118	25	30	51	53	24	155	29	16	119	160	12	18	0	2	28
S	28	69	110	106	108	24	30	65	53	25	155	30	17	127	159	13	18	2	0	28
T	6	89	133	128	136	31	47	71	74	1	178	17	31	136	184	43	35	28	28	0

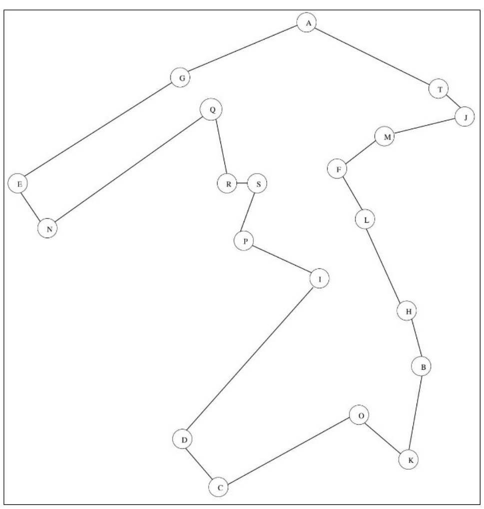

Figure 4. The Result after Removing Some Odd Degree Vertex

After 18 iterations, the result is 813 minutes. However, there are many intersecting lines on the tours. Thus, it is still possible for us to fix the tour by removing intersecting lines to get better solution. There are many possible way do that, by doing some deletion and addition of lines so that there are no intersections. The results obtained after deleting the intersecting lines and adding other lines obtained the final result is 764 minutes with the new tour, namely E-G-A-T-J-M-F-L-H-B-K-O-C-D-I-P-S-R-Q-N-B.

In addition to manual calculations, calculations are also performed using Python language programming. Figure 2 shows part of the source code.

```
# Graph plot
plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
nx.draw(G, pos, with_labels=True, node_color='skyblue', node_si
edge_labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels)
plt.title("Graf Lengkap")

# MST Plot
mst_G = nx.Graph()
for u, v, w in mst_edges:
    mst_G.add_edge(u, v, weight=w)

plt.subplot(1, 2, 2)
nx.draw(mst_G, pos, with_labels=True, node_color='lightgreen',
edge_labels_mst = {(u, v): w for u, v, w in mst_edges}
nx.draw_networkx_edge_labels(mst_G, pos, edge_labels=edge_label
plt.title("Minimum Spanning Tree (Kruskal)")

plt.tight_layout()
plt.show()
```

Figure 5. The Part of Source Code for Determine the MST (Kruskal Algorithm)

The program produces output with tour E-G-P-S-R-A-T-J-M-L-I-C-D-O-K-B-H-F-Q-N-E and a total weight of 772 minutes. However, there is an intersection of lines on the tour. Therefore, we do fix the tour by doing deletion and addition of lines so that there is no intersection, and we get better solution after deleting and adding lines is 717 minutes with tour E-G-Q-P-R-S-F-M-A-T-J-L-I-H-B-K-O-C-D-N-E.

3.2 The Christofides Algorithm

Christofides' algorithm generates many possible solutions as there are various ways to connect odd degree vertices in the graph. The process of this algorithm begins with the formation

Table 3. The All Iterations of CIH

Iteration	Subtour	Total Weight	E=n?	Added Line/ Edge	Discard Edge	Total Weight after Addition	New Subtour
1	E-N-Q-E	194	No	Q,G G,E	Q,E	205	E,N,Q,G,E
2	E,N,Q,G,E	205	No	N,H H,Q	N,Q	289	E,N,H,Q,G,E
3	E,N,H,Q,G,E	289	No	H,I I,Q	H,Q	299	E,N,H,I,Q,G,E
4	E,N,H,I,Q,G,E	299	No	P,Q I,P	I,Q	308	E,N,H,I,P,Q,G,E
5	E,N,H,I,P,Q,G,E	308	No	P,R I,R	I,P	335	E,N,H,I,R,P,Q,G,E
6	E,N,H,I,R,P,Q,G,E	335	No	I,L L,R	I,R	340	E,N,H,I,L,R,P,Q,G,E
7	E,N,H,I,L,R,P,Q,G,E	340	No	L,M M,R	L,R	340	E,N,H,I,L,M,R,P,Q,G,E
8	E,N,H,I,L,M,R,P,Q,G,E	340	No	S,R M,S	M,R	343	E,N,H,I,L,M,S,R,P,Q,G,E
9	E,N,H,I,L,M,S,R,P,Q,G,E	343	No	F,S M,F	M,S	365	E,N,H,I,L,M,F,S,R,P,Q,G,E
10	E,N,H,I,L,M,F,S,R,P,Q,G,E	365	No	L,T T,M	L,M	400	E,N,H,I,L,T,M,F,S,R,P,Q,G,E
11	E,N,H,I,L,T,M,F,S,R,P,Q,G,E	400	No	T,J J,M	T,M	387	E,N,H,I,L,T,J,M,F,S,R,P,Q,G,E
12	E,N,H,I,L,T,J,M,F,S,R,P,Q,G,E	387	No	A,T L,A	L,T	411	E,N,H,I,L,A,T,J,M,F,S,R,P,Q,G,E
13	E,N,H,I,L,A,T,J,M,F,S,R,P,Q,G,E	411	No	N,B B,H	N,H	460	E,N,B,H,I,L,A,T,J,M,F,S,R,P,Q,G,E
14	E,N,B,H,I,L,A,T,J,M,F,S,R,P,Q,G,E	460	No	I,D D,L	I,L	607	E,N,B,H,I,D,L,A,T,J,M,F,S,R,P,Q,G,E
15	E,N,B,H,I,D,L,A,T,J,M,F,S,R,P,Q,G,E	607	No	D,C C,L	D,L	618	E,N,B,H,I,D,C,L,A,T,J,M,F,S,R,P,Q,G,E
16	E,N,B,H,I,D,C,L,A,T,J,M,F,S,R,P,Q,G,E	618	No	O,B N,O	N,B	810	E,N,O,B,H,I,D,C,L,A,T,J,M,F,S,R,P,Q,G,E
17	E,N,O,B,H,I,D,C,L,A,T,J,M,F,S,R,P,Q,G,E	810	No	O,K K,B	O,B	813	E,N,O,K,B,H,I,D,C,L,A,T,J,M,F,S,R,P,Q,G,E
18	E,N,O,K,B,H,I,D,C,L,A,T,J,M,F,S,R,P,Q,G,E	813	Yes				

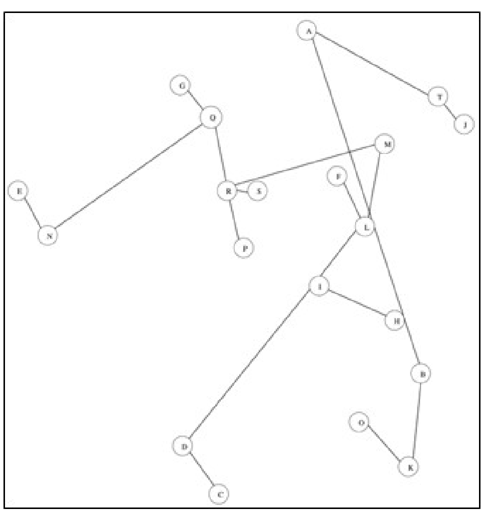

Figure 6. The Result MST using Python

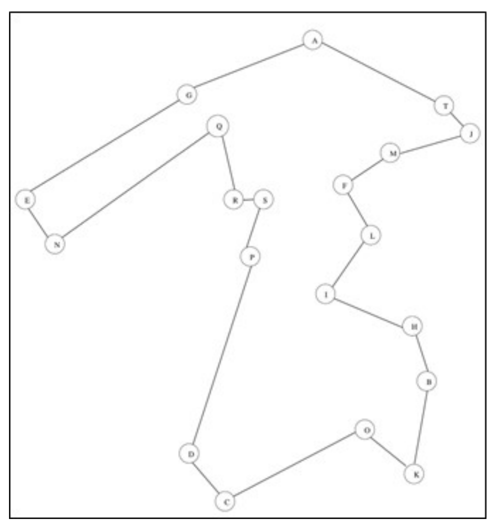

Figure 7. The Result After Removing Some Odd Degree Vertex

of a Minimum Spanning Tree (MST) of 20 vertices using Kruskal's Algorithm. Figure 3 displays the result.

The next step is to form an Eulerian tour, by connecting vertices of odd degree so that all vertices in the graph have even degree. As discussed earlier, there are various possibilities in selecting odd degree vertex pairs, resulting in various possible Eulerian paths.

Figure 4 shows one possible result, where the odd vertices have been paired to be even. To obtain the Hamiltonian tour, the Euler path is then altered by eliminating shortcutting, i.e. by removing edges that cause vertices to have more than two degrees, resulting in an optimal Hamiltonian tour with a total weight of 746 minutes.

Beside doing it manually, the Christofides Algorithm solution is also done using Python programming language. Figure 5 shows part of the source code for finding the MST, as the fisrt step of Christofides'algorithm, and Figure 6 displays the MST obtained.

The MST obtained is as shown in the figure. Next, we will search for the Hamiltonian tour manually.

By connecting the odd degree vertices and reducing the vertices whose degree more than 2, the Hamiltonian tour is found with tour E-G-A-T-J-M-F-L-I-H-B-K-O-C-D-P-S-R-Q-N-E and a total weight of 746 minutes. Figure 7 shows the result.

4. CONCLUSIONS

Based on the Results and Discussion section, the calculation using the CIH algorithm manually resulted in a total weight of 813 minutes, while w, a total weight of 717 minutes was obtained with the tour E-G-Q-P-R-S-F-M-A-T-J-L-I-H-B-K-O-C-D-N-E. Meanwhile, by using the Christofides algorithm, both manually and Phyton, resulted in the same total weight of 746 minutes with the tour E-G-A-T-J-M-F-L-I-H-B-K-O-C-D-P-S-R-Q-N-E. Of all the methods used, the best calculation results are obtained through the CIH algorithm with Phyton programming, with a total weight of 717 minutes and the tour is E-G-Q-P-R-S-F-M-A-T-J-L-I-H-B-K-O-C-D-N-E. Note that the result obtained is the result after fixing the interconnecting lines in the tour.

5. ACKNOWLEDGEMENT

The authors would like to thank the Mathematics and Statistics Research Laboratory at Universitas Lampung University for its support.

REFERENCES

[1] Hasana Fadilla. Pengembangan Sektor Pariwisata untuk Meningkatkan Pendapatan Daerah di Indonesia. *Benefit: Journal of Bussiness, Economics, and Finance*, 2(1):36–43, 2024.

[2] Miko Purnomo, Justin Eduardo Simarmata, and Debora Chrisinta. Optimasi Jarak Terpendek untuk Mencapai Wisata Super Prioritas Indonesia dengan Menggunakan Pendekatan Traveling Salesman Problem. *MathVision: Jurnal Matematika*, 7(1):53–61, 2025.

[3] Christoph Hansknecht, Imke Joormann, and Sebastian Stiller. Dynamic Shortest Paths Methods for the Time-Dependent TSP. *Algorithms*, 14(1):21, 2021.

[4] Nasywa Shafa Azzahra, Najwa Nayra Aulia, Arista Binarsih, and P Paduloh. Analisis Optimasi Jalur Distribusi Menggunakan Pendekatan TSP (Traveling Salesman Problem) untuk Meningkatkan Efisiensi Biaya Distribusi Pada Toko Uthe Grosir. *HUMANITIS: Jurnal Homaniora, Sosial Dan Bisnis*, 2(6):542–553, 2024.

[5] Sriyani Violina. Analysis of Brute Force and Branch & Bound Algorithms to Solve the Traveling Salesperson Problem (TSP). *Turkish Journal of Computer and Mathematics Education*, 12(8):1226–1229, 2021.

[6] Md Ziaur Rahman, Sakibur Rahamn Sheikh, Ariful Islam, and Md Azizur Rahman. Improvement of the Nearest Neighbor Heuristic Search Algorithm for Traveling Salesman Problem. *Journal of Engineering Advancements*, 5(01):19–26, 2024.

[7] Noraini Mohd Razali, John Geraghty, et al. Genetic algorithm performance with different selection strategies in solving TSP. In *Proceedings of the world congress on engineering*, volume 2, pages 1–6. International Association of Engineers Hong Kong, China, 2011.

[8] Xuesong Yan, Can Zhang, Wenjing Luo, Wei Li, Wei Chen, and Hanmin Liu. Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm. *International Journal of Computer Science*, 9(6):264–271, 2012.

[9] S Dhanasekar, Saroj Kumar Dash, and Neena Uthaman. A Branch and Bound Algorithm to Solve Travelling Salesman Problem (TSP) with Uncertain Parameters. *Mathematics and Statistics*, 10(2):358–365, 2022.

[10] Murat Sahin. Solving TSP by Using Combinatorial Bees Algorithm with Nearest Neighbor Method. *Neural Computing and Applications*, 35(2):1863–1879, 2023.

[11] Dwi Rizka Amelia Putri, Niken Sabella Oktavia, Siti Laelatul Chasanah, Riza Sawitri, and Felicia Andrade Paskalia. Implementation of Christofides Algorithm to Determine the Shortest Tour of Some Hospitals in Palembang City. *Integra: Journal of Integrated Mathematics and Computer Science*, 2(1):15–19, 2025.

[12] Micelle Yap Aswin, Wamiliana Wamiliana, Fitriani Fitriani, Muslim Ansori, and Notiragayu Notiragayu. Perbandingan Cheapest Insertion Heuristic dan Algoritma Christofides untuk Menentukan Tour Pasar Tradisional di Kota Bandar Lampung. *Jurnal Pepadun*, 5(2):182–194, 2024.

[13] Etza Nofarita Nofarita. The Utilization of the Best First Search Algorithm in the Solution of the Traveling Salesman Problem Case in City X. *IJISTECH (International Journal of Information System and Technology)*, 5(4):360–365, 2021.

[14] Xuan-Shi Yao, Yun Ou, and Kai-Qing Zhou. TSP Solving Utilizing Improved Ant Colony Algorithm. In *Journal of Physics: Conference Series*, volume 2129, page 012026. IOP Publishing, 2021.

[15] Annisa Salsabila, Nadhir Rotun Nikmah, Rafif Syadid Bakhtiananda, Micelle Yap Aswin, and Dina Eka Nurvazly. Products Distribution from Suppliers to Retailers in Bandarlampung City (Case Study: Retailers location in Teluk Betung). *Integra: Journal of Integrated Mathematics and*

Computer Science, 1(1):6–12, 2024.

[16] Abdi Restu Dinata, Wamiliana Wamiliana, Muslim Ansori, Fitriani Fitriani, and Notiragayu Notiragayu. Determining the Shortest Tour Location of Tourist Attractions in Bandar Lampung Using Cheapest Insertion Heuristic (CIH) and Modified Sollin Algorithm. *Jurnal Pepadun*, 6(1):92–102, 2025.

[17] K Pavani, M Anitha, and B Priyanka. Delivery Route Optimization Using Dijkstra's Algorithm for the Travelling Salesman Problem. *International Journal of Advance Scientific Research*, 9(6), 2025.

[18] Farid Fargiana, Respitawulan Respitawulan, Yusuf Fajar, Didi Suhaedi, and Erwin Harahap. Implementation of Cheapest Insertion Heuristic Algorithm in Determining Shortest Delivery Route. *International Journal of Global Operations Research*, 3(2):37–45, 2022.

[19] Rizki Putra Sinaga and Faridawaty Marpaung. Perbandingan Algoritma Cheapest Insertion Heuristic dan Nearest Neighbor dalam Menyelesaikan Traveling Salesman Problem. *Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam*, 2(2):238–247, 2023.

[20] Ahya Sofa Ananda, Notiragayu Notiragayu, Wamiliana Wamiliana, and Muslim Ansori. Penentuan Lintasan Terpendek Perjalanan Pengiriman Barang Menggunakan Algoritma Cheapest Insertion Heuristic (Studi Kasus PT. Indah Logistik Cargo Bandar Lampung). *Jurnal EurekaMatika*, 11(2):111–120, 2023.

[21] Jogamohan Medak. Review and Analysis of Minimum Spanning Tree Using Prim's Algorithm. *International Journal of Computer Science Trends and Technology*, 6(2), 2018.

[22] Jiayu Chen. The Analysis and Application of Prim Algorithm, Kruskal Algorithm, Boruvka Algorithm. *Applied and Computational Engineering*, 19:84–89, 2023.

[23] Dian Rachmawati, Frederik Yan Putra Pakpahan, et al. Comparative Analysis of the Kruskal and Boruvka Algorithms in Solving Minimum Spanning Tree on Complete Graph. In *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pages 55–62. IEEE, 2020.