P-ISSN 1991-8941
E-ISSN 2706-6703
Vol. 20, No. 2 (2026)

Journal of University of Anbar for Pure Science (JUAPS)

journal homepage: juaps.uoanbar.edu.iq

Open Access

Original Paper

A comparative study of heuristic and exact algorithms: the case of cheapest insertion heuristic and branch and bound algorithm

Shams Raed Abdulkareem [1], **Waleed Mohammed Elaibi** [1*]

[1] Department of Statistics, College of Administration and Economics, Mustansiriyah University, Iraq

ARTICLE INFO

Received: 22/09/2025
Revised: 10/12/2025
Accepted: 18/12/2025
Available online: 08/08/2026
August Issue
10.37652/juaps.2025.165439.1711

CITE @ JUAPS

Corresponding author
Waleed Mohammed Elaibi
waleed_m@uomustansiriyah.edu.iq

ABSTRACT

The Traveling Salesman Problem (TSP) is one of the most prominent combinatorial optimization problems, attracting significant interest in both research and practical applications. This paper compares the performance of two approaches for solving the TSP: the Cheapest Insertion Heuristic (CIH) and the Branch-and-Bound (B&B) algorithm. The results show that CIH produced a route of 584 km. This approach is simple and fast to execute, but it provides approximate solutions and does not guarantee optimality. In contrast, B&B was effective in finding the optimal solution by relying on lower-bound (LB) calculations and systematically eliminating nonpromising branches. The best route obtained using B&B was A $\to$ D $\to$ C $\to$ B $\to$ E $\to$ A. The comparison suggests that CIH is more suitable for large-scale problems requiring quick solutions, whereas B&B is preferable when obtaining the optimal solution is essential, despite its higher computational cost and longer execution time.

Keywords: *Branch and bound algorithm, CIH, Geographic Information Systems (GIS), traveling salesman problem*

1 INTRODUCTION

The Branch-and-Bound algorithm is one of the exhaustive search techniques used in combinatorial optimization problems, where the objective is to identify the optimal solution from a set of feasible alternatives. This algorithm is particularly relevant to integer programming, sequencing problems, and routing problems, such as the Traveling Salesman Problem (TSP).

The core concept of the algorithm is to decompose a large problem into smaller subproblems, referred to as branches. Each subproblem is then evaluated using bounds to determine whether it has the potential to contain the optimal solution or can be eliminated without examining all possible alternatives.

Branch-and-Bound has wide applications in industrial resource optimization and mineral resource management, including raw material extraction and production-chain design. It can play an important role in reducing costs while maximizing economic value in the exploitation of natural resources such as limestone, gravel, sand, and secondary gypsum.

In addition, the algorithm can be integrated with Geographic Information Systems (GIS) to address spatial resource optimization and complex geographic decision-making. This integration supports improved spatial planning, optimal site selection, and cost-efficient routing, while also considering environmental and geographical constraints.

Particular emphasis is placed on mineral resources, specifically limestone, gravel, sand, and secondary gypsum, because of their potential to generate significant economic returns. The combination of Branch-and-Bound with GIS provides a useful framework for analysis and planning, enabling optimized resource extraction, improved transportation planning, and more comprehensive environmental-impact assessment [1, 2].

1.1 Overview of limestone material iraqi geological survey / ministry of industry and minerals

Limestone, also known as calcium carbonate rock ($CaCO_3$), is a sedimentary rock formed over millions of years through the accumulation and consolidation of calcium carbonate deposits. Limestone originates through several natural processes, including the following:

- Accumulation of marine organism remains: Limestone commonly forms from the deposition of shells, coral fragments, and other calcium carbonate-rich remains on the floors of ancient seas and oceans.
- Chemical precipitation: Calcium carbonate can gradually precipitate from calcium-rich waters through chemical interactions with the surrounding environment.
- Geological compaction and tectonic activity: Over geological time, compaction, pressure, and tectonic processes help solidify loose carbonate deposits and transform them into hard rock.

Some of the oldest limestone formations are attributed to these combined processes, particularly compaction and tectonic activity, which played a key role in converting loose sediments into solid limestone rock.

Fig. 1 The limestone

2 THE ROLE OF GEOGRAPHIC INFORMATION SYSTEMS (GIS) IN SUPPORTING SPATIAL DECISIONS

Geographic Information Systems (GIS) are powerful tools for supporting spatial decision-making through the analysis, representation, and integration of geographic data with descriptive information. These systems rely on spatial analysis techniques, data modeling, and visualization to provide accurate information that supports planning and decision-making.

GIS also facilitates the collection, storage, management, and analysis of spatial data, including topographic data, land-use patterns, soil types, and hydrological characteristics. Such data are essential for modeling and analyzing watershed characteristics [3–5].

3 THE IMPORTANCE OF GEOGRAPHIC INFORMATION SYSTEMS (GIS) IN MANAGING MINERAL RESOURCES

A Geographic Information System (GIS) consists of four main subsystems: data collection, data management, data analysis, and information production. One of the main advantages of GIS is that it functions as a central database for storing relevant data, making information accessible to users whenever needed. It also promotes data sharing and supports teamwork among stakeholders.

3.1 Data collection and integration

Integrating topographic, land-use, and soil data within a GIS provides a comprehensive view of mineral resource locations. This holistic perspective enables stakeholders to understand the relationships among different environmental elements and develop effective management strategies. For example, GIS can be used to model land-use change scenarios and estimate their potential impacts on mineral resources. This supports planning and decision-making with greater accuracy and efficiency, particularly in relation to mineral management and extraction methods [5–7].

4 THE TRAVELING SALESMAN PROBLEM USING THE CHEAPEST INSERTION HEURISTIC (CIH)

The Cheapest Insertion Heuristic (CIH) is one of the insertion-based algorithms used to solve the Traveling Salesman Problem (TSP). At each step, a new city is added to the partial tour, or subtour, at the position that produces the minimum insertion cost. The algorithm may produce different routes depending on the order in which cities are inserted into the partial tour. CIH is considered suitable for TSP instances involving a large number of cities because it provides fast approximate solutions.

The TSP can be solved using several heuristic methods, and among these approaches, CIH can be applied effectively when a quick and computationally simple solution is required.

5 STEPS FOR APPLYING THE CIH ALGORITHM:

Step 1: The search starts from a first location, which is connected to a final location.
Step 2: A subtour is created between these two locations. This means a route that starts at the first location and ends at the same location.
Step 3: One of the arcs between the two locations is replaced with two new arcs, and is chosen from the locations not yet included in the subtour in such a way that the increase in distance is minimal.
Step 4: Repeat Step 3 until all locations are included in the subtour.

Insertion methods start from any random point to form a subtour or partial tour. Then, nodes not yet in the subtour are inserted based on predefined criteria to minimize the total distance increase of the subtour.
Definition 1: Suppose a node is to be added to an edge; according to the cost function, then:

$$c(i,k,j) = d(i,k) + d(k,j) - d(i,j) \tag{1}$$

Definition 2: Suppose we have a Traveling Salesman Problem (TSP) graph. (N, d), a subtour T, and a node in N not in T. Then is defined as the subtour built according to the following rules:
1. If T passes through more than one point:

a Find an edge (x, y) in T that minimizes.

b Delete edge (x, y), and add edges (x, k) and (k, y) to obtain Tour(T, k).

2. If T passes through only one node (i), then it is defined Tour(T, k) as the tour with two nodes consisting of edges $(i, k), (k, i)$. In both cases, we say Tour(T, k) that is obtained by inserting a node (k) into T.

The goal of each insertion method is to add a node to an edge (i.e., between two nodes) at minimal cost. Knowing the subtour ^{T} and that it is the next node (k) to be inserted, the insertion technique inserts (k) between i^*, j^* in T_i according to:

$$(i^*, j^*) = \underset{(ij)\in T}{\arg\min} c(i,k,j) \tag{2}$$

Insertion techniques are desirable because of their speed, ease of implementation, quality of solutions, and adaptability to complex constraints. There are four generally well-known insertion techniques: nearest insertion (used in this work), cheapest insertion, random insertion, and farthest insertion. Other techniques include most expensive insertion, quick insertion, and largest-angle insertion.

This method is similar to the nearest insertion method. We start from node i, which can be either a random or a fixed node, and identify the cities (i, k, j), where i and j are the endpoints of an edge belonging to the partial tour, and k does not belong to that tour, in such a way that $ci + ck - cj$ is minimized. Once all nodes have been selected, we stop; otherwise, we repeat the process. [4, 8–14].

6 TRAVELLING SALESMAN PROBLEM USING THE BRANCH AND BOUND ALGORITHM

The Traveling Salesman Problem (TSP) is one of the most extensively studied combinatorial optimization problems in the literature. Its application scope is broad and diverse in real-life situations.

Assume that the salesman must visit (n) cities, denoted by $(1, 2, 3, \ldots, n)$. The salesman departs from a base city, indexed as city 0, visits each of the other (n) cities exactly once, and then returns to city 0. During the trip, the salesman is required to return to city 0 exactly (t) times, including the final return, where the value of (t) may vary. In addition, the salesman is not allowed to visit more than (P) cities in a single tour. In this context, a “tour” refers to a sequence of visits to cities that begins and ends at city 0. The objective is to determine a route that minimizes the total distance traveled by the salesman.

The Branch-and-Bound algorithm is one of the methods used to solve the TSP. The mathematical model for the Branch-and-Bound algorithm relies on a formal formulation of the spatial problem to be solved. This includes defining the solution space, constraints, and objective function. The following is a general mathematical formulation that can be customized according to the nature of the spatial problem [3, 15–17].

7 DETAILED STEPS

1. Branching:
   Initially, all possible paths between cities, representing the complete solution space, are considered. The solution space is then divided into two main cases:

   - Including a specific path: For example, specifying that the path between cities A and B is included in the solution.
   - Excluding a specific path: For example, specifying that the path between cities A and B is excluded from the solution.

2. Bounding:
   The minimum possible total distance for each branch is estimated using relaxation methods, such as:

   - Temporarily relaxing or ignoring some constraints.
   - Using fast algorithms to compute an initial lower-bound estimate, such as Kruskal's algorithm for constructing a minimum spanning tree.

3. Pruning:
   If the calculated lower bound for a particular branch is greater than the current best feasible solution, that branch is discarded because it cannot lead to an optimal solution.

8 STEPS OF THE BRANCH AND BOUND ALGORITHM

1. Formulate the Mathematical Model:

   - Objective Function: Minimize the total distance

$$f(x)\,\text{min} = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} d_{ij} \tag{3}$$

   Such that:
   d_{ij} : The distance between city i and city j
   x_{ij} : decision variable $x_{ij} = 1$, if the path between i and j is chosen, and 0 otherwise.

   - Flow constraints
     Constraint of visiting each city only once:

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1 \quad \forall i \tag{4}$$

   Constraint on leaving each city only once:

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1 \quad \forall j \tag{5}$$

   Constraints on the absence of sub-courses:

$$U_i - U_j + nX_{ij} \le n - 1 \tag{6}$$

   Binary Constraints:

$$X_{ij} = \begin{cases} 1 & \text{If we go from city i to city j} \\ 0 & \text{Otherwise} \end{cases}$$

$$X_{ij} \in \{0, 1\}, \forall i, j \tag{7}$$

2. Bounding:
   A lower bound or upper bound of the objective function is determined to evaluate the effectiveness of sub-solutions.

   Example of a lower bound in the TSP problem: an initial estimate can be calculated using relaxation methods, such as linear programming.

3. Solution Space:

   - S: the set of all possible solutions (which could be a set of locations or assignments).
   - S is divided into smaller subgroups (subspaces) during the branching process.

9 MATHEMATICAL OPERATIONAL STEPS

1. Find the minimum/maximum (bounding) for each subset Si.

$$f(x) \min_{x\in S} = B(S_i) \tag{8}$$

   This is done using relaxation methods, such as linear relaxation or quick initial solutions.

$$B\left(S_i\right) = \sum_{i=1}^{n} \left(\min 1_i + \min 2_i\right) \tag{9}$$

$$LB = \frac{\sum_{i=1}^{n} B\left(S_i\right)}{2} \tag{10}$$

2. Branching: The group Si is divided into smaller subgroups based on specific criteria, such as:
   - Dividing geographic points based on coordinates.
   - Dividing possible paths.
   - Calculating the number of possible branches from a path of length k.
3. Pruning: If the minimum $B(Si)$ of a node is worse than the current solution $f\left(x^*\right)$, Si is discarded.
4. Iteration: The process continues until all nodes are analyzed or a satisfactory solution is reached. [18–20]

$$\text{If} \quad \left.\begin{matrix} LB_{\text{first}} \geq LB_{\text{new}} & \text{Pruning} \\ LB_{\text{first}} < LB_{\text{new}} & \text{Choose} \end{matrix}\right\}$$

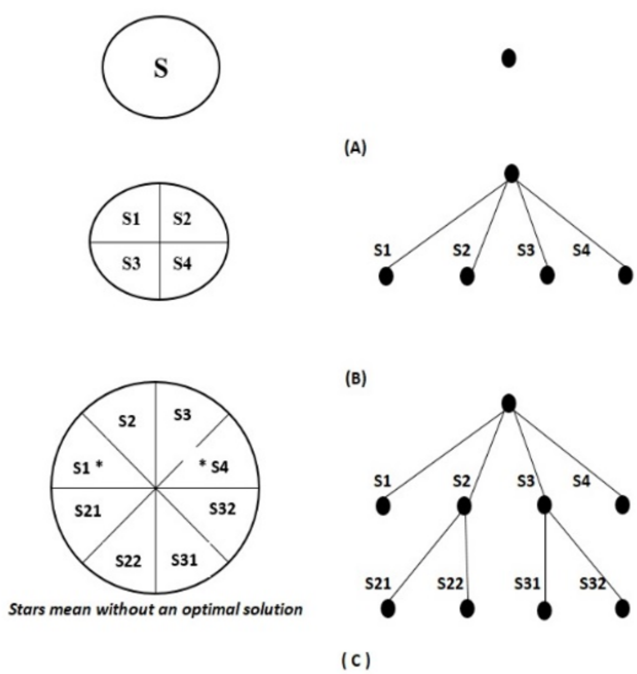

Fig. 2 An illustration of the search space in the branch and .bound algorithm

10 PRACTICAL ASPECT

Mineral resources are among the most important natural resources that support the national economy and promote industrial development. Many provinces, especially in geologically rich countries such as Iraq, contain diverse mineral resources, including gold, phosphate, iron, copper, limestone, gravel, and other economically valuable materials. With advances in mining technology and increasing interest in investing in natural resources, these provinces represent important geographical and economic models that can be used to support sustainable development.

In this study, we focus on five key provinces in the mining sector that contain extractable mineral resources:

Table 1 Distance between the five cities in kilometers

	Baghdad	Muthanna	Najaf	Karbala	Diwaniyah
Baghdad	∞	220	165	160	223
Muthanna	220	∞	98	188	20
Najaf	165	98	∞	89	95
Karbala	160	188	89	∞	179
Diwaniyah	223	20	95	179	∞

Symbols are used to represent the cities in order to simplify the coding process instead of using full city names.

Step 1: A distance matrix is constructed for the five cities, denoted as A, B, C, D, and E. The values marked as (∞) indicate that traveling from a city to itself is not allowed. Using Equations (9) and (10), the reduction process is performed by identifying the two smallest values in each row. The sum of these selected values is then used to calculate the lower bound (LB).

	A	B	C	D	E
A	∞	220	165	160	223
B	220	∞	98	188	20
C	165	98	∞	89	95
D	160	188	89	∞	179
E	223	20	95	179	∞

$A = 325, B = 118, C = 184, D = 249, E = 115$

Initial minimum path length estimate

Step 2: Branching from (A) to (B, C, D, E). One of the branches is closed, and at each step the matrix is modified and (LB) is recalculated.

	A	B	C	D	E
A	∞	220	∞	160	223
B	220	∞	98	188	20
C	∞	98	∞	89	95
D	160	188	89	∞	179
E	223	20	95	179	∞

$A = 380, B = 118, C = 184, D = 249, E = 115$

Excludes

	A	B	C	D	E
A	∞	220	165	∞	223
B	220	∞	98	188	20
C	165	98	∞	89	95
D	∞	188	89	∞	179
E	223	20	95	179	∞

$A = 385, B = 118, C = 184, D = 268, E = 115$

Choose

	A	B	C	D	E
A	∞	220	165	160	∞
B	220	∞	98	188	20
C	165	98	∞	89	95
D	160	188	89	∞	179
E	∞	20	95	179	∞

$A = 325, B = 118, C = 184, D = 249, E = 115$

Excludes

Branches with a smaller lower bound (LB) are considered more promising and are retained for further exploration, as they may contain the optimal solution. In contrast, branches with larger LB values are discarded if their LB exceeds the current best feasible solution, or upper bound (UB).

The final result of the second step gives the minimum expected path after branching, with (LB = 535), which represents the best lower-bound estimate at this stage.

Step 3: Branching is then performed from node D to the possible nodes B, C, and E.

	A	B	C	D	E
A	∞	220	165	∞	223
B	220	∞	98	∞	20
C	165	98	∞	89	95
D	∞	∞	89	∞	179
E	223	20	95	179	∞

$A = 385, B = 118, C = 184, D = 268, E = 115$

Excludes

	A	B	C	D	E
A	∞	220	165	∞	223
B	220	∞	98	188	20
C	165	98	∞	∞	95
D	∞	188	∞	∞	179
E	223	20	95	179	∞

$A = 385, B = 118, C = 193, D = 367, E = 115$

Choose

	A	B	C	D	E
A	∞	220	165	∞	223
B	220	∞	98	188	20
C	165	98	∞	89	95
D	∞	188	89	∞	∞
E	223	20	95	∞	∞

$A = 385, B = 118, C = 184, D = 277, E = 115$

Excludes

The final result of step three: the lowest expected path after branching (LB = 589), which is the best estimate.

Step 4: branching from (C) to (B, E).

	A	B	C	D	E
A	∞	220	165	∞	223
B	220	∞	∞	188	20
C	165	∞	∞	∞	95
D	∞	188	∞	∞	179
E	223	20	95	179	∞

$A = 385, B = 208, C = 260, D = 367, E = 115$

Choose

	A	B	C	D	E
A	∞	220	165	∞	223
B	220	∞	98	188	20
C	165	98	∞	∞	∞
D	∞	188	∞	∞	179
E	223	20	∞	179	∞

A = 385, B = 118, C = 263, D = 367, E = 199

Excludes

The final result of step four: the minimum expected path after branching (LB = 667.5), which is the best estimate.

Step five: branching from (B) to (E).

	A	B	C	D	E
A	∞	220	165	∞	223
B	220	∞	∞	188	∞
C	165	∞	∞	∞	95
D	∞	188	∞	∞	179
E	223	∞	95	179	∞

$A = 385, B = 408, C = 260, D = 367, E = 274$

After all the branching, we determine the path that gives us the highest value (LB) among all the studied branches.

We then continue branching from it in the same way until we reach a path that covers all the points and closes the route by returning to the governorate (A).

Final step: end the journey by returning to (A).

	A	B	C	D	E
A	∞	220	165	∞	∞
B	220	∞	∞	188	∞
C	165	∞	∞	∞	95
D	∞	188	∞	∞	179
E	∞	∞	95	179	∞

$A = 385, B = 408, C = 260, D = 367, E = 274$

We will discuss the results of the same data using the Minimum Inferential Method (CIH).

Step 1: Start with the shortest distance in the first row, representing the initial cycle, leaving the remaining cities.

Step 2: The first insertion for calculating the increments from the current edge is based on Equation (18).

Table 2 shows the addition of the first sub-tour arc based on Equation (1).

Brackets to be replaced	Brackets added to subproofs	Add destination
Input the cities B, C, E on the edge (A, D)		
(A, D)	(A, B, D)	$d(A, B) + d(B, D) - d(A, D)$ $= 220 + 188 - 160 = 248$
(A, D)	(A, C, D)	$d(A, C) + d(C, D) - d(A, D)$ $= 165 + 89 - 160 = 94$
(A, D)	(A, E, D)	$d(A, E) + d(E, D) - d(A, D)$ $= 223 + 179 - 160 = 242$

From Table 2, the smallest additional distance is obtained if:

Based on equation (18), the parentheses were replaced as follows: for inserting city (B), for inserting city (C), and also when inserting city (E). Among these possibilities, the one with the smallest distance can be chosen, represented by the added parentheses, with a value equal to (94).

The new tour after inserting city (C) is $A \to D \to C \to A = 414$

Remaining Cities B, E

Step 3: Current Edges $[(A, D), (A, C), (C, D)]$

Next, a table is created to store the remaining cities that can be inserted into the sub-tour along with their additional distances, as shown in the table (3).

Table 3 Bracket Add Second Sub-tour

Brackets to be replaced	Brackets added to subproofs	Add destination
Input the city (B) on the edge $[(A, D), (A, C), (C, D)]$		
(A, D)	(A, B, D)	$d(A, B) + d(B, D) - d(A, D)$ $= 220 + 188 - 160 = 248$
(A, C)	(A, B, C)	$d(A, B) + d(B, C) - d(A, C)$ $= 220 + 98 - 165 = 153$
(C, D)	(C, B, D)	$d(C, B) + d(B, D) - d(C, D)$ $= 98 + 188 - 89 = 197$
Input the city (E) on the edge $[(A, D), (A, C), (C, D)]$		
(A, D)	(A, E, D)	$d(A, E) + d(E, D) - d(A, D)$ $= 223 + 179 - 160 = 242$
(A, C)	(A, E, C)	$d(A, E) + d(E, C) - d(A, C)$ $= 223 + 95 - 165 = 153$
(C, D)	(C, E, D)	$d(C, E) + d(E, D) - d(C, D)$ $= 95 + 179 - 89 = 185$

From Table 1, the shortest tour is obtained as follows:

When replacing the three edges $[(A, D), (A, C), (C, D)]$ with the edges $[(A, B, D), (A, B, C), (C, B, D)]$, the three edges reproduce themselves in the next parentheses. Similarly, when inserting city (E), we obtain the possible set of alternatives:

$$[(A, E, D), (A, E, C), (C, E, D)]$$

From these possibilities, we can choose the path that has the smallest total distance. The representative path with parentheses is (A, B, C), whose value is (153).

The new cycle after inserting city (B) became:

$$A \to D \to C \to B \to A = 567$$

Remaining cities (E).

Step 4: the current edges are $[(A, D), (A, C), (C, D), (A, B), (B, C)]$ We insert the remaining city (E).

Table 4 Add another city on the edges

Brackets to be replaced	Brackets added to subproofs	Add destination
on the edge $[(A, D), (A, C), (C, D), (A, B), (B, C)]$ (E)Input the city		
(A, D)	(A, E, D)	$d(A, E) + d(E, D) - d(A, D) = 223 + 179 - 160 = 242$
(A, C)	(A, E, C)	$d(A, E) + d(E, C) - d(A, C) = 223 + 95 - 165 = 153$
(C, D)	(C, E, D)	$d(C, E) + d(E, D) - d(C, D) = 95 + 179 - 89 = 185$
(A, B)	(A, E, B)	$d(A, E) + d(E, B) - d(A, B) = 223 + 20 - 220 = 23$
(B, C)	(B, E, C)	$d(B, E) + d(E, C) - d(B, C) = 20 + 95 - 98 = 17$

It is clear from Table 4 above that the smallest increase occurred with the bracket (B, E, C), when the city (E) was entered between (B, C). Therefore, the final cycle became $A \rightarrow D \rightarrow C \rightarrow B \rightarrow E \rightarrow A = 584$

Cheapest Insertion Heuristic (CIH): Produced a route of 584 km by gradually inserting cities according to the minimum additional cost at each step. This method is fast and simple but heuristic in nature, aiming to provide a good solution quickly, though it does not always guarantee optimality.

Branch and Bound (B&B): Relied on calculating lower bounds (LB) for each branch of possible routes and progressively eliminating infeasible ones. This approach is more systematic and ensures the optimal solution by continually narrowing the search space, ultimately yielding the optimal route $(A \rightarrow D \rightarrow C \rightarrow B \rightarrow E \rightarrow A)$.

11 CONCLUSION

CIH is suitable for quickly obtaining approximate solutions in large-scale problems but may not always be ideal.

B&B is more efficient for reaching the optimal solution, though it requires more complex computations and longer processing time as the number of cities increases.

Acknowledgement

The researchers offer their sincere gratitude and thankfulness to the Dept. of Statistics, College of Administration and Economy, Mustansiriyah University, Ministry of Industry and Minerals/Iraq Geological Survey and their esteemed journal for supporting this work.

Funding source

No funds received.

Data availability

N/A

DECLARATIONS

Conflict of interest

The authors declare no competing interests

Consent to publish

N/A

Ethical approval

N/A

REFERENCES

[1] Bavishi T, Shekhar S. Application of Geographic Information System (GIS) in Watershed Management. Plant Archives. 2025;25(Special Issue (ICTPAIRS-JAU)):483-8. ICTPAIRS-JAU Special Issue, January 2025. 10.51470/PLANTARCHIVES.2025.v25.SP.ICTPAIRS-069

[2] Asani EO, Okeyinka AE, Ajagbe SA, Adebiyi AA, Ogundokun RO, Adekunle TS, et al. A Novel Insertion Solution for the Travelling Salesman Problem. Computers, Materials & Continua. 2024;79(1):1581-97. 10.32604/cmc.2024.047898

[3] Alatartsev S, Augustine M, Ortmeier F. Constricting Insertion Heuristic for Traveling Salesman Problem with Neighborhoods. Proceedings of the International Conference on Automated Planning and Scheduling. 2013;23(1):2-10. 10.1609/icaps.v23i1.13539

[4] Chacón J, Irigaray C, Fernández T, El Hamdouni R. Engineering Geology Maps: Landslides and Geographical Information Systems. Bulletin of Engineering Geology and the Environment. 2006;65(4):341-411. 10.1007/s10064-006-0064-z

[5] Curtin KM. Network Analysis in Geographic Information Science: Review, Assessment, and Projections. Cartography and Geographic Information Science. 2007;34(2):103-11. 10.1559/152304007781002163

[6] Demez H. Combinatorial Optimization: Solution Methods of Traveling Salesman Problem [Master's thesis]. Famagusta, North Cyprus: Eastern

Mediterranean University; 2013. Department of Mathematics; supervisor: Arif Akkeleş. Available from: https://hdl.handle.net/11129/654

[7] Davendra D, editor. Traveling Salesman Problem, Theory and Applications. Rijeka, Croatia: InTech; 2010. 10.5772/547

[8] Dawood SM, Elaibi WM, Chachan HA. Solving the Fuzzy Assignment Problems via Utilizing Branch and Bound Algorithm—Practically Treated. Italian Journal of Pure and Applied Mathematics. 2022;48:481-90. Available from: https://ijpam.uniud.it/journal/onl_2022-48.htm

[9] Elaibi WM, Rahi ARK. The Lowest Merit Index in the Branch-and-Bound Algorithm of Structure a Railway Network with Application. In: AIP Conference Proceedings. vol. 2475. AIP Publishing; 2023. p. 080003. 2nd International Conference on Mathematical Techniques and Applications (ICMTA2021). 10.1063/5.0102767

[10] Hu P. Travel the Same Path: A Novel TSP Solving Strategy; 2022. Available from: https://arxiv.org/abs/2210.05906

[11] Işık EE, Şimşir M. Maximizing Total Net Profit for Traveling Salesman Problem with Profits Using Metaheuristic Algorithms. The European Journal of Research and Development. 2023;3(1):46-59. 10.56038/ejrnd.v3i1.228

[12] Kianfar K. Branch-and-Bound Algorithms. In: Cochran JJ, editor. Wiley Encyclopedia of Operations Research and Management Science. John Wiley & Sons; 2011. 10.1002/9780470400531.eorms0116

[13] Mele UJ, Gambardella LM, Montemanni R. A New Constructive Heuristic Driven by Machine Learning for the Traveling Salesman Problem. Algorithms. 2021;14(9):267. 10.3390/a14090267

[14] Papadimitriou CH, Steiglitz K. Combinatorial Optimization: Algorithms and Complexity. Mineola, NY: Courier Corporation; 1998. Unabridged reprint of the 1982 Prentice-Hall edition

[15] Rosenkrantz DJ, Stearns RE, Lewis PM II. An Analysis of Several Heuristics for the Traveling Salesman Problem. SIAM Journal on Computing. 1977;6(3):563-81. 10.1137/0206041

[16] Renaud J, Boctor FF, Ouenniche J. A Heuristic for the Pickup and Delivery Traveling Salesman Problem. Computers & Operations Research. 2000;27(9):905-16. 10.1016/S0305-0548(99)00066-0

[17] Sinaga RP, Marpaung F. Perbandingan Algoritma Cheapest Insertion Heuristic Dan Nearest Neighbor Dalam Menyelesaikan Traveling Salesman Problem. Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam. 2023;2(2):238-47. 10.55606/jurrimipa.v2i2.1614

[18] Sui J, Ding S, Huang X, Yu Y, Liu R, Xia B, et al. A Survey on Deep Learning-Based Algorithms for the Traveling Salesman Problem. Frontiers of Computer Science. 2025;19(6):196322. Published online 12 December 2024. 10.1007/s11704-024-40490-y

[19] Utomo RG, Maylawati DS, Alam CN. Implementasi Algoritma Cheapest Insertion Heuristic (CIH) dalam Penyelesaian Travelling Salesman Problem (TSP). Jurnal Online Informatika. 2018;3(1):61-7. 10.15575/join.v3i1.218

[20] Mohd Yunus MZ, Hassan H. Managing Road Maintenance Using Geographic Information System Application. Journal of Geographic Information System. 2010;2(4):215-8. 10.4236/jgis.2010.24030

How to cite this article

Abdulkareem SR, Elaibi WM. A Comparative Study of Heuristic and Exact Algorithms: The Case of Cheapest Insertion Heuristic and Branch and Bound Algorithm. Journal of University of Anbar for Pure Science. 2026; 20(2):386-394. doi:10.37652/juaps.2025.165439.1711