Turkish Journal of Computer and Mathematics Education Vol.12 No.8 (2021)1226-1229
Research Article

Analysis of Brute Force and Branch & Bound Algorithms to solve the Traveling Salesperson Problem (TSP)

Sriyani Violina [1]

[1]Informatics Department, Engineering Faculty, Widyatama University Jalan Cikutra No 204A Bandung
sriyani.violina@widyatama.ac.id

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Brute Force and Branch & Bound algorithms are two methods commonly used to solve optimization problems. Some examples of problems that can be solved by both Brute Force and Branch & Bound are the Knapsack Problem, Traveling Salesman Problem, Scheduling Problem and many other optimization problems. The solution to the TSP problem with the Brute Force algorithm always reaches the optimal solution, but it takes time and a long stage and will get into trouble if the number of cities increases. The Branch and Bound algorithm can solve the TSP problem more efficiently because it does not calculate all possibilities.

Keywords: Brute Force, Branch & Bound, TSP

1. Introduction

Brute Force and Branch & Bound algorithms are two methods commonly used to solve optimization problems. Many problems can be solved using these two methods. Some examples of problems that can be solved by both Brute Force and Branch & Bound are the Knapsack Problem, Traveling Salesman Problem, Scheduling Problem and many other optimization problems. Brute Force Algorithm is a direct approach to solving a problem, usually directly based on the problem statement and the definition of the concepts involved. The strength of this approach is that it always produces the optimal solution, while the drawback is that it takes a long time to produce the solution The Branch and Bound algorithm is an algorithm that uses a state space tree to solve a problem, in this case it is similar to the backtracking algorithm.

Traveling Salesperson Problem (TSP) is an optimization problem that can be solved with several standard algorithms. Salesperson Traveling Problem is a combinatorial problem where a salesperson has to visit n cities exactly once. To minimize travel time, the shortest route must be determined starting from the salesperson's hometown. This paper will discuss about solving the Traveling Salesperson Problem (TSP) case using the Brute Force and Branch & Bound algorithms.

2. Method

2.1 Travelling Salesperson Problem (TSP)

Traveling salesman problem (TSP) is a problem that is illustrated by the following question: "Given the list of cities and the distance between each two cities, what is the shortest possible route to visit each city exactly once and return to the hometown?" This issue is included in the NP-hard problem. There is also the problem of determining whether a graph has a path that is shorter than a distance L, this problem is called the traveling salesman decision problem (TSDP). This issue is included in the NP-complete problem. This issue has been studied since 1930 and is still an issue that continues to be studied and optimized. Even though it is difficult, many ways have been found to solve this problem, either with a heuristic method or a definite algorithm.

TSP can be described as a graph problem. Each city is represented as a node, while the sides are the distance between cities. Usually the graph used is an undirected graph with each vertex connected to another vertex through one edge. This problem becomes a minimization problem with a point as the starting and ending point, and every other node being visited exactly. TSP which is represented by undirected graph is also called symmetrical TSP. But in reality, the distance traveled is often not the same between A to B and B to A. Not only the distance, but maybe what is different is the time or cost required. Because of these possible inequalities, there is a case where the TSP is represented by a directed graph, also known as asymmetric TSP. This asymmetry can be caused by factors such as congestion, one-way roads, different round-trip fares, etc.

2.2 Algoritma Bruteforce

Brute-force is one of the basic algorithms in the search algorithm. The way brute-force works is to try one by one the existing possibilities until all possibilities have been tried then compare the results obtained and choose the smallest one. In the case of the Traveling Salesman Problem (TSP) using brute-force, the result must be optimal,

but less efficient. Using a brute-force algorithm to solve TSP cases means that before getting optimal results you have to try all the permutations that exist. By using the polynomial approach, we get O (n!) Where n is the number of cities that must be passed. Then a solution like this is impossible, even if the city to pass is only 20 cities.

2.3 Branch and Bound Algorithm

The Branch and Bound (B&B) algorithm is a method or method of systematically searching the solution space. In this case the solution space will be organized into a status space tree. The formation of a status space tree in this algorithm is built with a broad search scheme or Breadth First Search which is abbreviated as BFS. However, the difference is that the expanded node is not based on the order of generation, but the node that has the smallest "cost" value among other lifecycle nodes. (Bonal et al., 2019)

Each node has a "cost" value, which is the estimated value or the estimated path of the smallest cost from the node to the destination node (goal node), and can be expressed in general terms as follows:

$$\hat{c}(i) = \hat{f}(i) + \hat{g}(i)$$

So that,

ĉ(i) = value for node i

f(i) = the value reaches vertex i from the root

ĝ(i) = value reaches the destination node of the node.

The way the Branch and Bound algorithm works is based on the following two principles.

1. Recursively divide the status space into smaller spaces and minimize "costs" on these spaces. This process is called branching.
2. Branching will be equivalent to brute-force enumeration. To improve performance, bound is used to limit the space in the status that is generated, eliminating candidate solutions that are proven to not contain optimal solutions.

As the name implies, this algorithm has a bound or limiting function. This constraint function is useful for delimiting paths that are considered not leading to a solution node.

2.1 Testing Scenario

Testing of these two algorithms will be carried out in one case example, namely a salesperson must visit 5 cities, each of which has a certain distance. The salesperson departs from city A and has to visit all cities exactly once and return to the hometown. The distance from each city can be seen in Figure 1.

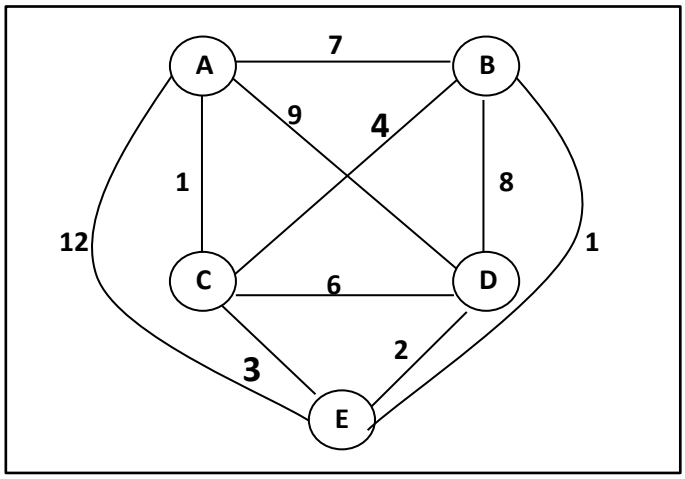


Fig 1 The case of 5 Cities Travelling Salesperson Problem

Data from these cases can be seen in table 1:

Tabel 1

	A	B	C	D	E
A	∞	7	11	9	12
B	7	∞	4	8	10
C	11	4	∞	6	3
D	9	8	6	∞	2
E	12	10	3	2	∞

3. Result and Analysis

3.1 Brute Force

The solution to the Traveling Salesman Problem (TSP) using Branch and bound is done by calculating all the possibilities of the existing path and calculating the distance from each of these possibilities. For n cities, the probability is O (n!), If 5 cities then there are 60 possibilities. An example of a calculation is as follows:

Tabel 2
Alternative Solution using Brute Force

NO	PATH	DISTANCE
1.	A-B-C-D-E-A	7+4+6+2+12=31
2.	A-B-C-E-D-A	7+4+3+2+9=25
3.	A-B-D-C-E-A	7+8+6+3+12=36
4.	A-B-D-E-C-A	7+8+2+3+11=31
	.	.
.		.
.	.	.
60.	A-E-D-C-B-A	12+2+6+4+7=31

Then from that alternative a path with a minimum distance is selected.

3.2 Branch and Bound

The solution to the Traveling Salesman Problem (TSP) using Branch and Bound is done by determining the lower limit by adding the minimum distance from each row in table 1.

So that

LB = 7 + 4 + 3 + 2 + 2 = 18

Choose from city A to city B, because each city is only visited once, so everything going to city B cannot be selected, from B to A also cannot be selected because city A will be visited when returning to the initial city,

Table 3

	A	B	C	D	E
A	∞	7	11	9	12
B	~~7~~	∞	4	8	10
C	11	4	∞	6	3
D	9	8	6	∞	2
E	12	10	3	2	∞

then select the minimum value of each line available.

AB = 7 + 4 + 3 + 2 + 2 = 18

Do the same for the other three possibilities:

AC = 11 + 7 + 3 + 2 + 2 = 25

AD = 9 + 4 + 3 + 2 + 3 = 21

AE = 12 + 4 + 4 + 6 + 2 = 28

Then formed a tree like Figure 2

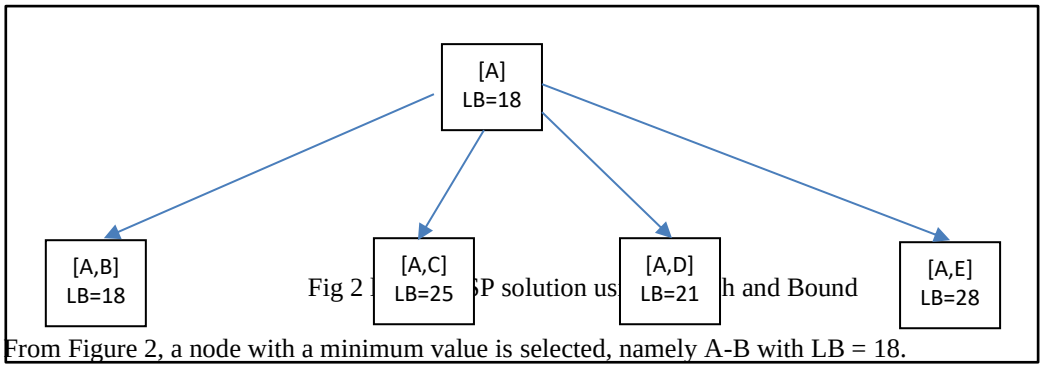
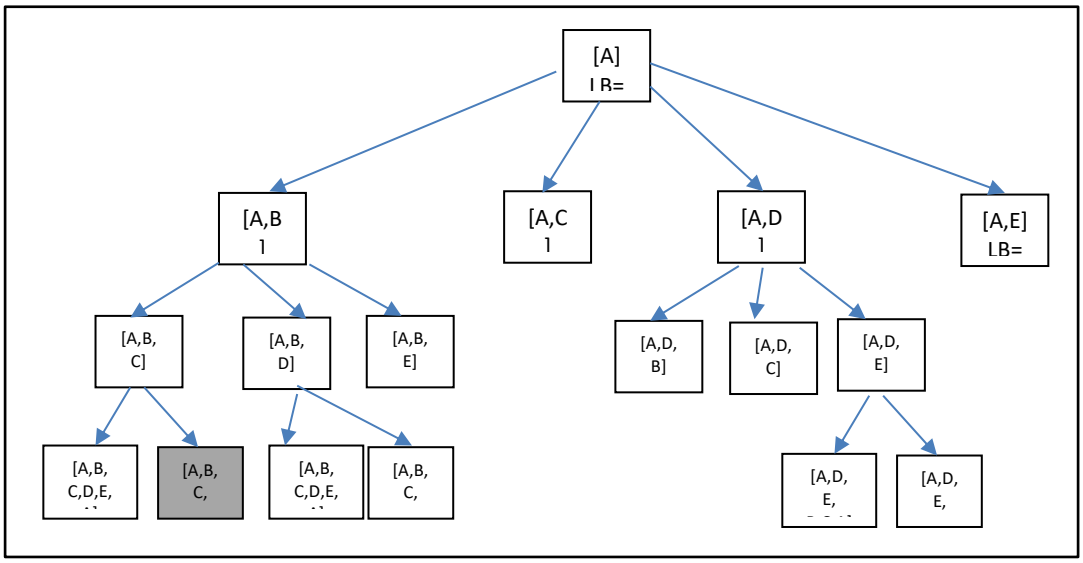


Fig 2 [illegible]SP solution us[illegible]h and Bound

From Figure 2, a node with a minimum value is selected, namely A-B with LB = 18.

Fig 3 Solving TSP using Branch and Bound



3.3 Analysis

In the case of the Traveling Salesman Problem (TSP), the results of using the brute-force optimization approach resulted in an optimal solution but it was time consuming and ineffective. The use of branch and bound results in a shorter solution because the branch and bound algorithm performs calculations recursively, while still calculating the best value, that is what is known as branching. In addition, the best value is recorded for each calculation, so that it can improve the performance of this algorithm, that is what is known as bounding.

4. Conclusion

The solution to the TSP with the Brute Force algorithm always reaches the optimal solution, but it takes time and a long stage and will get into trouble if the number of cities increases. The Branch and Bound algorithm can solve the TSP problem more efficiently because it does not calculate all possibilities.

References

1.Neapolitan, Richard, "Foundations of Algorithms", Jones & Bartlett Publishers, 2014
2. Steven S. Skiena, The Algorithm Design Manual, Springer, 1997
3. Wang, W., Tian, G., Zhang, T., Jabarullah, N. H., Li, F., Fathollahi-Fard, A. M., ... & Li, Z. (2021). Scheme selection of design for disassembly (DFD) based on sustainability: A novel hybrid of interval 2-tuple linguistic intuitionistic fuzzy numbers and regret theory. *Journal of Cleaner Production*, 281, 124724.
4. Bonal, J. R., Lorenzo Calvo, A., & Jiménez Saiz, S. L. (2019). Key Factors on Talent Development of Expertise Basketball Players in China. Revista de psicología del deporte, 28(3), 0009-16.