Course: Advanced Algorithm and Problem Solving

WEEK 5 - Backtracking and Branch-and-Bound

Lemlem Kassa (Ph.D.)

Addis Ababa Science and Technology University, Ethiopia

April, 2025

Week 5: Backtracking and Branch-and-Bound

Contents

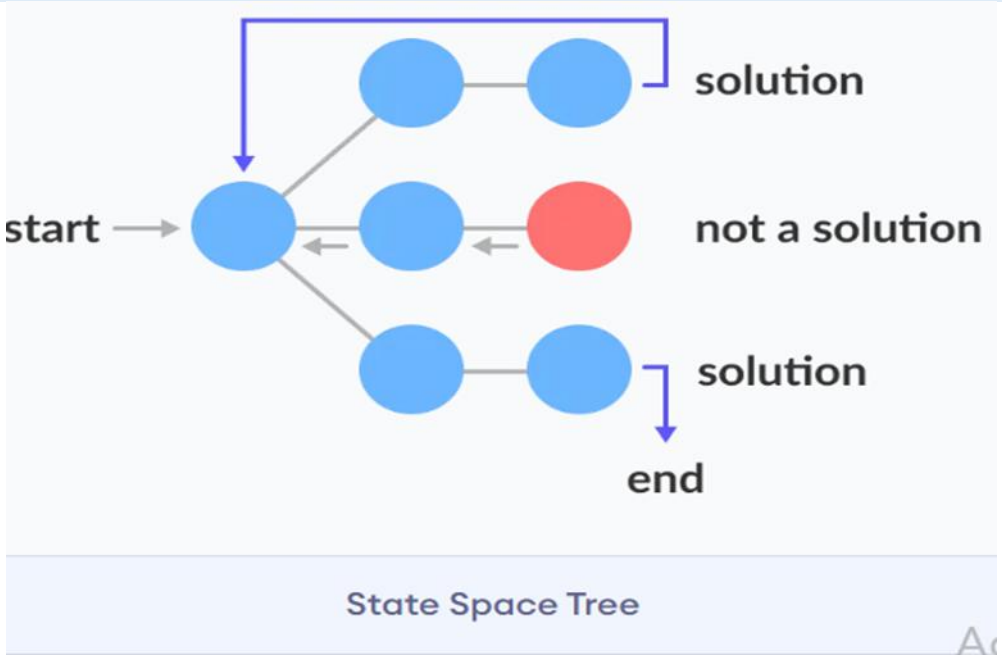
- Introduction to Backtracking Algorithm
- Examples of Backtracking Algorithms
- Advantages and disadvantages of Backtracking Algorithms
- Applications of Backtracking Algorithm
- Branch and bound algorithm
- Backtracking Vs Branch and Bound Techniques

Lecture Learning Outcome

- Understand backtracking algorithms
- Understand the components of backtracking algorithm
- Understand how backtracking algorithms work
- Understand searching techniques in branch and bound
- How does branch and bound algorithm work?
- Identify the advantages and disadvantages of backtracking algorithms
- Understand the difference between Backtracking and Branch and Bound Technique

Introduction to Backtracking Algorithm

- A backtracking algorithm :- a problem-solving algorithm that uses a brute force approach for finding the desired output.
- The Brute force approach tries out all the possible solutions and chooses the desired/best solutions.



State Space Tree

State Space Tree

- a tree representing all the possible states (solution or nonsolution) of the problem from the root as an initial state to the leaf as a terminal state.

[1]. Programiz, Backtracking Algorithm, https://www.programiz.com/dsa/backtracking-algorithm

Introduction to Backtracking Algorithm ….(cont'd)

- A backtracking algorithm is a way to solve problems by trying out different options one by one, and if an option doesn't work, it "backtracks" and tries the next option.
- It's like exploring a maze: we try one path, and if we hit a dead end, we go back and try a different path until we find the exit.
- The goal is to explore all possible paths until we find the correct solution.

Backtracking Algorithm

```
Backtrack (x)
   if x is not a solution
      return false
   if x is a new solution
      add to list of solutions
   backtrack(expand x)
```

[2]. WsCube Tech, Backtracking Algorithm,https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Introduction to Backtracking Algorithm ….(cont'd)

Components of Backtracking Algorithm

1. Decision Tree

- A backtracking algorithm explores a problem by building a decision tree.
- Each node in the tree represents a decision or choice that can be made, and each branch represents the consequences of that choice.

2. Recursive Approach

- Backtracking uses recursion to explore the decision tree.
- At each step, the algorithm makes a choice and then recursively tries to solve the smaller problem that results from that choice.

3. Backtracking Step

- If the current path or choice doesn't lead to a valid or optimal solution, the algorithm backtracks by undoing the last decision and tries a different path.
- This step ensures that all possible options are explored until the solution is found

How Backtracking Algorithms Work?

1. Start at the Initial Position

- The algorithm begins at the initial position or the root of the decision tree. This is the starting point from where different paths will be explored.

2. Make a Decision

- At each step, the algorithm makes a decision that moves it forward. This could be moving in a certain direction in a maze or choosing a particular option in a decision tree.

3. Check for Validity

- After making a decision, the algorithm checks if the current path is valid or if it meets the problem's constraints.
- If the path is invalid or leads to a dead end, the algorithm backtracks to the previous step.

[2]. WsCube Tech, Backtracking Algorithm, https://www.wscubetech.com/resources/dsa/backtracking-algorithm

How Backtracking Algorithms Work? …(cont'd)

4. Backtrack if Necessary

- If a dead end is reached or if the path doesn't lead to a solution, the algorithm backtracks by undoing the last decision.
- It then tries a different option from the previous decision point.

5. Continue Exploring

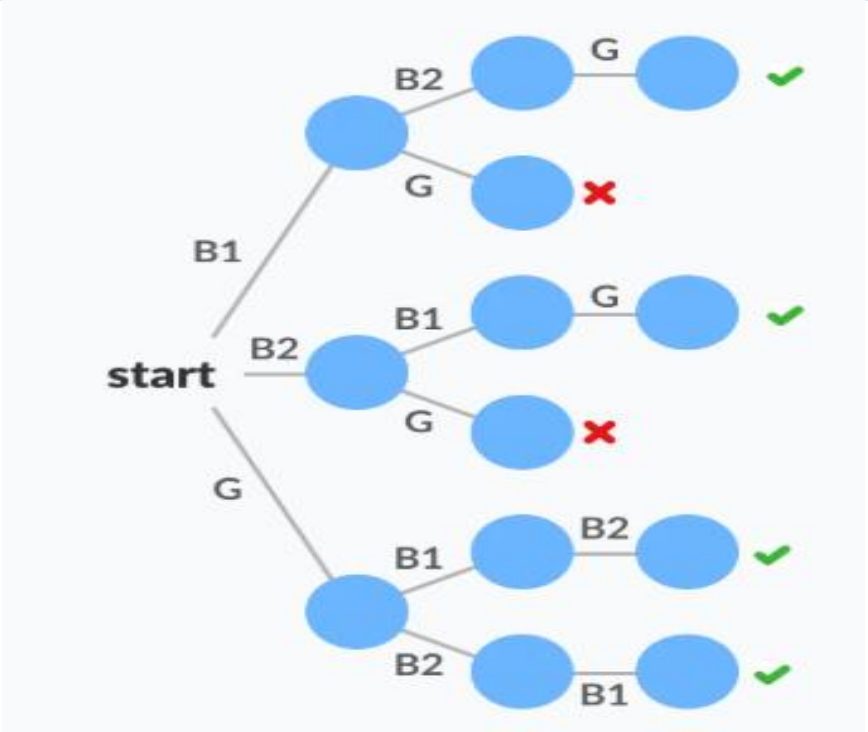
- The algorithm continues to explore different paths, making decisions, checking validity, and backtracking when necessary.
- This process repeats until a solution is found or all possible paths have been explored.

6. Find the Solution or Exhaust All Options

- The algorithm stops when it finds a valid solution or when all possible paths have been explored and no solution exists.

Examples of Backtracking Algorithms

- **Problem**: Find all the possible ways of arranging 2 boys and 1 girl on 3 benches.
  - **Constraint:** Girl should not be on the middle bench.
- **Solution:** There are a total of 3! = 6 possibilities. We will try all the possibilities and get the possible solutions. We recursively try all the possibilities.
- The following **state space tree** shows the possible solutions.

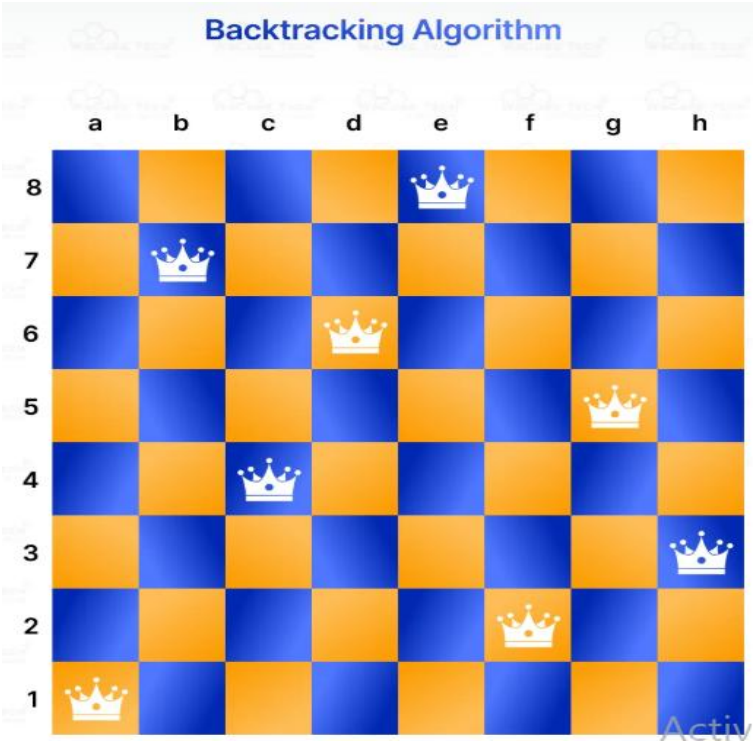


[1]. Programiz, Backtracking Algorithm, https://www.programiz.com/dsa/backtracking-algorithm

Examples of Backtracking Algorithms …(cont'd)

N-Queens Problem

- The N queen problem using backtracking algorithm involves placing N queens on an N×N chessboard such that no two queens threaten each other.
- This means no two queens can share the same row, column, or diagonal.
- The backtracking algorithm places queens one by one in different rows, checking for conflicts, and backtracking when a conflict is found until all queens are safely placed.



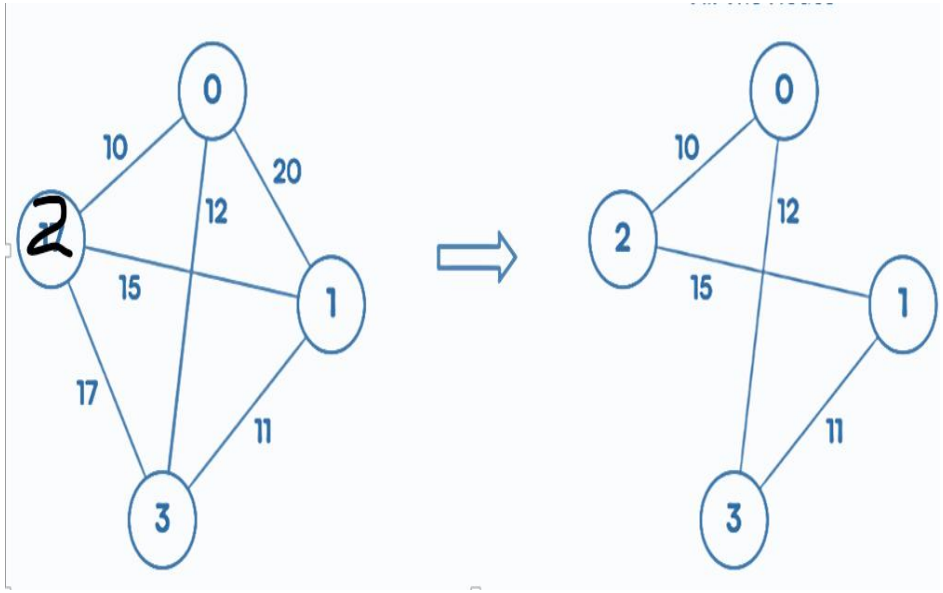
[2]. WsCube Tech, Backtracking Algorithm,https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Examples of Backtracking Algorithms …(cont'd)

Travelling Salesperson Problem

- It is an optimization problem where a salesperson must visit a given set of cities exactly once, starting and ending at the same city.
- The goal is to find the shortest possible route that covers all the cities and returns to the starting point.
- TSP is an NP-hard problem, meaning there is no known efficient solution for large datasets, but various algorithms can provide exact or approximate solutions.

Travelling Salesman Problem



[2]. WsCube Tech, Backtracking Algorithm, https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Examples of Backtracking Algorithms …(cont'd)

- The algorithm for salesman Travelling problem is important because it represents real-world problems that many industries face, such as:
    - Delivery services need to plan routes to drop off packages at multiple locations.
    - Logistics companies need to find the shortest path to transport goods efficiently.
    - Manufacturing companies need to reduce the travel of robotic arms in factories.
- Solving TSP helps reduce time, fuel costs, and energy, making operations faster and cheaper.

[2]. WsCube Tech, Backtracking Algorithm, https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Examples of Backtracking Algorithms …(cont'd)

- Let's say we have a salesperson who needs to visit four popular Indian cities: Mumbai, Delhi, Bengaluru, and Chennai, and they want to find the shortest route that visits each city exactly once and returns to the starting city.

The distances between the cities are:

	Mumbai	Delhi	Bengaluru	Chennai
Mumbai	0	1,400 km	980 km	1,330 km
Delhi	1,400 km	0	2,150 km	2,200 km
Bengaluru	980 km	2,150 km	0	350 km
Chennai	1,330 km	2,200 km	350 km	0

Problem: The salesperson starts in Mumbai and must visit Delhi, Bengaluru, and Chennai exactly once, then return to Mumbai. The goal is to find the shortest route.

Examples of Backtracking Algorithms …(cont'd)

Possible Routes: Let's calculate the total distance for a few possible routes:

1. Mumbai → Delhi → Bengaluru → Chennai → Mumbai

- Mumbai → Delhi = 1,400 km
- Delhi → Bengaluru = 2,150 km
- Bengaluru → Chennai = 350 km
- Chennai → Mumbai = 1,330 km

Total distance = 1,400 + 2,150 + 350 + 1,330 = 5,230 km

2. Mumbai → Delhi → Chennai → Bengaluru → Mumbai

- Mumbai → Delhi = 1,400 km
- Delhi → Chennai = 2,200 km
- Chennai → Bengaluru = 350 km
- Bengaluru → Mumbai = 980 km

Total distance = 1,400 + 2,200 + 350 + 980 = 4,930 km

Examples of Backtracking Algorithms …(cont'd)

3. Mumbai → Bengaluru → Chennai → Delhi → Mumbai

- Mumbai → Bengaluru = 980 km
- Bengaluru → Chennai = 350 km
- Chennai → Delhi = 2,200 km
- Delhi → Mumbai = 1,400 km

Total distance = 980 + 350 + 2,200 + 1,400 = 4,930 km

[2]. WsCube Tech, Backtracking Algorithm,https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Examples of Backtracking Algorithms …(cont'd)

The shortest routes are:

- Mumbai → Delhi → Chennai → Bengaluru → Mumbai with a total distance of *4,930 km.*
- Mumbai → Bengaluru → Chennai → Delhi → Mumbai with a total distance of *4,930 km.*
- In both cases, the total travel distance is the same, and this is the most efficient route for the salesperson to minimize travel distance.

Key Takeaway:

- In this example, we used the salesman travelling problem to determine the shortest route between four major Indian cities.
- By calculating different possible routes, we found the shortest one, which helps reduce travel time and costs

[2]. WsCube Tech, Travelling Salesman Problem (TSP), https://www.wscubetech.com/resources/dsa/travelling-salesman-problem

Advantages of Backtracking Algorithms

- **Simple and Intuitive:** Easy to understand and implement for a wide range of problems.
- **Flexibility:** Can be applied to many types of problems, especially combinatorial and constraint satisfaction problems.
- **Exhaustive Search:** Guarantees finding a solution if one exists by exploring all possible options.
- **Effective Pruning:** Can reduce the search space significantly by eliminating paths that cannot lead to a solution.
- **Handles Complex Problems:** Suitable for problems with complex constraints that other algorithms may struggle with.

Disadvantages of Backtracking Algorithms

- **High Time Complexity:** Can be very slow due to the need to explore all possible solutions, leading to exponential time complexity in many cases.
- **Memory Intensive:** Uses a lot of memory, especially for deep recursion and large decision trees.
- **Inefficient for Large Problems:** Not practical for large-scale problems due to the combinatorial explosion of possibilities.
- **Non-Optimal Solutions:** While it finds a solution, it may not be the most efficient or optimal one without further optimization.
- **Requires Careful Pruning:** Effective use of backtracking depends on good pruning techniques, which can be complex to implement correctly.

[2]. WsCube Tech, Backtracking Algorithm, https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Applications of Backtracking Algorithm

- **Puzzle Solving:** Used in solving puzzles like Sudoku, crosswords, and the N-Queens problem.
- **Combinatorial Optimization:** Generates all permutations, combinations, and subsets of a given set.
- **Constraint Satisfaction Problems:** Applied in problems like graph coloring, scheduling, and job assignment where constraints must be met.
- **Pathfinding:** Solves networks and other pathfinding problems

[2]. WsCube Tech, Backtracking Algorithm, https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Applications of Backtracking Algorithm

- **Game Solving:** Used in strategy games to explore possible moves, such as in the Knight's Tour problem.
- **Decision-Making:** Helps in making optimal decisions in situations where multiple choices are possible, like in the Subset Sum problem.
- **String Processing:** Generates valid strings that meet certain criteria, such as generating balanced parentheses.
- **Optimization Problems**: Finds solutions to complex optimization problems, such as maximizing profits or minimizing costs under certain constraints.

[2]. WsCube Tech, Backtracking Algorithm, https://www.wscubetech.com/resources/dsa/backtracking-algorithm

Branch and bound algorithm

- The branch and bound algorithm is a technique used for solving combinatorial optimization problems.
- First, this algorithm breaks the given problem into multiple sub-problems and then using a bounding function, it eliminates only those solutions that cannot provide optimal solution.
- Combinatorial optimization problems refer to those problems that involve finding the best solution from a finite set of possible solutions, such as the 0/1 knapsack problem, the travelling salesman problem and many more.

[3]. Tutorials Point, Branch and Bound Algorithm
https://www.tutorialspoint.com/data_structures_algorithms/dsa_branch_and_bound_algorithm.htm

Branch and bound algorithm …(cont'd)

When Branch and Bound Algorithm is used?

- Whenever we encounter an optimization problem whose variables belong to a discrete set. These types of problems are known as discrete optimization problems.
- This algorithm is also used to solve combinatorial optimization problem.
- If the given problem is a mathematical optimization problem, then the branch and bound algorithm can also be applied.

[3]. Tutorials Point, Branch and Bound Algorithm
https://www.tutorialspoint.com/data_structures_algorithms/dsa_branch_and_bound_algorithm.htm

Branch and bound algorithm …(cont'd)

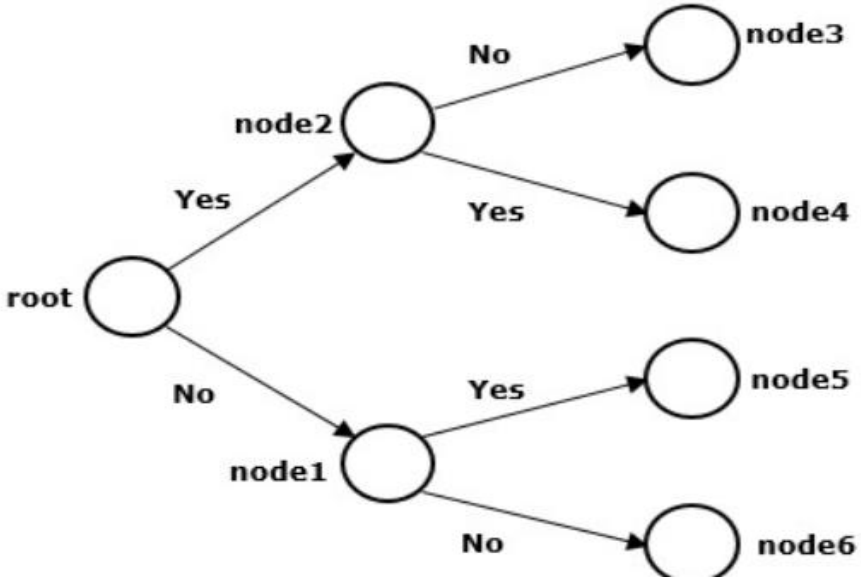
How does Branch and Bound Algorithm work?

- The branch and bound algorithm works by exploring the search space of the problem in a systematic way.
- It uses a tree structure (state space tree) to represent the solutions and their extensions.
- Each node in the tree is part of the partial solution, and each edge corresponds to an extension of this solution by adding or removing an element.
- The root node represents the empty solution.

[3]. Tutorials Point, Branch and Bound Algorithm
https://www.tutorialspoint.com/data_structures_algorithms/dsa_branch_and_bound_algorithm.htm

Branch and bound algorithm …(cont'd)

- The algorithm starts with the **root node** and moves towards its **children nodes**.
- At each level, it evaluates whether a child node satisfies the constraints of the problem to achieve a feasible solution.
- This process is repeated until a **leaf node** is reached, which represents a complete solution.



[3]. Tutorials Point, Branch and Bound Algorithm
https://www.tutorialspoint.com/data_structures_algorithms/dsa_branch_and_bound_algorithm.htm

Branch and bound algorithm

The solution of the Branch and the bound problem can be represented in two ways:

- **Variable size solution:** Using this solution, we can find the subset of the given set that gives the optimized solution to the given problem.
  - E.g., if we have to select a combination of elements from {A, B, C, D} that optimizes the given problem, and it is found that A and B together give the best solution, then the solution will be {A, B}.

- **Fixed-size solution: F**or the above example, the solution will be given by {1, 1, 0, 0}, here 1 represent that we have select the element which at ith position and 0 represent we don't select the element at ith position.

[3].GeeksforGeeks, Introduction to Branch and Bound, May 8, 2023, https://www.geeksforgeeks.org/introduction-to-branch-and-bound-data-structures-and-algorithms-tutorial/

Branch and bound algorithm …(cont'd)

Searching techniques in Branch and Bound

- There are different approaches to implementing the branch and bound algorithm.
- The implementation depends on how to generate the children nodes and how to search the next node to expand.
- **Breadth-first search** − It maintains a queue of nodes to expand, which means this searching technique Least cost search − This searching technique works by computing bound value of each node. The algorithm selects the node with the lowest bound value to expand next.
- **Depth-first search** − It maintains a stack of nodes to expand, which means this searching technique uses Last in First out order to search the next node. Uses FIFO order to search next node.

Branch and bound algorithm …(cont'd)

Advantages of Branch and Bound Algorithm

- It can reduce the time complexity by avoiding unnecessary exploration of the state space tree.
- It has different search techniques that can be used for different types of problems and preferences.

Disadvantages of Branch and Bound Algorithm

- In the worst case scenario, it may search for all the combinations to produce solutions.
- It can time consuming if the state space tree is too large

[4]. Tutorials Point, Branch and Bound Algorithm
https://www.tutorialspoint.com/data_structures_algorithms/dsa_branch_and_bound_algorithm.htm

Backtracking Vs Branch and Bound Technique

Parameters	Backtracking	Branch-N-Bound
Why it is used?	This is used to solve the decision-based problem.	This is used to solve the optimization problem.
Nodes	This is a state space tree where the node explored the depth-first search.	This explored the optimization problem.
Efficient	More efficient.	Less Efficient.
Function	It involves the feasibility function.	It involves the bounding function.
Traverse	It traverses the tree by depth-first search	It traverses the tree by breadth-first search
Solves	Backtracking can solve the game of chess and sudoku.	It doesn't solve any game problem.
Application Used	This application is used to solve the N-queen problem, the Hamilton cycle, and the problem based on graph coloring.	This type of application is used to solve the problem based on the Travelling salesman problem.

[5]. TutorialsPoint, Difference Between Backtracking and Branch-N-Bound Technique, Tapas Kumar Ghosh, May 10, 2023. https://www.tutorialspoint.com/difference-between-backtracking-and-branch-n-bound-technique

Summary

- A backtracking algorithm is a way to solve problems by trying out different options one by one, and if an option doesn’t work, it "backtracks" and tries the next option.
- A backtracking algorithm explores a problem by building a decision tree, recursive approach and backtracking step.
- The branch and bound algorithm is a technique used for solving combinatorial optimization problems. It first breaks the given problem into multiple sub-problems and then using a bounding function, it eliminates only those solutions that cannot provide optimal solution.
- Breadth-first search and Depth-first search are the common searching using Branch and Bound techniques

References

1. Programiz, Backtracking Algorithm, https://www.programiz.com/dsa/backtracking-algorithm

2. WsCube Tech, Backtracking Algorithm, https://www.wscubetech.com/resources/dsa/backtracking-algorithm

3. GeeksforGeeks, Introduction to Branch and Bound, May 8, 2023, https://www.geeksforgeeks.org/introduction-to-branch-and-bound-data-structures-and-algorithms-tutorial/

4. Tutorials Point, Branch and Bound Algorithm, https://www.tutorialspoint.com/data_structures_algorithms/dsa_branch_and_bound_algorithm.ht

5. TutorialsPoint, Difference Between Backtracking and Branch-N-Bound Technique, Tapas Kumar Ghosh, May 10, 2023. https://www.tutorialspoint.com/difference-between-backtracking-and-branch-n-bound-technique

Thank You!

For your attention