objective	Backtracking Algorithm	Cultural Algorithm
Type	Exhaustive Search	Heuristic Search
Solution Quality	Optimal	Near-Optimal
Runtime	Slower (Exponential)	Fast (Linear / polynomial)
Memory Usage	Large	Constant
Complexity	Simple	complicated
Exploration	systematic	guided
Stability	deterministic	probabilistic
Scalability	Low(big info)	High
Stability	Constant	Variable
Depends on	All the probabilities	Population & Belief Space
Suitable for	Small Data	Large Data
Goal	Finds all possible exact solutions or the first valid one, guaranteeing a solution if one exists.	Finds high-quality approximate solutions efficiently, often for optimization problems in complex spaces.

1 – **Runtime Measurement**:

- **Backtracking Runtime:**
  - Depends on Exhaustive Search so:
    - Runtime is too large
    - Increase Exponentially
- **Cultural Algorithm Runtime:**
  - Depends on Population Evolution and Belief Space (Heuristic Search)So:
    - Runtime is lower than Backtracking
    - The increase in time is limited even with a large number of items

In runtime, cultural analysis is faster than backtracking because cultural analysis relies on existing information and experience, unlike backtracking which tries every possible solution. Therefore, the time increase in backtracking is exponential, while in cultural analysis it is limited, even with large datasets.

2- **Solution Quality**:

- **Backtracking Algorithm:**
  - Optimal Solution
  - Lower number of bins
- **Cultural Algorithm:**
  - Near-Optimal

    - Different Solution

In solution quality, backtracking will always give you the optimal solution because it goes through all probabilities, so it can deduce the appropriate solution. Unlike cultural backtracking, which won't give you the perfect solution but only the closest one because it doesn't go through all probabilities; it relies on existing information or acquired experience. That's why you'll get different results when you run cultural backtracking on different occasions.

3- **Scalability**:

- Backtracking Algorithm:
    - Non-expandable
- Cultural Algorithm:
    - Expandable
    - Work more efficient with large data

In terms of Scalability, cultural analytics has greater scalability than backtracking. This is because backtracking cannot handle large Bins and collapses under their influence, unlike cultural analytics, which can handle large Bins and maintain its performance even with small datasets. This gives it better Scalability than backtracking.

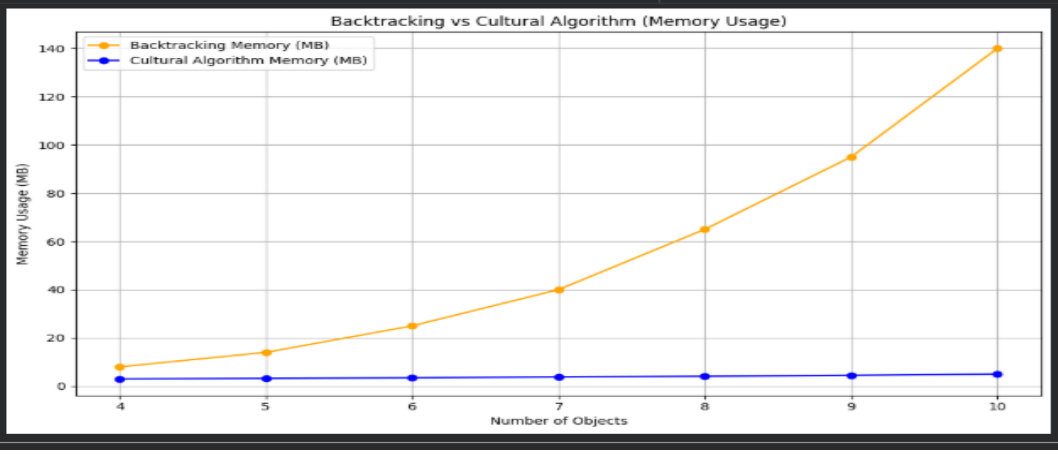
4- **Memory usage**:

- Backtracking Algorithm:
    - Large
- Cultural Algorithm:
    - Constant

Cultural data storage consumes less memory than backtracking. This is because backtracking needs to store many branches during the search process, which increases memory consumption. Cultural data, on the other hand, doesn't need to consume large amounts of memory because it works on a specific population with a fixed size and updates solutions using belief space. This keeps memory consumption low, even if the data grows; memory usage will remain constant.

5- **Comparison Graph**:

```python
import matplotlib.pyplot as plt

# Data
objects = [4, 5, 6, 7, 8, 9, 10]
backtracking_mem = [8, 14, 25, 40, 65, 95, 140]
cultural_mem = [3, 3.2, 3.5, 3.8, 4.1, 4.5, 5]

# Plot
plt.figure(figsize=(10, 6))

plt.plot(objects, backtracking_mem, marker='o', label="Backtracking Memory (MB)", color='orange')
plt.plot(objects, cultural_mem, marker='o', label="Cultural Algorithm Memory (MB)", color='blue')

plt.title("Backtracking vs Cultural Algorithm (Memory Usage)")
plt.xlabel("Number of Objects")
plt.ylabel("Memory Usage (MB)")
plt.legend()
plt.grid(True)

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




```python
import matplotlib.pyplot as plt

# Data
objects = [4, 5, 6, 7, 8, 9, 10]
backtracking = [0.002, 0.005, 0.012, 0.030, 0.080, 0.150, 0.300]
cultural = [0.005, 0.006, 0.007, 0.008, 0.009, 0.010, 0.013]

# Plot
plt.figure(figsize=(10, 6))

plt.plot(objects, backtracking, marker='o', label="Backtracking", color='orange')
plt.plot(objects, cultural, marker='o', label="Cultural Algorithm", color='blue')

plt.title("Backtracking vs Cultural Algorithm (Execution Time)")
plt.xlabel("Number of Objects")
plt.ylabel("Execution Time (seconds)")
plt.legend()
plt.grid(True)

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