

Algorithms Homework 3

Backtracking & Branch-and-Bound

Due data: May 29, 2025 23:59

Many problems don't have an efficient algorithm which runs in polynomial time, especially those NP-complete problems, such as *0-1 Knapsack problem* and *Traveling salesmen problem (TSP)*. With the helps of finding a good bound and appropriate *promising function* (bound function), we can determine a state node is worthy to continue explore or not.

In this problem, you are asked to solve either 0-1 knapsack problem or TSP (**You only need to choose one problem to implement.**) by implementing:

1. the depth-first search order
2. the bread-first search order
3. the best-first search order

In your implementation, you are asking to compare the number of state space tree nodes visiting by different algorithms, and compares this value with the total number of nodes in the tree. Experiments can based on:

- different number of items in 0-1 knapsack (e.g. 10, 15, 20, ...)
- different weight range for 0-1 knapsack(e.g. [1-1], [1-10], [1-10]×10, [1-1000], ...)
- different number of cities for TSP (e.g. 6, 8, 10, 12, ...)
- different distance between each cities for TSP (e.g. [1-10], [1-100])

You need to submit a zip/rar file with

- A folder with name *code* includes all of your codes (.cpp, .h, and etc.)
 - Including all the versions you implemented.
- A pdf file with name report.pdf.

In your report, at least you need to explain

- The contains for each of your source code (what you implement)
- The different input setting you chose to test your programs.
- The experiment results of the different input.

You may also

- Try to find a better bound function.
- Try to find better algorithms for special case.
- Try different things may affect the performance.