SIENA
COMPUTER SCIENCE

Computer Science 385
Design and Analysis of Algorithms
Siena University
Spring 2026

Topic Notes: Exhaustive Search

An *exhaustive search* is a brute force solution to a problem involving search for an element with a special property, usually among combinatorial objects such as permutations, combinations, or subsets of a set.

The typical approach involves:

1. Generation of a list of all potential solutions to the problem in a systematic manner.
2. Evaluation of the potential solutions one by one, disqualifying infeasible ones and, for an optimization problem, keeping track of the best one found so far.
3. Reporting the solution when the search ends.

We will look briefly at a few examples of problems that can be solved with exhaustive search.

Traveling Salesman

The *traveling salesman problem (TSP)* asks us, given n cities with known distances between each pair, find the shortest tour that passes through all the cities exactly once before returning to the starting city.

Solutions to this kind of problem have important practical applications (delivery services, census takers, etc.)

A related problem is to find the shortest *Hamiltonian circuit* in a weighted, connected graph. A Hamiltonian circuit is a cycle that passes through all of the vertices of the graph exactly once.

The text has a small example in Fig. 3.7, p. 117, and we will work through a similar one in class as part of the handout.

We will also look at the brute-force TSP visualization in METAL (see the link on the lecture page).

Knapsack Problem

Another well-known problem is the *knapsack problem*.

Given n items with weights $w_1, w_2, ..., w_n$ and values $v_1, v_2, ..., v_n$, what is the most vaulable subset of items that can fit into a knapsack that can hold a total weight W.

The exhaustive search here involves considering all possible subsets of items and finding the subset whose total weight is no more than W with the highest value.

We will work through an example on the handout.

Assignment Problem

Our final exhaustive search example is the *assignment problem*. It may be stated as the assignment of n jobs among n people, one job per person, where the cost of assigning person i to job i is given by $C[i, j]$, and we wish to minimize the total cost across all possible assignments.

Our exhaustive search here involves the generation of all possible assignments, computing the total cost of each, and selection of the lowest cost.

We will work through an example on the handout.

Exhaustive Search Wrapup

- Exhaustive-search algorithms run in a realistic amount of time only on very small instances.
- Sometimes, much better approaches exist (and we will see some of those soon).
- But sometimes, an exhaustive search is the only known way to be guaranteed an exact solution.