Introduction to Combinatorial Optimization

CS 149 Staff

September 8, 2009

Combinatorial optimization is useful for solving many problems outside of computer science. For example, consider a museum robber who would like to steal the crown jewels. Unfortunately, he forgot to bring a Mack truck, so has to pick and choose. One natural heuristic is to be greedy, placing the item with the highest value per pound (that will fit) in one's knapsack until no more items will fit. This is good enough for your average burglar, but the Mafia won't settle for suboptimality. Later on in the course we will discuss how to systematically search through the possibilities, thereby proving a solution to the optimal. This is a lot less exhausting if one can infer that any solution better than the best currently known must include or exclude certain items, without having to explicitly enumerate all the possibilities. This is inference.

Optimal burglary may not be very important, but this problem does show up all the time in the real world. Airplanes need to be loaded with valuable cargo without being overloaded. When buying 7 gallons of paint, you need to decide whether two 3-gallon cans and one 1-gallon can is cheaper than one 5-gallon can and a 3-gallon can. The choice of which cans you *don't* buy is a knapsack problem.

Suppose you're running a college. You need to decide how to allocate the various departments among the buildings. Each building has a certain amount of office space, classroom space, and lab space. Each department has a known demand for the different types of space. Can you assign the departments to the buildings so that no building runs out of a type of space? People would be unhappy if a department were split between several buildings. This problem is called the market split problem. The standard motivation for this problem is a little contrived. Buildings, types of space, and apartments are replaced with divisions of the company, goods, and retailers. The problem is the same: assign the retailers to divisions so that no division runs out of any good.

Practically any large project can benefit from optimization. Combinatorial optimization can even help you solve puzzles.