1D and 2D cutting optimization problems in industry

Research proposal, 2022 Fall

August 9, 2022

Problem description

In this research proposal, we would like to consider optimization problems coming from industrial applications. The 1D cut problem is the following. The status of the warehouse, i.e., the lengths of the available steel rods is a multiset:

$$W := \{w_1, w2, \ldots, w_n\}.$$

The orders, given as intervals of lengths (a, b) is also a multiset:

$$N := \{(a_1, b_1), (a_2, b_2), \ldots, (a_m, b_m)\}.$$

A pair means that we can cut a rod of any length in the interval (a_i, b_i). We are interested in if the order can be fulfilled, and if so, what is the minimum number of cuts necessary. In a more involved problem, we also consider the what are the remaining rods, each remaining rod of length w' gets a score $s(w')$ and then we would like to minimize the number of cuts plus $\sum_i s(w'_i)$.

In the 2D cut problem there are iron sheets of rectangular shape in the warehouse, given as a multiset

$$U := \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_m)\}.$$

The orders are a multiset of rectangles with tolerance, that is

$$M := \{[(a_1, b_1), (\epsilon_1, \delta_1)], [(a_2, b_2), (\epsilon_2, \delta_2)], \ldots, [(a_m, b_m), (\epsilon_m, \delta_m)]\}.$$

For each quadruple, any rectangle whose first dimension is between a_i and $a_i + \epsilon_i$ and second dimension is between b_i and $b_i + \delta_i$ sufficies. Only horizontal and vertical cuts are possible. We are interested in if an order can be satisfied and if so, what is the minimum number of cuts. In a more involved model, each cut gets a weight, and we would like to minimize the sum of the weights of the cuts. Also, we might consider the remaining rectangles, each gets a score, and we would like to minimize the sum of the weights of the cuts plus the sum of the weights of the remaining rectangles.

Qualifying problems

1. Balls are distributed in n boxes, such that in the k^{th} box there are k balls. In one step we can choose an arbitrary subset of the boxes and take out the same number of balls from each. Give an optimal algorithm that uses the smallest number of such steps to empty out all the boxes.

2. We are given n chips that can test each other in the following way: if two of the chips are connected, then both determine whether the other one is faulty or correct. A correct chip decides properly about the other, while a faulty chip can give an arbitrary answer. Assume, that more than half of the chips are correct. Give an algorithm that uses less than n tests of the above type to find a correct chip.

3. There are n^2 lightbulbs installed on an $n \times n$ table in a classroom. There exists a pushbutton for each row and each column of the table that changes the state of each bulb in the respective row or column to the opposite (from "on" to "off" or vice versa). At the beginning of the break each bulb is "off". During the break the students keep pushing the buttons in a random fashion. How many pushes the teacher needs to restore the original "off" state of each bulb if it cannot be seen on a pushbutton how many times it was pushed during the break?

4. The word $v \in \Sigma^*$ is an initial segment of $u \in \Sigma^*$, if there exists $w \in \Sigma^*$, such that $u = vw$. Similarly, v is a final segment of u, if there exists $y \in \Sigma^*$, such that $u = yv$. Give an $O(n)$ running time method that finds the longest proper initial segment of the length n word u, which is also a final segment of u. u is given by the character array $A[1 : n]$.

5. A certain string-processing language offers a primitive operation which splits a string into two pieces. Since this operation involves copying the original string, it takes n units of time for a string of length n, regardless of the location of the cut. Suppose, now, that you want to break a string into many pieces. The order in which the breaks are made can affect the total running time. For example, if you want to cut a 20-character string at positions 3 and 10, then making the first cut at position 3 incurs a total cost of $20 + 17 = 37$, while doing position 10 first has a better cost of $20 + 10 = 30$.

   Give a dynamic programming algorithm that, given the locations of m cuts in a string of length n, finds the minimum cost of breaking the string into $m + 1$ pieces.

6. You are given a rectangular piece of cloth with dimensions $X \times Y$, where X and Y are positive integers, and a list of n products that can be made using the cloth. For each product $i \in [1, n]$ you know that a rectangle of cloth of dimensions $a_i \times b_i$ is needed and that the final selling price of the product is c_i. Assume the a_i, b_i, and c_i are all positive integers. You have a machine that can cut any rectangular piece of cloth into two pieces either horizontally or vertically. Design an algorithm that determines the best return on the $X \times Y$ piece of cloth, that is, a strategy for cutting the cloth so that the products made from the resulting pieces give the maximum sum of selling prices. You are free to make as many copies of a given product as you wish, or none if desired