THE ROPE ORDERING PROBLEM

Matthew Tromp

E-mail: matthewktromp@gmail.com

1 Introduction

Emily is a rope saleswoman. She has n customers coming in today who each want a certain length of rope. Maybe the first customer wants 7 meters, the second customer wants 3 meters, and so on. We can represent the length of rope each customer wants as a sequence.

$$C' = (c'_1, c'_2, \dots, c'_n)$$

Emily has recent bought a new machine to make her job easier: the Rope Cutter 3000. At the start of the day, Emily feeds a length of rope into the Rope Cutter 3000, and as each customer comes in, the rope is cut to their desired length. It's a wonderful invention, but here's the problem: Emily doesn't have a nice long piece of rope to feed into the machine. All she has is a barrel of m rope pieces of different lengths and thicknesses. Maybe the first rope is 5 meters long and 2mm thick, the second one is 9 meters long and 6mm thick and so on. We can represent the ropes in the barrel as a multiset of (integer, integer) tuples representing the lengths and thicknesses of each rope.

$$R = \{(l_1, t_1), (l_2, t_2), \dots, (l_m, t_m)\}$$

Conveniently for Emily, it just so happens that the total length of rope in the barrel exactly equals the total length of rope that her customers want. That is,

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{m} l_i$$

Also conveniently, Emily's customers aren't picky. They don't care how thick the rope is, or if there's knots along it, as long as it's the right length. So here's Emily's plan: she's going to tie all the m piece of rope together to make one long rope, and then feed it into the machine. That is, she is going to create an ordered sequence

$$R' = ((l'_1, t'_1), (l'_2, t'_2), \dots, (l'_m.t'_m))$$

Such that every (l'_i, t'_i) in R' corresponds to a unique (l_j, t_j) in R.

Since the amount of rope in her barrel is exactly equal to the total length of rope that her customers want to buy, no matter how Emily ties the rope together, the machine is going to have to make exactly $n-1$ cuts. However, the thicker the piece of rope the machine is cutting, the longer it takes, so Emily would like to minimize the total thickness of rope that the machine needs to cut, $Thickness(C', R')$

For instance, let's say Emily has two customers coming in who want ropes of length 5 and 15.

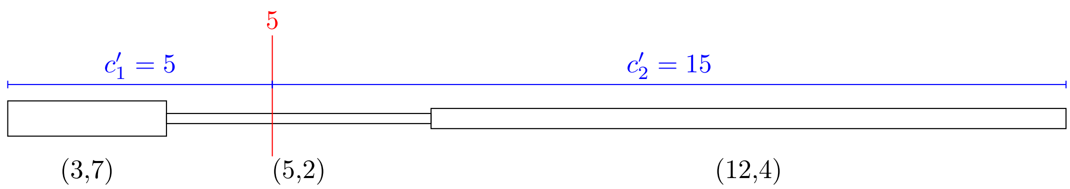
$$C' = (5, 15)$$

So we are going to need to make exactly 1 cut on our rope, at 5m. Emily has 3 ropes: a rope of length 12 and thickness 4, a rope of length 3 and thickness 7, and a rope of length 5 and thickness 2

$$R = \{(12, 4), (3, 7), (5, 2)\}$$

To minimize the total thickness of rope that needs to be cut, Emily would order the rope like so:

$$R' = ((3, 7), (5, 2), (12, 4))$$



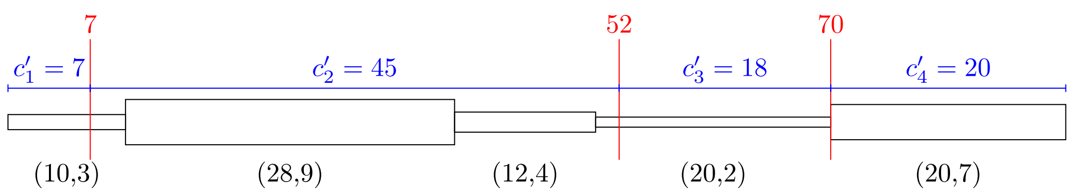
This way, the cut at 5m happens on the rope with thickness 2, so the total thickness of rope that needs to be cut $Thickness(C', R') = 2$ is minimized.

Let's consider a more involved example:

$$C' = (7, 45, 18, 20)$$
$$R = \{(10, 3), (12, 4), (20, 2), (20, 7), (28, 9)\}$$

Emily is going to need to make cuts at 7m, 52m and 70m along the rope. One optimal way of ordering the rope is like so:

$$R' = \{(10, 3), (28, 9), (12, 4), (20, 2), (20, 7)\}$$



The Rope Cutter 3000 cannot undo knots: if it wants to make a cut at a point where there happens to be a knot, it will cut the rope just before the knot. Also note that a rope can be cut more than once. So in this case, our cuts will fall on the ropes $(10, 3)$, $(20, 2)$ and $(20, 2)$ (again) for a total thickness of $Thickness(C', R') = 3 + 2 + 2 = 7$.

We will now consider a few variants of the rope ordering problem.

1.1 The rope/customer ordering problem

Filled with a newfound futurist fervour by her fantastic rope cutting machine, Emily has also made an appointment app, so she can tell customers in what order they should arrive to pick up their rope. In this variant, the customers are also a multiset C:

$$C = \{c_1, c_2, \ldots, c_n\}$$

That we can order into a sequence C' however we want, as we are ordering our ropes. We will call this variant the rope/customer ordering problem.

1.2 The customer ordering problem

Emily is now so busy maintaining the Rope Cutter 3000 and her new appointment app that she's had to hire an intern. Upon coming into work one morning, she finds that her intern has already tied her ropes up, without even checking which customers were coming in today! That is, the sequence R' of ropes has already been decided, and now all that Emily can do is change the order in which the customers come in.

2 NP-completeness

Each of the three variants above has a decision form (given an instance of the rope ordering problem $< C', R >$ or an instance of the rope/customer ordering problem $< C, R >$ or an instance of the customer ordering problem $< C, R' >$, and a maximum thickness M, does there exist an ordering of the ropes $< R' >$ or an ordering of the ropes and customers $< C', R' >$ or an ordering of the customers $< C' >$ such that the total thickness of the cut ropes is at most T?) are in NP. Our certificate consists of the ordered sequence of ropes and/or customers $< R' >$, $< C', R' >$ or $< C' >$, and we verify the solution by computing the total thickness of the rope being cut $Thickness(C', R')$ is at most M. They are all also NP-hard through reductions from variants of the subset sum problem, making them NP-complete.

2.1 The rope ordering problem is NP-hard

Given an instance of the subset sum problem consisting of a multiset of numbers

$$S = \{s_1, s_2, \ldots, s_n\}$$

and a target sum T, we can construct an equivalent instance of the rope ordering problem as follows. Let $\sum S$ be the sum of all elements in S. Define

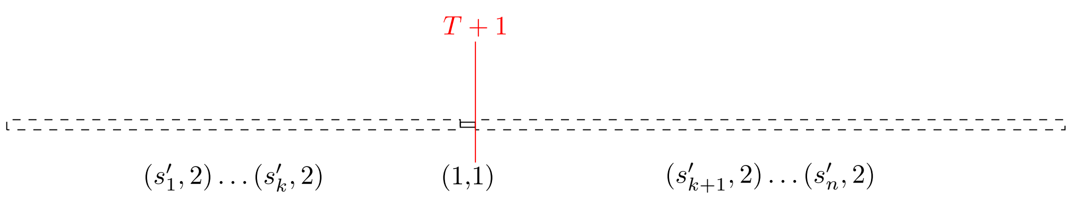
$$C' = (T+1, \left(\sum S\right) - T)$$
$$R = \{(s_1, 2), (s_2, 2), \ldots, (s_n, 2), (1, 1)\}$$

To ensure that this is a valid instance of the rope ordering problem, we must verify that

$$\sum_{c_i} c_i = \sum_{l_i} l_i$$
$$(T+1) + \left(\left(\sum S\right) - T\right) = \left(\sum_{s_i} s_i\right) + 1$$
$$\left(\sum S\right) + 1 = \left(\sum S\right) + 1$$

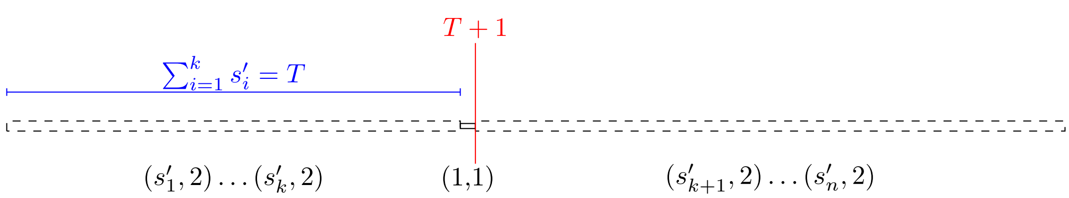
As we can see, this problem is very similar to the original subset sum problem. For each number s_i in the subset sum problem, we have a rope with length s_i and a thickness of 2, and we have added one additional input length with associated cost 1. Since only one cut will be made (at T), the total thickness of the cuts that have to be made will either be 1 or 2. Our decision problem will be "Given the C', R defined above and a maximum total thickness $M = 1$, is there an ordering R' of R such that $Thickness(C', R') \leq M$?"

The only way for this to happen is if the rope $(1,1)$ has its rightmost knot at exactly $T+1$ i.e. its leftmost knot must be at T:



If it were one meter to the right, the rope before it (with thickness 2) would be cut, and if it were one meter to the left, the rope after it (with thickness 2) would be cut

Now consider the ropes to the left of the rope $(1,1)$. $(1,1)$ will have its leftmost knot at T iff the sum of the lengths of the ropes to the left of $(1,1)$ add up to T.



That is, there is a subset $S_1 = \{s'_1, s'_2, \ldots, s'_k\}$ of S such that $\sum_{i=1}^{k} s'_i = T$. So the rope ordering problem is NP-hard via a reduction from the subset sum problem.

2.2 The rope/customer ordering problem is NP-hard

For this variant, we will be using a special version of the subset sum problem called the partition problem, which is also known to be NP-complete. In the partition problem, we have a multiset of positive integers

$$S = \{s_1, s_2, \ldots, s_n\}$$

and our target value is exactly half of the sum of all the values in S:

$$T = \frac{1}{2} \sum i = 1^n s_i$$

We define our instance of the rope/customer ordering problem as follows:

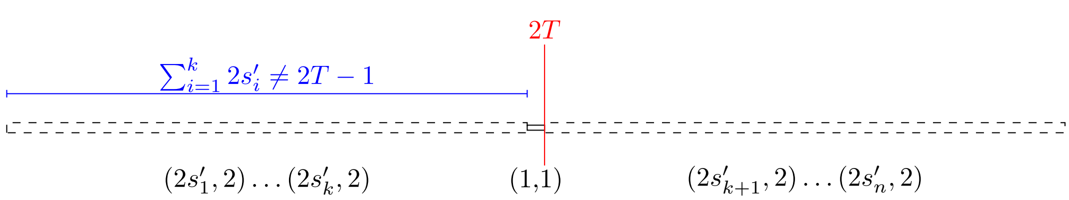
$$\begin{aligned} &C = \{2T+1, 2T\} \\ &R = \{(2s_1, 2), (2s_2, 2), \ldots (2s_n, 2), (1, 1)\} \end{aligned}$$

To ensure that this is a valid instance of the rope/customer ordering problem, we must verify that

$$\begin{aligned} \sum_{c_i} c_i &= \sum_{l_i} l_i \\ 2T + 1 + 2T &= \left(\sum_{s_i} 2s_i \right) + 1 \\ 4\left(\frac{1}{2} \sum i = 1^n s_i\right) + 1 &= 2\left(\sum S\right) + 1 \\ 2\left(\sum S\right) + 1 &= 2\left(\sum S\right) + 1 \end{aligned}$$

Since only one cut will be made (at either $2T+1$ or at $2T$), the total thickness of the cuts that have to be made will either be 1 or 2. Our decision problem will be "Given the C, R defined above and a maximum total thickness $M = 1$, are there orderings C' and R' of C and R such that $Thickness(C', R') \leq M$?"

First, note that any pair of orderings (C', R') with total thickness at most 1 must have $C' = (2T+1, 2T)$. As in the previous example, for our total thickness to be 1, the rightmost knot of the $(1, 1)$ rope must be lined up with the cut. If we select $C' = (2T, 2T+1)$, our cut will be at $2T$. Note that this is an even position. However, since every rope in R except $(1, 1)$ is of even length, the total length of rope that comes before $(1, 1)$ will be even. Therefore the leftmost knot of $(1, 1)$ will be at an even position, so the rightmost knot of $(1, 1)$ will be at an odd position, so it would be impossible for the cut at $2T$ to be lined up with it.



Therefore, we can conclude that any optimal ordering (C', R') will have $C' = (2T+1, 2T)$.

We can find an arrangement R' of R which places the rightmost knot of the rope $(1,1)$ at $2T+1$ iff there exists some subset $S_1 = \{s'_1, s'_2, \ldots, s'_k\}$ of S such that

$$\sum_{i=1}^{k} 2s'_i = 2T$$
$$\sum_{i=1}^{k} s'_i = T$$

So the rope/customer ordering problem is NP-hard via a reduction from the partition problem.

2.3 The customer ordering problem is NP-hard

Given an instance of the subset sum problem consisting of a multiset of numbers

$$S = \{s_1, s_2, \ldots, s_n\}$$

and a target sum T, we can construct an equivalent instance of the rope ordering problem as follows. Define

$$C = S = \{s_1, s_2, \ldots, s_n\}$$
$$R' = ((T-1, 2)(1,1)(\left(\sum S\right) - T, 2))$$

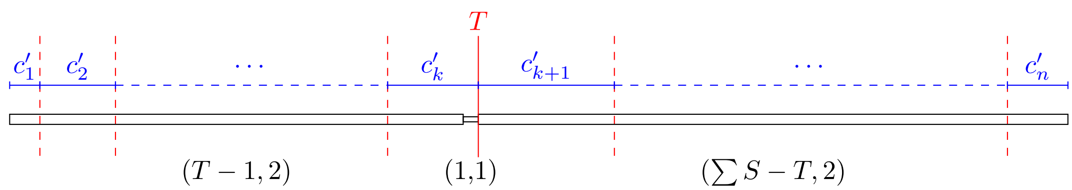
To ensure that this is a valid instance of the customer ordering problem, we must verify that

$$\sum_{c_i} c_i = \sum_{l_i} l_i$$
$$\sum S = (T-1) + 1 + \left(\left(\sum S\right) - T\right)$$
$$\sum S = \sum S$$

For this instance of the customer ordering problem, note that we have ropes spanning from 0 to $T-1$, from $T-1$ to T and from T to $\sum S$. Since we need to make one fewer cuts than there are customers, we will need to make $n-1$ cuts. Most of these cuts will fall along one of the two ropes of thickness 2, but at most one will fall along the rope of thickness 1 (by landing on the rightmost knot of the rope $(1,1)$). So, the total cut thickness will either be $2n-2$ (if no cut is made on the rope $(1,1)$) or $2n-3$ (if a cut is made on the rope $(1,1)$. Our decision problem will be "Given the C, R' defined above and a maximum total thickness $M = 2n-3$, is there an ordering C' of C such that $Thickness(C', R') \leq M$?"

Since the rightmost knot of the rope $(1,1)$ lies at position T, some cut k in an arrangement of customers $C' = (c'_1, \ldots, c'_n)$ will cut the rope $(1,1)$ iff the lengths of all cut segments up to and including k add up to T.

$$\sum_{i=1}^{k} c'_i = T$$



So, our decision problem will evaluate to "Yes" iff there exists a subset $S_1 = \{s'_1, \ldots, s'_k\}$ of S such that $\sum_{i=1}^{k} s'_i = T$. So the customer ordering problem is NP-hard via a reduction from the subset sum problem.