ICAI 2023
12th International Conference on Applied Informatics
Eger, Hungary, 2023

Optimal cutting arrangements in 1D

Bowen Li[a*], **Attila Sali**[bc†]

[a]Carleton College
lib2@carleton.edu

[b]Alfréd Rényi Institute of Mathematics
sali.attila@renyi.hu

[c]Department of Computer Science
Budapest University of Technology and Economics

Abstract

The following industrial problem [2], introduced in the framework of a Slovenian–Hungarian applied mathematics joint project, is treated. We are given a warehouse of steel rods of (maybe) different lengths. Orders of pieces of rods come in and they need to be served. However, cutting the rods is costly, so the number of cuts needs to be minimized. The way to do that is finding *exact fits*, that is collection of orders that fit on some rod of the warehouse and also exhaust that rod totally. In each exact fit case one cut can be saved, as "the remaining piece of the rod" need not be cut off.

The input is given as the status of the *warehouse*, i.e., the lengths of the available steel rods, which is a multiset:

$$W = \{w_1, w_2, \ldots, w_n\},$$

furthermore the *orders* are given as pairs (a, b), also forming a multiset:

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

*Research was done in Undergraduate Research Course at Budapest Semesters In Mathematics

†The second author was supported by the National Research, Development and Innovation Office (NKFIH) grants K–116769 and SNN–135643. This work was also supported by the BME-Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC) and by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Artificial Intelligence National Laboratory of Hungary.

The pair (a, b) means that some tolerance is allowed, that is we can cut a rod of any length in the interval $[a, b]$. We need to find a partition $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ of the multiset N and find a *valid* cutting assignment

$$\pi : \mathcal{P} \to W$$

such that: $\forall P_i \in \mathcal{P} : \sum_{(a,b)\in P_i} a \leq \pi(P_i)$. That is, the the orders that belong to the partition class P_i are assigned to be cut from rod $\pi(P_i)$. Amongst these valid cutting assignments we look for one that minimizes the number of cuts needed. Define the *exact fit* for $w \in W$ be the case when $\pi(P_i) = w$ and $\sum_{(a,b)\in P_i} a \leq \pi(P_i) \leq \sum_{(a,b)\in P_i} b$.

Fact 1. *If a valid cutting assignment exists then the number of cuts is minimized iff the number of exact fits are maximized.*

Theoretical results The following two decision problems are relevant.

Definition 2. *The* CutFeasibility *problem is as follows.*

Input *A warehouse multiset $W = \{w_1, w_2, \ldots, w_n\}$ and an orders multiset $N = \{(a_1, b_1), \ldots, (a_m, b_m)\}$.*

Question *Is there a valid cutting assignment?*

In the other problem it is assumed that a valid cutting assignment exists.

Definition 3. *The* MaxExactFit *problem is as follows.*

Input *A warehouse multiset $W = \{w_1, w_2, \ldots, w_n\}$ and an orders multiset $N = \{(a_1, b_1), \ldots, (a_m, b_m)\}$ such that a valid cutting assignment for W and N exists, furthermore a natural number k.*

Question *Is there a valid cutting assignment with at least k exact fits?*

For both of these problems it is clear that they are in NP, since a valid cutting assignment or one with the desired property is a good witness.

Proposition 4. *CutFeasibility is NP-complete.*

In fact, BinPacking is a special case of CutFeasibility.

Theorem 5. *MaxExactFit is also NP-complete.*

Although BinPacking is closely related to MaxExactFit, their optima may be attained in different cases-

Proposition 6. *There are examples of set of weights $s_1, s_2, \ldots, s_m$ such that optimal solution for BinPacking uses less bins of capacity one, than the optimal solution for MaxExactFit in case of unit length rods in the warehouse and $a_i = b_i = s_i$ for all $1 \leq i \leq m$.*

Practical approaches: Dynamic programming and clique search

Definition 7. *Let P be a set of orders and w be an element from the warehouse. Define*

$$fit(P,w) = \begin{cases} 1 & if \ \sum_{(a_i,b_i)\in P} a_i \le w \le \sum_{(a_i,b_i)\in P} b_i \\ 0 & otherwise \end{cases}$$

Define a compatibility graph G for order sets and warehouse sets as the following

$$V(G) = \{(P,w) : fit(P,w) = 1\}$$
$$E(G) = \{\{(P_i,w_i),(P_j,w_j)\} : P_i \cap P_j = \varnothing \text{ and } w_i \neq w_j\}$$

1. Identify all subsets of orders that match a certain rod *exactly*, taking into account the tolerances using *dynamic programming*.
2. Construct the compatibility graph and find the largest compatible set of sets that is a *largest clique* in G.

This method is ineffective, the size of the compatibility graph becomes too large even for moderate sized inputs.
$0-1$-linear programming Define indicator variables x_{ji} by

$$x_{ji} = \begin{cases} 1 & \pi(a_j,b_j) = w_i \\ 0 & \text{otherwise} \end{cases}$$

Then we have the following set of constraints:

(i) $\forall i\colon \sum_j a_j x_{ji} \le w_i$,

(ii) $\forall i\colon \sum_j b_j x_{ji} \ge \lambda_i w_i$,

(iii) $\forall j\colon \sum_i x_{ji} = 1$,

where $x_{ji}, \lambda_i \in \{0,1\}$. Condition (i) states that the orders assigned to rod i fit to that rod, (ii) means that if $\lambda_i = 1$ then we have an exact fit, while (iii) is to assure that the cutting assignment is valid, that is every order is assigned to exactly one rod. We want to maximize $\sum_i \lambda_i$. We used the Gurobi Solver [1] to solve test cases. We found that it is much more effective than the compatibility graph method.

References

[1] https://www.gurobi.com/.

[2] U. Čibej: *Personal communication*, 2022.