Choice of Method for General One-Dimensional Cutting Stock Problem

PETER TRKMAN, MIRO GRADIŠAR
Faculty of Economics
University of Ljubljana
Kardeljeva ploščad 17, 1000 Ljubljana
SLOVENIA

Abstract: - Problems in operations research are often NP complete and optimal solution for larger problems cannot be obtained, therefore heuristic methods have to be used. The boundary between the size of the problem that can be solved optimally and those where heuristic methods should be used is not always clear. To solve this problem we propose a method based on decision trees. The method can be also used to determine which variables have the largest influence on complexity of the problem. The proposed method was tested on General One-Dimensional Cutting Stock Problem. The result is reduced trim loss.

Key-Words: - cutting stock problem, decision trees, optimization, linear programming, heuristic methods, exact method

1 Introduction

One-dimensional stock cutting occurs in many industrial processes and during the past few years it has attracted increased attention of researchers from all over the world. Most standard problems related to one-dimensional stock cutting are known to be NP-complete and in general a solution can be found by using approximate methods and heuristics. However, the unbelievable development of computers and constantly growing processing power are pushing the complexity limit of the cutting problems where exact methods could be used slightly up. Therefore importance of exact methods is growing and the number of practical situations where they can be used increases [4, 11].

The problem we deal with in this paper is General One-Dimensional Cutting Stock Problem (G1D-CSP), where all available stock lengths are different. Pattern-oriented approach (as classified in [1]) where most methods are based on Gilmore and Gomory´s "delayed pattern generation"[2,3] is not suitable for this purpose. Therefore item-oriented approach has to be used where every stock length is treated individually as patterns cannot be determined. There are two possibilities for the solution of this problem: either exact methods (branch and bound, dynamic programming) or approximation algorithms in form of sequential heuristic procedures (SHP). Normally the time complexity of SHP is lower and such methods are more suitable for larger cases. On the other hand exact method is better when the size of the problem does not exceed acceptable limits and becomes intractable.

If different methods are available for the solution of the same problem, then it can be difficult to select the right one for each individual case. Therefore we decided to propose an approach based on decision trees for the selection of suitable solution method, which will take in the account not only the problem size and its characteristics but also computer speed and the quality of the used solver. The decision tree that is generated as the result of the experiment can be used to select the appropriate method in further cases. The selection of the right method leads to a reduced trim loss that is an objective of our cutting stock plan.

The paper is organized as follows: in the next section the definition of cutting stock problem and short description of available methods for its solution is given. Then the reasons for the selection of decision tress are given. New approach based on decision trees for the selection of suitable solution method is proposed. At the end the approach is experimentally tested and it is shown on practical example that it brings reduced trim loss.

2 Problem Definition

The problem is defined as follows: for every customer order a certain number of stock lengths is

available. In general all stock lengths are different. We consider the lengths as integers. If they are not originally integers we assume that it is always possible to multiply them with a factor and transform them to integers. It is necessary to cut a certain number of order lengths into required number of pieces. The goal is to minimize trim loss in the cutting stock process.
Following notation is used:
s_i = order lengths: i=1,….,n
b_i= required number of pieces of order length s_i
d_j= stock lengths; j=1,…,m
x_{ij}=number of pieces of order length s_i having been cut from stock length j
Depending on the available material two cases are possible:
Case 1: the order can be fulfilled as the abundance of material is in stock.

(1) $\min \sum_{j=1}^{m} t_j$ (minimize trim loss which is smaller than *UB* (upper bound for the trim loss))
s.t.

(2) $\sum_{i=1}^{n} (s_i * x_{ij}) + \delta_j = d_j (1 - y_j) \quad \forall j$ (knapsack constraints)

(3) $\sum_{j=1}^{m} x_{ij} = b_i \;\; \forall\, i$ (demand constraints - the numbers of pieces are all fixed)

(4) $UB - \delta_j + UB(u_j - 1) \le 0 \;\; \forall j$ (u_j indicates whether the remainder of stock length *j* is greater than *UB*)

(5) $\sum_{j=1}^{m} u_j \le 1$ (maximum number of residual lengths that can be larger than *UB*)

(6) $\delta_j - t_j - (u_j + y_j) * (\max d_j) \le 0 \;\; \forall j$ (t_j indicates the extent of trim loss relating to stock length *j*)

(7) $UB \le \max s_i$
$x_{ij} \ge 0$, integer $\forall\, i, j$
$t_j \ge 0$ $\forall j$
$\delta_j \ge 0$ $\forall j$
$u_j \in \{0,1\}$ $\forall j$
$y_j \in \{0,1\}$ $\forall j$

Case 2: the order cannot be fulfilled entirely due to shortage of material

(1) $\min \sum_{i=1}^{n} \delta_j$ (minimize sum of trim losses)
s.t.

(2) $\sum_{i=1}^{n} (s_i * x_{ij}) + \delta_j = d_j \quad \forall j$ (knapsack constraints)

(3) $\sum_{j=1}^{m} x_{ij} \le b_i \;\; \forall\, i$ (demand constraints)

(4) $x_{ij} \ge 0$, integer $\forall\, i, j$
$\delta_j \ge 0$ $\forall j$.

In the cases with surplus of material an unutilized stock length that is larger than some UB (upper bound for trim loss) can be used further so it is not considered as waste but is returned to the stock. In order to prevent cutting plans that would cause ever growing stock additional condition is set: a maximally one stock length can be returned to the stock (case 1, condition 5). UB can be set arbitrarily between 0 and max s_i. As UB=min s_i is found in practise [5] it is also used for problems in this paper. In case 2 all available material is used in the cutting plan so UB is not set.
Currently 3 solution methods exist for this problem. Two of them are heuristic methods: COLA [5] and CUT [6] (it is shown in [7] that CUT is superior of the two methods so it used for comparison with exact method in this paper) and the exact method [8, 10]. Obviously within reasonable time limits the exact method finds an optimal solution only for smaller cases. But since branch and bound gradually works towards optimal solution even if optimal solution is not found, the obtained approximate solution in some acceptable time period can be comparable with the solution obtained with CUT.
In [8, 10] it is written that the exact method is suitable only for smaller problems. However neither the definition of "small problem" nor the criteria for the selection of the method for each individual case is given. The main contribution of this paper is to fill this void and provide exact procedure for selection of method for each problem based on its characteristics. The question that needs to be answered is how likely it is (for each individual case) that the optimal solution will be found with exact method within the given time limit.
This question can be answered by using mathematical analysis of computational complexity. But for precise answer we would need a very precise data of speed of particular processor executing specific instructions generated by specific compiler and detailed data about solver, which are usually not available. Even if they would be, the mathematical analysis would be extremely complicated. Therefore we decided to answer the question by using statistics. A new approach based on creation of decision tree

and its use for selection of right method is presented in the next section.

3 Problem Solution

Decision trees were chosen as our kind of the problem fulfils the key requirements that are needed for successful implementation of decision trees (as listed in [9]):

- attribute-value description: each test case in our example can be described with the same attributes (number of stock and order length, average demand per order length, different ratios etc.),
- predefined classes: each case is assigned to one of the two predefined classes (either the case can be solved optimally within time limit or not),
- discrete classes: both classes in our example are discrete,
- sufficient data: sufficient number of problem instances can be automatically generated and solved using problem generator and solving procedure as described in this paper,
- “logical” classification models: our example can be expressed as decision trees or sets of production rules.

From the previously published papers [8, 10] it is obvious that exact method is suitable for problems where the number of stock lengths is less than 5 and unsuitable for larger problems (number of stock lengths over 10).

Therefore we have decided to test the cases with number of stock and order lengths between 5 and 10 as we need to determine the appropriate method for problems in this range.

In order to implement this approach we need sufficient amount of input data. The easiest way to obtain this data is by using a problem generator. We decided to use problem generator PGEN that was suitable for our purpose. PGEN (described in [7]) is a problem generator for General One-Dimensional Cutting Stock Problems. It generates input data according to problem descriptors as random sample of one or more test problems.

Problem descriptors are:

u_1,u_2 – lower and upper bounds for stock lengths ($u_1 \le d_j \le u_2$; j=1….m)

m – number of different stock lengths

d – average demand per order length

v_1,v_2 –lower and upper bounds for order lengths ($v_1 \le s_i \le v_2$; i=1…n)

n – number of different order lengths

r – number of consecutive generated problem instances

The procedure shown in fig. 1 was used to determine 243 test cases. 5 test instances were generated for each case so we have 1215 problem instance in total.

```
for g=1 to 3
  for h=1 to 3
    for i=1 to 3
      for j=1 to 3
        for k=1 to 3
            u1=1000*g
            u2=2000*g
            m=(j*2)+3
            d=(i*2)+3
            v1=h*100
            v2=h*200
            n=(g*2)+3
            seed=1000000*n+1000*v1+10*v2+10*d+m
            r=5
            call PGEN (n, v1, v2,, d, m, u1, u2, seed, r)
        next
      next
    next
  next
next
```

Fig. 1: The dynamic programming scheme of the procedure PROGEN

All problem instances were then solved using the exact method with the MPL/CPLEX solver and for every instance the solution time, total trim loss and the fact whether the problem instance was solved optimally or not was recorded. Time limit for optimal solution was set at 1 minute. The experiments were carried on a PC (AMD, 1300 MHz). All cases were then distributed into two classes: 1 (optimally solved cases) and 0 (cases not solved optimally). The time limit for the solution for each problem instance was set at 1 minute.

The whole experiment, which means generating the data and solving all problem instances within the time limit of 1 minute, took just over 10 hours. MS Excel was used for collecting and saving the results. The procedure for the whole experiment was written in Visual Basic for Application.

1215 cases were then used as inputs for building a decision tree. First we had to decide which variables to use as attributes. Obviously the variables that are expected to have the influence on computational complexity of the model should be used. However the number of variables and constraints in the model alone is not a sufficient indicator of time complexity of the problem. Therefore we have chosen the following variables:

- m, n, d - obviously those variables have the influence on the size of the model as m and n influence the number of variables and constraints in the model, while d influence the number of possible combinations.

- v_1, v_2, u_1, u_2 were not included as absolute values but as part of the following ratios:

  - r - ratio between the average stock length and average order length: $\frac{u_1 + u_2}{v_1 + v_2}$. Earlier it was statistically established that higher ratio means better solutions with a heuristic method [6].

  - q - ratio between available material and total needs: $\frac{(u_1 + u_2) * m}{d * n * (v_1 + v_2)}$. Problems with higher q should be easier to solve than those with this ratio closer to 1.

```
q > 2.18239:
:...m <= 7:  1  (126/2)
:   m > 7:
:   :...n > 5: 1 (76/2)
:       n <= 5:
:       :...q <= 2.86825 : 0 (16/4)
:               q > 2.86825  : 1 (44/1)
q <= 2.18239:
:...r>15: 1 (30/2)
    r<=15
    :...m > 5
            :.....q > 1.69128
            :          :...n <= 5 : (28/2)
            :          :   n > 5  : (50/20)
            :          q <= 1.69128
            :          :...m > 7  : 0 (124/6)
            :              m<=7
            :              :
            :              ...n>=5 : 0 (115/20)
            :                 n<=5
            :                 :....r <= 5 : 1 (22/4)
            :                         r >  5 : 0 (16/2)
            m<=5
            :...r <=5 : 1 (65/7)
            r > 5
             :....q>1.6137 : 1 (19)
                 q<=1.6137
                  :..n<=5 : 1 (27/9)
                     n>5
                     :...r <=6.667: 0(20/7)
                        r > 6.667
                        :.n<=7: 0(38/17)
                          n>7
                          :..q<=0.78137: 1(14/3)
                             q >0.78137 : 0 (20/4)
```

Fig. 2: Decision tree used for selection of the method

70% of the data was used as training, 30% as test data. To avoid over fitting of the data the test required two branches with at least 10 cases. The decision tree shown in fig. 2 was generated using C5 program. The numbers in the brackets mean how many of the training cases belong to this leaf. The first number is the number of correctly classified cases and the second of incorrectly.

In total the decision tree predicted correctly whether the problem would be solved optimally within 1 minute in 86.1% of the training cases and 84,1% of the test data.

The decision tree can easily be implemented as a subroutine in any programming language and then used for the selection of appropriate method for each individual case.

From fig. 2 it is obvious that for this sort and size of the problem the ratio between available material and total needs has the greatest influence on the complexity of the problem, followed by number of stock lengths. On the other hand the influence of the number of order lengths and average demand per order length is surprisingly low.

Further testing of this decision tree was done on previously published problems [8] where 27 different problems (10 problem instances for each case) were generated and solved with both CUT and exact method, but no guidelines for selection of appropriate method were given.

The results are shown in table 1. Trim loss is calculated as a sum of trim losses of all 10 instances. Time limit for the solution with exact method is the same as previously – 1 minute. All experiments were carried on the same computer to assure that the results are relevant (obviously the exact method would be more suitable on a faster computer and vice versa).

Based on the decision tree the following cases were solved with the exact method: 1,4,7,11,12,13,16,19, 21,22, while others were solved heuristically.

In 2 problems both methods gave the same results. From the remaining 25 problems the right decision was made in 21 (84,0%) of the cases. From the 4 mistakes made 3 resulted in only marginally higher trim loss while only in one case (case no. 21) trim loss was considerably higher (438 cm instead of 0 cm).

The most important result is that the total trim loss would be 5593 cm if we solved all problems with exact method, 32268 cm with exact method and 5310 if we solve each problem with the method selected on the basis of the decision tree. This shows that the proposed approach can indeed lead to improved results and lower trim loss. Obviously each problem could be solved with both methods and the better results would be kept. However that would require additional time and effort while the use of decision tree enables us to solve each problem just once.

	Trim loss CUT		Trim loss exact	
Case no.	cm	%	cm	%
1	8	0.0213%	1	0.0027%
2	0	0.0000%	0	0.0000%
3	0	0.0000%	0	0.0000%
4	1182	1.5460%	1028	1.3445%
5	28	0.0162%	212	0.1229%
6	9	0.0039%	62	0.0269%
7	1940	2.7256%	1702	2.3912%
8	213	0.0980%	1457	0.6706%
9	285	0.0832%	1780	0.5196%
10	59	0.0807%	18	0.0246%
11	0	0.0000%	8	0.0052%
12	2	0.0009%	23	0.0103%
13	88	0.1103%	8	0.0100%
14	172	0.0575%	1613	0.5393%
15	22	0.0046%	1833	0.3866%
16	227	0.3155%	49	0.0681%
17	272	0.0949%	2074	0.7236%
18	541	0.0836%	6533	1.0099%
19	7	0.0095%	0	0.0000%
20	10	0.0042%	429	0.1803%
21	0	0.0000%	438	0.1217%
22	47	0.0618%	1	0.0013%
23	36	0.0120%	1043	0.3468%
24	159	0.0242%	5615	0.8531%
25	81	0.1085%	41	0.0549%
26	93	0.0311%	984	0.3294%
27	112	0.0163%	5316	0.7742%
total trim loss	5593		32268	

Table 1: The comparison of results between CUT and exact method

4 Conclusion

In the paper we proposed a new approach for selection of appropriate method for General One-Dimensional Cutting Stock Problem that could be used for other similar problems in operations research as well. Using a decision tree an appropriate method for each individual case can be chosen based on its characteristics and probability that the problem of this size can be solved optimally within the given time limit with exact method.

In the paper we showed that with this approach we can select the appropriate method in vast majority of cases and that it leads to reduced trim loss.

The other advantage of this approach is that it takes computer speed and quality of the solver into account. Also the approach can easily be understood and used even by people with little background in mathematics or operations research.

References

[1] H. Dyckhoff, A typology of cutting and packing problems, *European Journal of Operational Research*, Vol. 44, No. 2, 1990, pp. 145-159.

[2] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem, *Operations Research*, Vol. 9, No. 6, 1961, pp. 849-859.

[3] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem, Part II, *Operations Research*, Vol. 11, No. 6, 1963, pp. 863-888.

[4] C. Goulimis: Optimal solutions for the cutting stock problem, *European Journal of Operational Research,* Vol. 44, No. 2, 1990, pp. 197-208.

[5] M. Gradišar, J. Jesenko, G. Resinovič: Optimization of roll cutting in clothing industry, *Computer&Operations Research,* Vol. 24, No. 10, 1997, pp. 945-953.

[6] M. Gradišar, G. Resinovič, J. Jesenko, M. Kljajić: A sequential heuristic procedure for one-dimensional cutting, *European Journal of Operational Research,* Vol. 114, No. 3, 1999, pp. 557-68.

[7] M. Gradišar, G. Resinovič, M. Kljajič: Evaluation of algorithms for one-dimensional cutting, *Computers& Operations Research,* Vol. 29, No. 9, 2002, pp. 1207-1220.

[8] M. Gradišar, P. Trkman, M. Indihar Štemberger: Exact solution of general one-dimensional cutting stock problem, *Working Paper No. 124*, Faculty of Economics, University of Ljubljana, 2002.

[9] R. Quinlan: C4.5: *Programs for machine learning*, Morgan Publishing House, 1993.

[10] P. Trkman, M. Gradišar: Eksaktna rešitev problema enodimenzionalnega razreza (in Slovene), *Slovenian Informatics Conference*, Portorož, Slovenia, 2002.

[11] P.H. Vance, *Branch-and-Price Algorithms for the One-Dimensional Cutting Stock Problem*, Computational optimizations and applications, Vol. 9, No. 3, 1998, pp 211-228.