ONE-DIMENSIONAL CUTTING STOCK OPTIMIZATION: THE CASE OF A LOW RATIO BETWEEN STOCK AND ORDER LENGTHS

MIRO GRADIŠAR

Faculty of Economics, University of Ljubljana, 1000 Ljubljana, Kardeljeva ploscad 17, Slovenia, miro.gradisar@ef.uni-lj.si

The article describes a new method for optimizing one-dimensional stock cutting in the case of a low ratio between stock and order lengths. The proposed method can resolve the general cutting stock problem which means that standard stock lengths, non-standard stock lengths or a combination of both can be cut in exactly the required number of pieces. A sample problem is presented and solved and a comparison with other methods is made. If the ratio between the average stock length and the average order length is less than 4, better results than existing methods can be expected.

Keywords: cutting, optimization, heuristic

Subject classifications: Programming: Integer: Algorithms: Heuristic.
Area of review: Production/scheduling: Cutting stock/trim.

OPTIMIZACIJA ENODIMENZIONALNEGA RAZREZA: PRIMER NIZKEGA RAZMERJA MED DOLŽINAMI NA ZALOGI IN V NAROČILU

V članku je opisana nova metoda za optimizacijo enodimenzionalnega razreza za primer nizkega razmerja med dolžino palic na zalogi in dolžino palic v naročilu. Predlagana metoda rešuje splošni problem razreza, kar pomeni, da omogoča optimizacijo razreza natančnega števila standardnih ali nestandardnih dolžin oziroma kombinacije obeh. Metoda je predstavljena na praktičnem primeru, hkrati pa je opravljena primerjava s podobnimi metodami, ki rešujejo isti problem. Če je razmerje med povprečno dolžino na zalogi in povprečno dolžino v naročilu manj kot 4, potem je metoda boljša od obstoječih, kar pomeni, da generira manj neuporabnega ostanka.

Ključne besede: razrez, optimizacija, hevristična metoda

1. Introduction

A one-dimensional cutting stock problem (CSP) occurs in different industrial processes. Many exact or heuristic methods based on an *item-oriented* (e.g., Gradisar et al. 1997, Gradisar et al. 1999a), *pattern-oriented* (e.g., Gau and Washer 1995) or mixed (e.g., Gradisar et al. 1999b) approach to solving CSPs have been developed. Exact methods (e.g., Amor et al. 2006, Cordeau 2006, Alves and de Carvalho 2008) offer optimal solutions but they can only be used to resolve small to medium-size instances. Also heuristic solutions can lead to results with a very low unnecessary trim loss which means that further considerable trim loss reductions are hardly possible. The enhancement of the method, for example, in (Gradisar and Trkman 2005) leads to a reduction of trim loss from 0.025% to 0.015%. Although the improvement is 60% it represents a very small contribution to a reduction of company costs.

The trim loss does not only depend on the optimization method but also on the nature of the problem. In practice, one can find a general one-dimensional CSP (Gradisar et al. 2002) with an average trim loss amounting to 10 or even 20% (e.g., Erjavec et al., 2008). The reason for that is the very low ratio between stock and order lengths. A considerable improvement of the optimization method would in such cases also lead to important cost reductions. However, generally acceptable solutions to this problem have not appeared in the literature so far. Therefore, the purpose of this article is to propose a new optimization method for solving a general CSP where the ratio between stock and order lengths is relatively low.

The rest of the article describes the cutting problem and development of a solution in the form of a computer program. A sample problem is presented and solved, while a comparison with other methods is made.

2. Definition of the problem

For every customer order a sufficiently large stock of material is available. Most of the stock has the same length or there are a few different standard lengths. Some of the stock can be of several different non-standard lengths as they are leftovers of previous orders and there is usually just one piece per one non-standard length. We consider the lengths as integers. If they are not integers then we assume that it is always possible to transform them into integers. An order consists of a request for a given number of order lengths into the required number of pieces. The following notation is used:

l_i order lengths; $i = 1,...,m$ (order lengths are sorted in a decreasing order: $l_1 \geq l_2 \geq l_3 \ldots$),
n_i the required number of pieces of order length l_i,

L_k stock lengths; $k = 1,\ldots,p$,
N_k number of pieces in stock of length L_k.

The cutting plan consists of cutting patterns that have been cut from different stock lengths and of the corresponding frequencies needed to satisfy all orders. Cutting pattern j that was cut from stock length k may be expressed by a vector

$$(a_{1jk}, a_{2jk}, a_{3jk}, \ldots, a_{mjk}) \tag{1}$$

that satisfies

$$\sum_{i=1}^{m} l_i \cdot a_{ijk} \leq L_k, \tag{2}$$

$$a_{ijk} \geq 0 \text{ and integer}, \tag{3}$$

in which a_{ijk} represents the number of times order length l_i appears in this particular pattern. Let us denote:

x_{jk} frequency of cutting pattern j having been cut from stock length k,

z_k total number of cutting patterns (1) cut from stock length k satisfying (2) and (3).

The following integer programming model can be formulated:

$$\min \sum_{k=1}^{p} \sum_{j=1}^{z_k} x_{jk} \cdot L_k \qquad \text{(minimize the sum of stock lengths to be cut)} \tag{4}$$

s.t.

$$\sum_{j=1}^{z_k} x_{jk} \leq N_k \qquad \forall\, k \quad \text{(stock constraints)} \tag{5}$$

$$p \quad z_k$$

$$\sum_{k=1} \sum_{j=1} a_{ijk} \cdot x_{jk} = n_i \quad \forall\, i \qquad \text{(demand constraints)} \tag{6}$$

$$x_{jk} \geq 0 \text{ and integer} \quad \forall\, j,k. \tag{7}$$

This description is similar to a general one-dimensional CSP and slightly different to those of Gau and Washer (1995) because of the multiple stock lengths, stock constraints and items of order lengths which must be cut into the exactly demanded number of pieces. Therefore, the described problem cannot be solved using the classic hybrid algorithm developed by Gilmore and Gomory (1961, 1963).

According to the typology of cutting and packing problems (Dyckhoff 1990, Washer et al. 2007, Trkman et al. 2007), this problem can be described as pure C&P with input minimization, heterogeneous large objects and weakly heterogeneous small objects or 1/V/D/M/IN where 1 stands for a one-dimensional problem, V means that all items need to be produced from a selection of large objects, D means that all large objects can be different, M indicates many small items of several dimensions and IN stands for instantaneous.

3. Development of the solution

The proposed sequential heuristic procedure is developed as an iteration of four basic steps. Each time a proportion of demand is satisfied. The procedure terminates when all of the demand is fulfilled. At the beginning, all stock lengths belong to the set of unprocessed stock lengths. The number of unprocessed pieces (UN_k) of each stock length L_k equals N_k while the number of unprocessed pieces (un_i) of each order length l_i equals n_i. The set of processed stock and order lengths is empty. Upon each iteration, the set of unprocessed pieces of stock lengths is reduced. The number of cut pieces of some order lengths also changes, as do the processed stock lengths which become equal to the trim loss. When all pieces of some stock length have been processed in a previous iteration this stock length cannot be used in the next one. At the end, all un_i equals 0. An iteration consists of the following steps:

Step 1: Solve the knapsack problem and find the optimal and next to optimal solution for each stock length by only taking unprocessed pieces of order lengths into account.
Step 2: Sort the cutting patterns obtained in step 1 according to a_{1jk}, a_{2jk}, a_{3jk}, ... The result is a sorted list of patterns. On top of the list are patterns containing longer order lengths, while the opposite are found on the bottom.

Step 3: Starting at the top of the sorted list of patterns and moving sequentially down select the corresponding frequency for each pattern by taking unprocessed pieces of order and stock lengths into account. The corresponding frequency is the highest frequency which at that moment is not higher than the number of unprocessed pieces of corresponding stock length and at the same time low enough to prevent the overproduction of any order length. Consequently, immediately after the selection of an individual frequency reduce the corresponding number of unprocessed pieces of stock and order lengths.
Step 4: If all the orders are still not fulfilled, then go back to step 1, otherwise stop.

The algorithm is developed on the following assumptions:

1. The optimal or near to optimal solution of the cutting problem in the case of a lower ratio between the average stock and order lengths consists of the optimal or next to optimal solution of knapsack problems.
2. It is easier to find a good solution if the cutting patterns containing largest order lengths are processed earlier.

The first assumption is based on the fact that in the case of a lower ratio between the average stock and order lengths the number of possible solutions is lower. The probability that the optimal or next to optimal solution of knapsack problems is also the optimal or near to optimal solution of the cutting problem is therefore greater.

The second assumption seeks to minimize the influence of *ending conditions* (Haessler and Sweeney 1991). It is a statistically proven general fact that it is easier to find a good solution if it is chosen from the largest possible set of possible solutions (Gradisar et al. 1999a). In order to keep the set of possible solutions as large as possible as long as possible during the cutting process the longer order lengths are processed earlier. The algorithm for the optimization of stock length cutting is shown in the flowchart in Figure 1.

Figure 1 Flowchart of the Cutting Algorithm

The flowchart indicates it is necessary to solve a series of knapsack problems in order to obtain cutting patterns and for each of them to then select the corresponding frequency for every iteration. The dynamic programming scheme of the *KNAPSACK* procedure for the case of four order lengths $l_1,\dots, l_4$ can be summarized as follows:

KNAPSACK procedure:

initialize $R_{min} \leftarrow R_{min-1} \leftarrow maxint,\ i \leftarrow j+1,\ a_{1jk} \leftarrow 0,\ a_{2jk} \leftarrow 0,\ a_{3jk} \leftarrow 0,\ a_{4jk} \leftarrow 0,\ a_{1ik} \leftarrow 0,\ a_{2ik} \leftarrow 0,\ a_{3ik} \leftarrow 0,\ a_{4ik} \leftarrow 0$

for $l = min\{un_1, int(L_k / l_1)\}$ **to** 0 **step** *-1* **do**

  $D_1 \leftarrow L_k - l_1 \cdot l$

  for $m = min\{un_2, int(D_1 / l_2)\}$ **to** 0 **step** *-1* **do**

    $D_2 \leftarrow D_1 - l_2 \cdot m$

    for $n = min\{un_3, int(D_2 / l_3)\}$ **to** 0 **step** *-1* **do**

      $D_3 \leftarrow D_2 - l_3 \cdot n$

        $t \leftarrow min\{un_4, int(D_3 / l_4)\}$

      $R \leftarrow D_3 - l_4 \cdot t$

        if $R < R_{min}$

        then

        $R_{min-1} \leftarrow R_{min}$

        $R_{min} \leftarrow R$

        $a_{1ik} \leftarrow a_{1jk},\ a_{2ik} \leftarrow a_{2jk},\ a_{3ik} \leftarrow a_{3jk},\ a_{4ik} \leftarrow a_{4jk}$

        $a_{1jk} \leftarrow l,\ a_{2jk} \leftarrow m,\ a_{3jk} \leftarrow n,\ a_{4jk} \leftarrow t$

      endif

    endfor

    endfor

endfor

In the *KNAPSACK* procedure a sequence of vectors (1) is generated in a lexicographically decreasing order to find the optimal and second to optimal pattern for L_k. The *int* function converts a numeric expression into an integer; all digits to the right of the decimal place are ignored.

The time complexity of the proposed algorithm can be calculated similarly as in (Gradisar et al. 1999a) and mostly depends on *m*. An acceptable response time can be expected if *m* is 7 or less.

4. Results

The proposed algorithm is written in the FORTRAN programming language which enables very rapid processing. The program consists of less than 1,000 lines of code. The data input and the printout of the results are made in 4GL. The program can be run on a personal computer. It was called LCUT. LCUT is not intended for general use but for those cases with a small ratio between stock and order lengths. It can be used as an independent solution or as 'add-in' for existing applications. Because of the complexity of calculations the time limit for solving each problem is set to 30 seconds. Therefore LCUT needs to have the following constraints:

- the ratio between the average stock and the average order length ≤ 10
- the number of different order lengths ≤ 7
- the number of pieces for each order length ≤ 99
- the number of different stock lengths ≤ 20
- the number of pieces for each stock length ≤ 99

Creating a cutting plan for a case falling within the listed parameters takes less than 30 seconds on a personal computer (Pentium 4).

As an illustration of the use of LCUT a typical practical case was selected. The data were supplied by a leading retailer of technical products in South-east Europe. One of its retail areas is the resale of various metal bars which need to be cut by the retailer to meet customers' demands. The input data are shown in Figure 2. The customer order contains four different order lengths. The sum of the required pieces is 34. There is an abundance of material in stock: 57 pieces of 15 different lengths. Standard and non-standard stock lengths are not treated separately. All lengths are in centimeters. The ratio between the largest stock and the shortest order length is 3.7.

Calculation of the results takes less than 1 second. 29 pieces of stock lengths are used. All patterns consists of one or a maximum of two pieces. The frequency of the patterns range from 1 to 11. The total trim loss is 3,345 cm, which makes up 11.98% of the total utilized stock lengths.

The cutting plan of the presented case was also calculated with three other computer programs: CUT (Gradisar et al. 1999a), C-CUT (Gradisar and Trkman 2005) and the LOPT commercial application which is available on the web (bestopt.de). These programs were selected for comparison because they solve the same type of problem. In all three cases the results are identical and are presented at the end of Figure 2. The total trim loss is 3,590 cm, which makes up 12.75% of the total utilized stock lengths. The comparison with LCUT shows that in the case of LCUT the trim loss is 245 cm less, which means a 0.77% saving of total utilized stock lengths.

Figure 2 Example of the LCUT, CUT, C-CUT and LOPT Results

To test LCUT more extensively a series of problem instances was generated. To generate the problem instances a slightly modified problem generator *CUTGEN1* (Gau and Washer 1995, Gradisar et al. 2002) was used. Input data were generated according to problem descriptors as a random sample of one or more test problems. The problem descriptors are:

m - number of different order lengths

v_1, v_2 - lower and upper bounds of order lengths, i.e. $v_1 \leq l_i \leq v_2$ ($i = 1,...,n$)

n - average demand per order length

P - number of different standard stock lengths

s_1, s_2 - lower and upper bounds of a standard stock length, i.e. $s_1 \leq L_k \leq s_2$ ($k = 1,...,P$)

N - number of pieces of standard stock lengths

p - number of non-standard stock lengths

u_1, u_2 - lower and upper bounds of a non-standard stock length, i.e. $u_1 \leq L_j \leq u_2$ ($j = 1,...,p$)

r - number of consecutive generated problem instances

The test problems were generated with the following parameter values:

- determination of order lengths and demands:

  By assigning different values to problem parameters m (m = 5, 6, 7), v_1 and v_2 (v_1 = 100 and v_2 = 200, v_1 = 200 and v_2 = 300, v_1 = 300 and v_2 = 400) and n (n = 10, 20, 30) and combining them with each another 27 test cases were generated.

- determination of standard stock lengths:

  The number of standard stock lengths P was 10 and the number of pieces N was 8. This means that the stock consists of 90 pieces, 80 of standard stock lengths and 10 of non-standard. In this case, it was also possible to solve the problem with CUT where the highest number of stock lengths is limited to 99. The lower and upper bounds of standard stock length s_1, and s_2 were set in such a way that the ratio between the largest stock and shortest order length for three pairs v_1, v_2 was 3, 4 and 5 (s_1 = 200 and s_2 = 300, s_1 = 600 and s_2 = 800, s_1 = 1200 and s_2 = 1500).

- determination of non-standard stock lengths:

  The number of non-standard stock lengths p was also 10. The lower and upper bounds of non-standard stock length u_1 and u_2 were set similarly as the standard lengths, except they were 50% shorter (u_1 = 100 and u_2 = 150, u_1 = 300 and u_2 = 400, u_1 = 600 and u_2 = 750).

For each test case 10 consecutive problem instances (r = 10) were generated. In total there were 270 problem instances. The problem descriptors and seeds used to generate the data for 27 test cases are presented in the first 12 columns of Table 1. All lengths are in centimeters.

Table 1 The Problem Descriptors and Results for 27 Test Cases

The problem instances were solved with two computer programs LCUT and CUT in order to compare both algorithms. CUT was selected for a more extensive comparison because LCUT is meant to be an ‘add-in’ or upgrade of CUT. Although both algorithms solve a similar problem there is a difference in the calculation of the trim loss between CUT and LCUT. CUT does not treat all unused parts of stock lengths longer than the shortest order length as a trim loss. For the purpose of comparison the leftovers of LCUT were treated in the same way.

The results are shown in the last five columns of Table 1. Each row presents one test case as the average trim loss in centimeters and in percent calculated from 10 problem instances for CUT and LCUT. The calculation of each of the 270 problem instances takes less than 1 second.

From the last column of Table 1 it is evident that in 19 out of 27 test cases CUT offers better results than LCUT. The difference between the average trim loss of CUT and LCUT is approximately 10%. The smallest difference is 1% in the case of m is 5 and the greatest 23% in the case of m is 6. If m is 7, then the average difference is 10%.

On the other hand, in all test cases with the lowest ratio between the largest stock and shortest order length and where n is 10 or 20 the results of LCUT are better. The greatest reductions of trim loss are seen in test cases 1 and 2. If n is 30 the result of LCUT is only better in test case 3. This means that the effectiveness of LCUT does not only depend on the ratio between the stock and order length but also on n. If n is smaller the results of LCUT are better. In the first three test cases n is 10, 20 and 30 and with a growing n the reduction of the trim loss decreases: 8.05%, 5.93% and 2.56%. A similar situation is found for the other test cases. Such dependency between n and the trim loss can be expected since a growing n increases the number of possible solutions and LCUT is based on the assumption that this number is low. In general, a low number of possible solutions means a higher trim loss. In all the test cases where the results of LCUT are better the average trim loss is relatively high, between 8% and 20%.

5. Conclusion

The paper analyzes the problem of reducing the trim loss in one-dimensional stock cutting where the ratio between the stock and order length is low. The new heuristic procedure was developed in the form of a computer program called LCUT. Testing LCUT and comparing it with CUT showed that LCUT provides better results if the ratio between the average stock and average order length is less than 4, the average number of pieces per order length is less than 30 and the average trim loss is more than 8%. In most other cases the results of CUT are better. Therefore, LCUT cannot be used to replace CUT or some other method but rather as a supplement or 'add-in' in the abovementioned cases especially because the algorithm is relatively fast and in most cases takes less than 1 second to calculate the cutting plan.

References

Amor H. B., J. Desrosiers, J. M. V. de Carvalho. 2006. Dual-Optimal Inequalities for Stabilized Column Generation. *Oper. Res.* 54 454-463.

Alves C. and V. de Carvalho. 2008. A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. *Comp. Oper. Res.* 35 1315-1328.

Cordeau J. F. 2006. A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Oper. Res.* 54 573-586.

Dyckhoff H. 1990. A typology of cutting and packing problems. *Eur. J. Oper. Res.* 44 145-159.

Erjavec J., P. Trkman, M. Gradisar. Renovation of the Cutting Stock Process. *Internat. J. Production Res.* In press.

Gau T., G. Washer. 1995. CUTGEN1: A problem generator for the Standard One-dimensional Cutting Stock Problem. *Eur. J. Oper. Res.* 84 572-579.

Gilmore P. C. and R. E. Gomory. 1961. A linear programming approach to the cutting stock problem. *Oper. Res.* 9 849-859.

Gilmore P. C. and R. E. Gomory. 1963. A linear programming approach to the cutting stock problem, Part II. *Oper. Res.* 11 863-888.

Gradisar M., J. Jesenko, G. Resinovic. 1997. Optimization of roll cutting in clothing industry. *Comp. Oper. Res.* 24 945-953.

Gradisar M., G. Resinovic, J. Jesenko, and M. Kljajic. 1999a. A sequential heuristic procedure for one-dimensional cutting. *Eur. J. Oper. Res.* 114(3) 557-568.

Gradisar M., M. Kljajic, and G. Resinovic. 1999b. A hybrid approach for optimization of one-dimensional cutting. *Eur. J. Oper. Res.* 119(3) 165-174.

Gradisar M., G. Resinovic, and M. Kljajic. 2002. Evaluation of algorithms for one-dimensional cutting. *Comp. Oper. Res.* 29(9) 1207-1220.

Gradisar M. and P. Trkman. 2005. A combined approach to the solution to the general one-dimensional cutting stock problem. *Comp. Oper. Res.* 32 (7) 1793-1807.

Haessler R. W., P. E. Sweeney. 1991. Cutting stock problems and solution procedures. *Eur. J. Oper. Res.* 54 141-150.

Trkman, P. and M. Gradisar. 2007. One-dimensional cutting stock optimization in consecutive time periods. *Eur. J. Oper. Res.* 179(2) 291-301.

Wascher, G., H. Hausner, and H. Schumann. 2007. An improved typology of cutting and packing problems. *Eur. J. Oper. Res.* 183(3) 1109-1130.

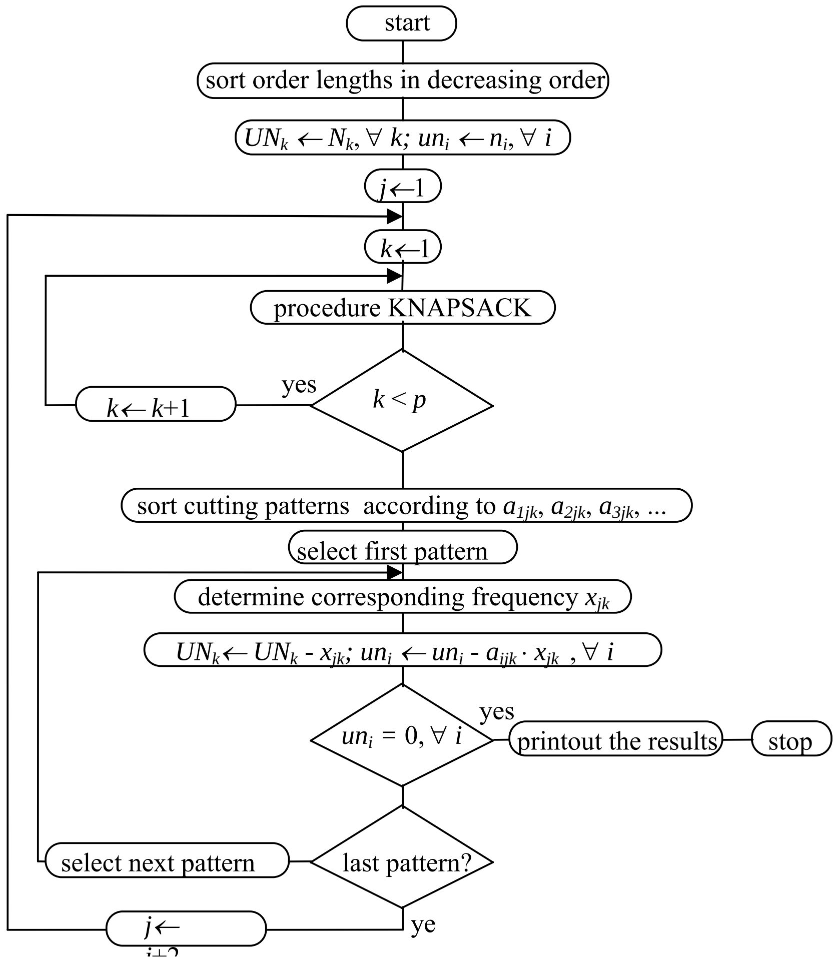

Figure 1 Flowchart of the Cutting Algorithm

DETAILS OF ORDER LENGTHS

No.	length	pieces
1	965	12
2	780	7
3	538	10
4	430	5

DETAILS OF STOCK LENGTHS

No.	length	pieces
1	1200	12
2	750	10
3	685	10
4	590	10
5	865	4
6	1600	2
7	600	1
8	500	1
9	1400	1
10	640	1
11	800	1
12	765	1
13	670	1
14	690	1
15	820	1

RESULTS OF LCUT

utilized stock lengths

No.	length	pieces	pattern	trim loss	-%
1	1200	11	1x965[1]	235	19.58
4	590	10	1x538[3]	52	8.81
5	865	2	2x430[4]	5	0.58
5	865	1	1x780[2]	85	9.83
6	1600	2	2x780[2]	40	2.50
9	1400	1	1x965[1]	5	0.36
			1x430[4]		
11	800	1	1x780[2]	20	2.50
15	820	1	1x780[2]	40	4.88

Total trim loss: 3345 (11.98%)

RESULTS OF CUT, C-CUT, LOPT

utilized stock lengths

No.	length	pieces	pattern	trim loss	-%
1	1200	12	1x965[1]	235	19.58

4	590	9	1x538[3]	52	8.81
5	865	2	2x430[4]	5	0.58
6	1600	2	2x780[2]	40	2.50
8	500	1	1x430[4]	70	14.00
9	1400	1	1x780[2]	82	5.86
			1x538[3]		
11	800	1	1x780[2]	20	2.50
15	820	1	1x780[2]	40	4.88

Total trim loss: 3590 (12.75%)

Figure 2 Example of the LCUT, CUT, C-CUT, and LOPT Results

												CUT		LCUT		
No.	*m*	v_1	v_2	*n*	*P*	s_1	s_2	*p*	*u1*	*u2*	seed	average trim loss	-%	average trim loss	-%	
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*13-15*
1	5	100	200	10	10	200	300	10	100	150	111	1865	28.29	1334	20.24	531
2	5	100	200	20	10	200	300	10	100	150	112	3526	25.22	2698	19.29	828
3	5	100	200	30	10	200	300	10	100	150	113	3029	14.23	2484	11.67	545
4	5	200	300	10	10	600	800	10	300	400	121	258	2.23	487	4.13	-229
5	5	200	300	20	10	600	800	10	300	400	122	779	3.28	1041	4.39	-262
6	5	200	300	30	10	600	800	10	300	400	123	1723	4.74	2097	5.77	-374
7	5	300	400	10	10	1200	1500	10	600	750	131	192	1.18	547	3.36	-355
8	5	300	400	20	10	1200	1500	10	600	750	132	569	1.55	708	1.93	-139
9	5	300	400	30	10	1200	1500	10	600	750	133	1186	2.10	1905	3.38	-719
10	6	100	200	10	10	200	300	10	100	150	211	1169	10.25	967	8.47	202
11	6	100	200	20	10	200	300	10	100	150	212	2510	13.71	2219	12.12	291
12	6	100	200	30	10	200	300	10	100	150	213	1706	8.28	2137	10.73	-431
13	6	200	300	10	10	600	800	10	300	400	221	277	1.86	533	3.58	-256
14	6	200	300	20	10	600	800	10	300	400	222	977	3.28	1258	4.22	-281
15	6	200	300	30	10	600	800	10	300	400	223	2280	7.10	2384	7.42	-104

16	6	300	400	10	10	1200	1500	10	600	750	231	67	0.31	405	1.92	-338
17	6	300	400	20	10	1200	1500	10	600	750	232	343	0.81	714	1.69	-371
18	6	300	400	30	10	1200	1500	10	600	750	233	645	1.10	1674	2.63	-1029
19	7	100	200	10	10	200	300	10	100	150	311	2806	24.38	2123	18.45	683
20	7	100	200	20	10	200	300	10	100	150	312	3254	14.68	2828	12.76	426
21	7	100	200	30	10	200	300	10	100	150	313	2504	11.28	2636	11.88	-132
22	7	200	300	10	10	600	800	10	300	400	321	638	3.41	784	4.20	-146
23	7	200	300	20	10	600	800	10	300	400	322	2538	7.25	3353	9.58	-815
24	7	200	300	30	10	600	800	10	300	400	323	4915	8.43	4789	8.21	126
25	7	300	400	10	10	1200	1500	10	600	750	331	156	0.64	569	2.33	-413
26	7	300	400	20	10	1200	1500	10	600	750	332	935	1.95	1480	3.09	-545
27	7	300	400	30	10	1200	1500	10	600	750	333	2761	3.24	4071	4.78	-1310
sum												43608		48225		-4617

Table 1 The Problem Descriptors and Results for 27 Test Cases