



ORIGINAL ARTICLE

A software for the one-dimensional cutting stock problem

Murat Erşen Berberler *, Urfat Nuriyev, Ahmet Yıldırım

Ege University, Science Faculty, Department of Mathematics, 35100 Bornova, İzmir, Turkey

Received 8 June 2010; accepted 14 June 2010

Available online 1 July 2010

KEYWORDS

Cutting stock problem;
Bin packing problem;
Dynamic programming;
Heuristic algorithm;
Packet program

Abstract In this paper, one-dimensional cutting stock problem is taken into consideration and a new heuristic algorithm is proposed to solve the problem. In this proposed algorithm, a new dynamic programming algorithm is applied for packing each of the bins. The algorithm is coded with Delphi and then by computational experiments with the real-life constraint optimization problems, and the obtained results are compared with the other one-dimensional cutting stock commercial packages. The computational experiments show the efficiency of the algorithm.

© 2010 King Saud University. All rights reserved.

1. Introduction

Cutting stock problems (CSPs) exist during the manufacturing processes of many products in the processing and manufacturing industries, such as the aluminum windows/doors frame manufacturing industry, the timber industry and the shipbuilding industry, and have become an important aspect of these industries. To ensure that orders are delivered efficiently, the management usually utilizes various manufacturing methods to minimize production costs. Therefore, the cutting process, which is common in these industries, has been considered from time to time as one that needs to be optimized. Reducing trim loss is one of the major goals during cutting processes and is

also the main goal of 1D-CSP. Thus, the problem considered in this paper is to find a cutting plan that would minimize the waste of material when a set of orders that are different in length and quantity is to be cut from a pack of stocks with constant length (Yang et al., 2006).

The 1D-CSP (or CSP) in Fig. 1 is defined by the following data: $(m, L, l = (l_1, \dots, l_m), b = (b_1, \dots, b_m))$, where L denotes the length of each stock piece, m denotes the number of smaller piece types and for each type $i = 1, \dots, m$, l_i is the piece length, and b_i is the order demand. In a cutting plan, we must obtain the required set of pieces from the available stock lengths. The objective is to minimize the number of used stock lengths or, equivalently, trim loss (waste). In a real-life cutting process there are additional criteria, e.g., the number of different cutting patterns (setups) and open stacks.

A special case in which the set of small objects is such that only one item of each product type is ordered, i.e., $b_i = 1 \forall i$ (sometimes also when b_i are very small), is known as the *Bin-Packing Problem* (BPP, 1D-BPP). This special case, having a smaller overall number of items, is more suitable for pure combinatorial solution approaches (Belov, 2003).

We suppose that there are a definite number of bins of the same capacity and many packages of different shapes. BPP is the problem of placing all the packages in a minimum number

* Corresponding author. Tel.: +90 5359233879.

E-mail addresses: murat.ersen.berberler@ege.edu.tr (M.E. Berberler), ahmet.yildirim@ege.edu.tr (A. Yıldırım).

1018-3647 © 2010 King Saud University. All rights reserved. Peer-review under responsibility of King Saud University.

doi:10.1016/j.jksus.2010.06.009



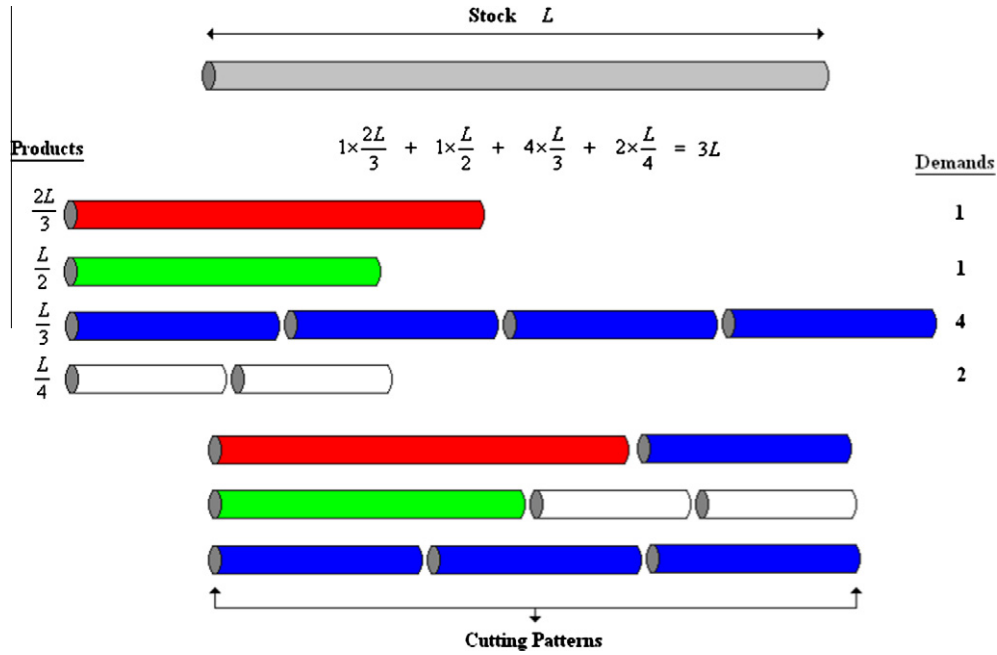


Figure 1 One-dimensional cutting stock problem with one stock type.

of bins without exceeding the capacity. It has many application fields such as scheduling television commercials of different lengths by 1 min time interval or cutting timbers, used for building, into pieces of different lengths (Gilmore and Gomory, 1961; Gilmore et al., 1963), loading trucks, storage problem and budget planning. As it is well known, the BPP is strong NP-complete, when formulated as a decision problem. As an optimization problem, bin packing is NP-hard (Garey and Johnson, 1979). Many heuristic algorithms are known for the solution of this problem in the literature (Kellerer et al., 2004; Martello and Toth, 1990). To solve the bin-packing problem, a dynamic programming-based algorithm is proposed in this study.

We give a new mathematical model of the CSP in the next section.

2. A new mathematical model of the CSP

This model differs from the model existing in the literature by accepting the inputs as the classified equal length items. One of the advantages of this difference is that the length of the array which includes the items shortens. The other is that there is no computational cost of classifying the items.

n items $\{a_1, \dots, a_n\}$ and m bins are given,
 w_j = weight of item a_j
 v_j = number of item a_j (demand)
 c = capacity of each bin

The objective is packing all the items into bins such that the total weight of each bin does not exceed its capacity c and the number of bins used is minimal, i.e., to minimize the wasted material. By this objective, a possible mathematical model is shown below,

$$\begin{aligned} & \text{minimize } z = \sum_{i=1}^m y_i \\ & \text{subject to } \sum_{j=1}^n w_j x_{ij} \leq c y_i, \quad i \in M \\ & \quad \sum_{i=1}^m x_{ij} = v_j, \quad j \in N \\ & \quad y_i = 0 \text{ or } 1, \quad i \in M \\ & \quad x_{ij} = 0 \text{ or } k, \quad i \in M, j \in N \end{aligned}$$

Here,

$$y_i = \begin{cases} 1 & \text{if bin } i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} k & \text{if amount of } k \text{ of item } j \text{ is assigned to bin } i \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that w_j and c are positive integers and $w_j \leq c$ ($j \in N$). $N = \{1, 2, \dots, n\}$, $M = \{1, 2, \dots, m\}$.

Solving the problem, a heuristic algorithm based on dynamic programming is developed. In this algorithm, for loading each of the bins, a novel dynamic programming algorithm is used. In the literature, this sub-problem is called Subset-Sum Problem (SSP).

3. A new algorithm for the CSP

The terms of BPP are used here to simplify the expression of the algorithm of CSP.

CSP (BPP) algorithm:

BPP1: $WV = \{(w_1 v_1), (w_2 v_2), \dots, (w_n v_n)\}$ and c are entered. The entries are the weight of the items, the demands and the capacity of a bin.

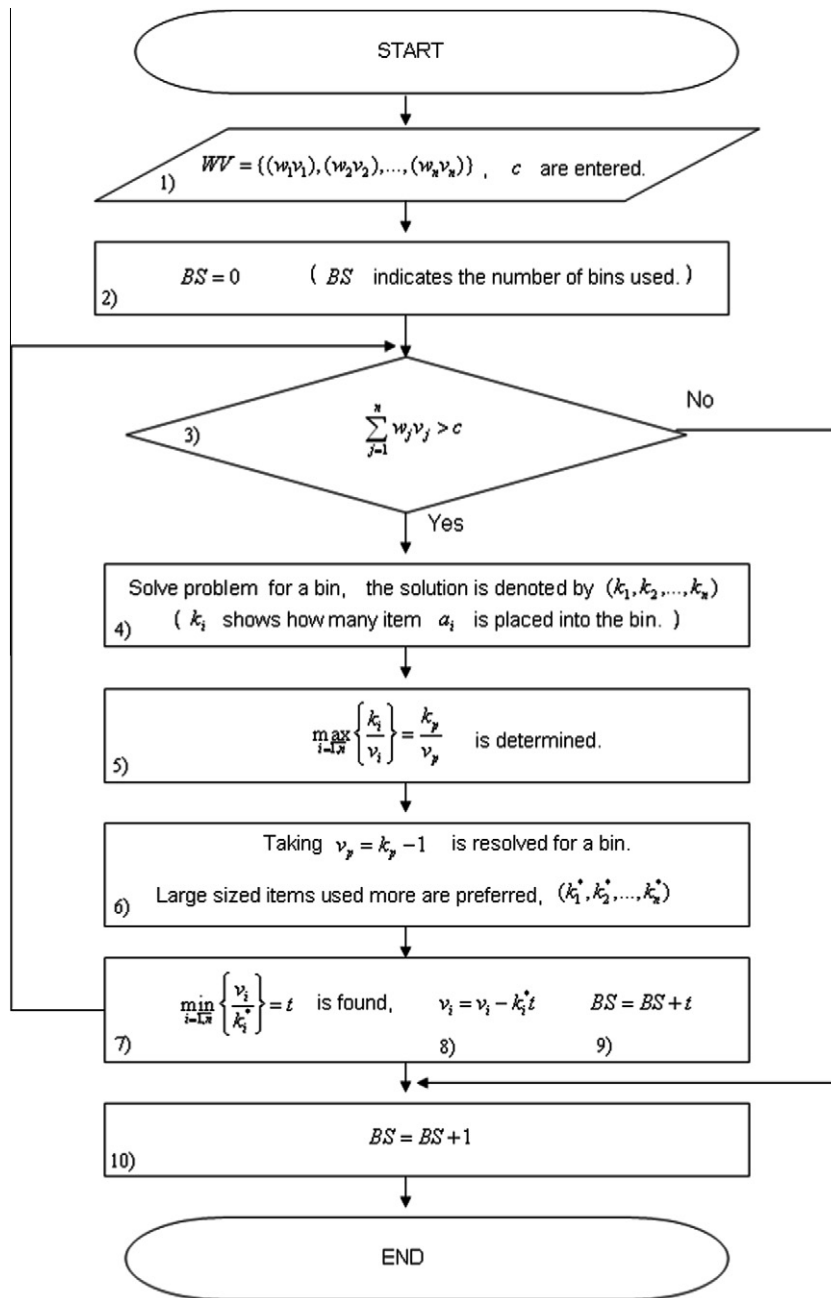


Figure 2 The flow chart of the algorithm CSP.

Table 1 Data for problem 1 and problem 2.

j	PI		PII	
	w _j	v _j	w _j	v _j
1	992	10	1304	6
2	806	20	978	12
3	771	5	843	18
4	604	12	702	8
5	496	8	661	10
6	200	30	504	4
7	120	20	440	12
8	44	18	319	4
9			230	10
10			175	8

BPP2: $BS = 0$ Number of bins is set to zero, here BS indicates the number of bins used.

BPP3: If $\sum_{j=1}^n w_j v_j \leq c \Rightarrow$ Go to BP10. (If the sum of the demands of whole items is smaller than the capacity (of a bin), then go to step BPP10.)

BPP4: Considering the data entry above, the exact solution is found by using the below dynamic programming algorithm for packing a bin.

The solution is denoted by (k_1, k_2, \dots, k_n) . Here, k_i shows how many item a_i are placed into the bin.

BPP5: $\max_{i=1..n} \left\{ \frac{k_i}{v_i} \right\} = \frac{k_p}{v_p}$ is determined. The use-rate of the most used item is determined in the solution vector which is

Table 2 (continued)

w_j	j/b_i	1	2	3	4	5	6	7	8	9	Σv_j
496	5	1	1	1	1	1	1	1	1		8
200	6	4	4	3	3	3	3	3	3	4	30
120	7	2	2	2	2	2	2	2	2	4	20
44	8	3	3	1	1	1	1	1	1	6	18
		6000	6000	5998	5998	5998	5998	5998	5998	2315	
		100.00	100.00	99.97	99.97	99.97	99.97	99.97	99.97	38.58	99.98

Table 3 Solution analysis of bbkp, A, B, C, D and E for problem 2.

w_j	j/b_i	1	2	3	4	5	6	7	8	9	10	Σv_j
<i>bbkp</i>												
1304	1	1	1	1	1	1	1					6
978	2			3	3	3	3					12
843	3							5	4	3	6	18
702	4	4	4									8
661	5			2	2	2	2			2		10
504	6	2	2									4
440	7	2	2	1	1	1	1		4			12
319	8								2	2		4
230	9							7	1	2		10
175	10							1		6	1	8
		6000	6000	6000	6000	6000	6000	6000	6000	5999	5233	
		100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.98	87.22	100.00
<i>A</i>												
1304	1	1	1	1	1					2		6
978	2					3	3	3	3			12
843	3	4	4	4	4					2		18
702	4					2	2	2	2			8
661	5	2	2	2	2						2	10
504	6									3	1	4
440	7					2	2	2	2		4	12
319	8					1	1	1	1			4
230	9					2	2	2	2		2	10
175	10									1	7	8
		5998	5998	5998	5998	5997	5997	5997	5997	5981	5271	
		99.97	99.97	99.97	99.97	99.95	99.95	99.95	99.95	99.68	87.85	99.93
<i>B</i>												
1304	1	1	1	1	1	1					1	6
978	2	2					5	5				12
843	3		3	3	3	3			2	2	2	18
702	4								4	4		8
661	5		2	2	2	2			1	1		10
504	6	4										4
440	7		1	1	1	1	2	2	1	1	2	12
319	8	1									3	4
230	9	1	1	1	1	1	1	1	1	1	1	10
175	10	1	1	1	1	1	1	1	1	1	1	8
		6000	6000	6000	6000	6000	6000	6000	6000	6000	5232	
		100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	87.20	100.00
<i>C</i>												
1304	1			2	2	2						6
978	2			2	2	2		1	2	2	1	12
843	3	1	1				4	3	3	3	3	18
702	4	2	2	1	1	1		1				8
661	5	4	4					1				10
504	6			1	1	1				1		4

(continued on next page)

Table 3 (continued)

w_j	j/b_i	1	2	3	4	5	6	7	8	9	10	Σv_j
440	7	1	1				4	1	2		3	12
319	8	1	1				2					4
230	9			1	1	1	1	3	2		1	10
175	10	2	2						1	2	1	8
		6000	6000	6000	6000	6000	6000	6000	6000	6000	5232	
		100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	87.20	100.00
<i>D</i>												
1304	1	2	2	2								6
978	2	1	1	1	3	3		3				12
843	3	1	2	2	1	1	1	1	5	4		18
702	4	1					7					8
661	5				3	3				2	2	10
504	6										4	4
440	7							5	4		3	12
319	8	2	1	1								4
230	9	1	1	1	1	1	1			1	3	10
175	10		1	1						6		8
		5999	5996	5996	5990	5990	5987	5977	5975	5974	5348	
		99.98	99.93	99.93	99.83	99.83	99.78	99.62	99.58	99.57	89.13	99.79
<i>E</i>												
1304	1	2	2	2								6
978	2	2	2	2	1	1	1	1	1	1		12
843	3				4	4	4	4			2	18
702	4	1	1	1					2	2	1	8
661	5				1	1	1	1	3	3		10
504	6	1	1	1							1	4
440	7				1	1	1	1	2	2	4	12
319	8				1	1	1	1				4
230	9	1	1	1	1	1	1	1	1	1	1	10
175	10								3	3	2	8
		6000	6000	6000	6000	6000	6000	6000	6000	6000	5232	
		100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	87.20	100.00

found at the previous step. The rates $\frac{0}{0}, \frac{0}{x}, \frac{x}{0}$ that are faced are not taken into consideration.

BPP6: This process is repeated for all the cases in which the bin can be packed completely full, the problem, taking $v_p = k_p - 1$, is resolved for a bin. Among these solutions, the ones in which the large sized items used are more preferred. Assume that the solution below is chosen eventually:

$$(k_1^*, k_2^*, \dots, k_n^*) \quad (*)$$

BPP7: $\min_{i=1, n} \left\{ \frac{v_i}{k_i^*} \right\} = t$ is found and the amount of t of bins is loaded considering solution (*).

The rates $\frac{0}{0}, \frac{0}{x}, \frac{x}{0}$ that are faced are not taken into consideration.

Table 4 Solution time for bbkp, A, B, C, D and E for problem 1 and problem 2.

	PI	PII
bbkp	51	39
A	60	40
B	135	363
C	91	64
D	702	1010
E	2150	2622

BPP8: $v_i = v_i - k_i^* t$ The demand of item i is updated by decreasing the amount of it being used.

BPP9: $BS = BS + t$, go to BPP3.

BPP10: $BS = BS + 1$ END.

The details of the algorithm are given above and the flow chart of the algorithm is seen in Fig. 2.

4. Software and computational experiments

MKA Yazılım ve Mühendislik Hizmetleri Ltd. Şti.¹ has motivated us to perform this software. MKA was founded in 2005 May and the aim of the firm is to be a specialist in engineering software. One of the computer software products of MKA is the ÇelikProIV which is used for the planar design of industrial structures, for their resolving, for preparing their quantity take-off and for their graphics. The demand of MKA is to embed the ÇelikProIV into the module CSP. The explanation (claim) of them is that the commercial packages in the market are more expensive and they take much more CPU time. Thus, the detailed problem analysis, mathematical model and the algorithm based on this model are all taken into account and

¹ İZMİR TEKNOLOJİ GELİŞTİRME BÖLGESİ İYTE Kampüsü, A3 Binası, No. 16 Gülbağçe, Urla 35430, İZMİR, Turkey. Tel.: +90 232 765 91 51; fax: +90 232 765 93 31. <http://www.mkayazilim.com.tr>

finally a software called bbkp is developed [bbkp]. Since the aim of the study is based on a modulation, the user interface was not paid much attention. Here a dynamic library DLL is given to MKA.

But, bbkp, as it stands, has been exhibited in the project markets and paid great attention by the firms of the manufacturing sector (Nuriyev et al., 2008, 2007).

The computational experiments are based on the higher difficulty degree real-life constraint optimization problems that are chosen from MKA. The two problems called PI and PII

are shown in Table 1. The length of the stick is 6000 mm for each of the two problems. The notation in Section 3 is used to understand the terms of the problems easily.

For benchmarking of the problems of MKA, we used the commercial 1D cutting stock software that is available via internet. The software packages were searched using the key word “one-dimensional cutting stock program” at www.google.com. The results were scanned to retrieve applicable software. Only five packages were chosen to have executable demo versions available. The names of the packages and their

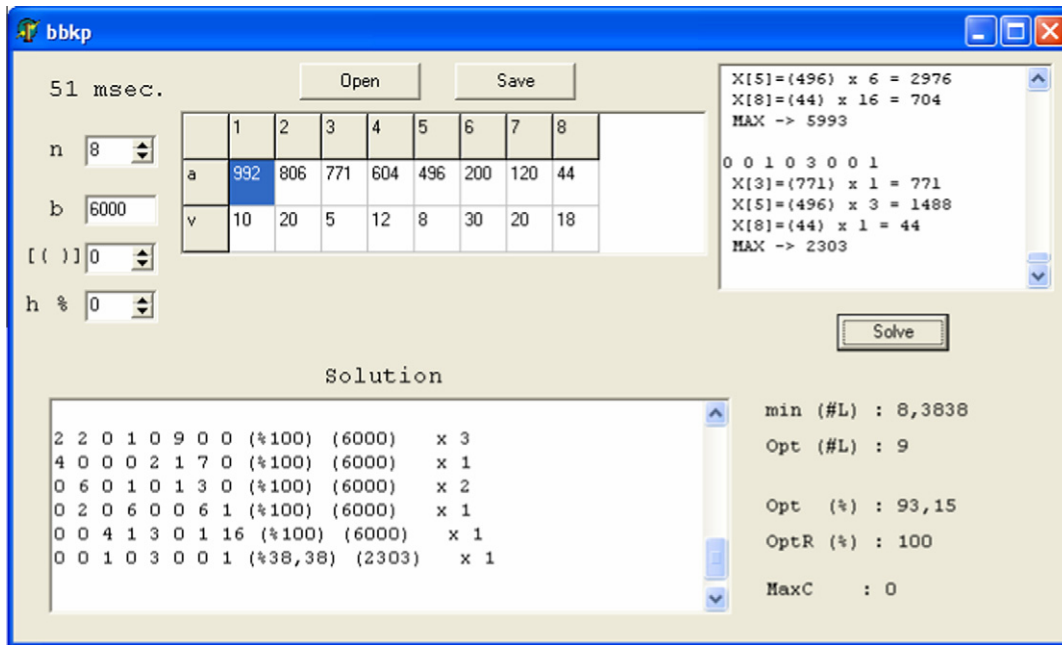


Figure 3 The screenshot of bbkp for PI.

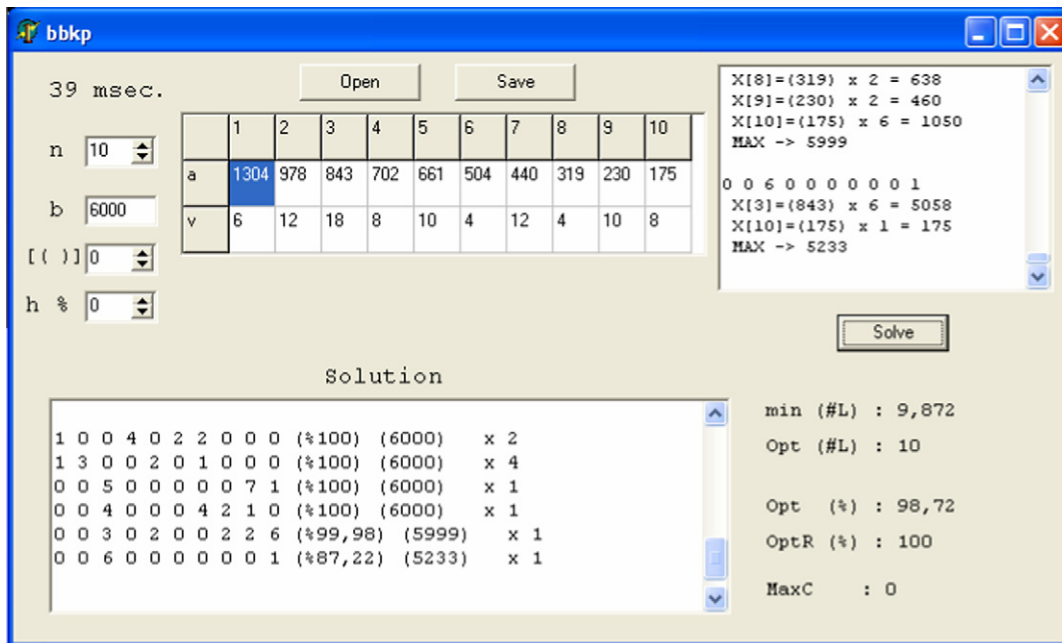


Figure 4 The screenshot of bbkp for PII.

web sites are listed at the end of the references [A], [B], [C], [D], [E].

bbkp gives optimum results and it can be seen from Tables 2 and 3 for PI and PII. On the other hand PI has optimum result for packages B and C and PII has optimum result packages B, C and E. These results show that the software is as well designed as the other commercial packages in terms of the solution quality.

In the following tables the columns indicate the bins (i) whereas the rows indicate the items (j). The first row below the tables belongs to the $\sum w_j v_j$ per an item j . The second one indicates the rate of fullness of the bin related to the column b_j . The gray cells are the optimum solutions.

Table 4 includes the comparison of the software with the other packages in terms of CPU time. As it is seen, the software has the best CPU time for each problems. When we compare the other packages of B, C and E which are equivalent related to their solution quality and the software, it is seen that the software is 1.65 times more efficient than C which is the closest competitor according to the rate of time over solution. Computational experiments run through Intel Pentium 4 CPU 2.40 GHz and 512 MB RAM devised computer.

The screenshots of the software bbkp for the problems PI and PII are shown in Figs. 3 and 4, respectively.

5. Conclusion

A new mathematical model for 1D-CSP is proposed and a new algorithm is developed relating this model. The algorithm is coded with Delphi and then by computational experiments with the real-life constraint optimization problems, and the obtained results are compared with the other 1D-CSP commercial packages. The computational experiments show the efficiency of the algorithm. For the time being, the Turkish version of this paper and the demo of the software are available through the link <<http://sci.ege.edu.tr/~math/projects/bbkp/>>.

References

- Belov, G., 2003. Problems, Models and Algorithms in One- and Two-Dimensional Cutting. Technische Universität Dresden.
- Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman & Co., San Francisco.
- Gilmore, P.C., Gomory, R.E., 1961. A linear programming approach to the cutting stock problem. Operat. Res. 9, 849–859.
- Gilmore, P.C., Gomory, R.E., 1963. A linear programming approach to the cutting stock problem, part II. Operat. Res. 11, 863–888.
- Kellerer, H., Prefschy, U., Pisinger, D., 2004. Knapsack Problems. Springer, Berlin.
- Martello, S., Toth, P., 1990. Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons.
- Nuriyev, U.G., Berberler, M.E., Gürsoy, A., Kaya, O., 2007. Preparation of a software package for one-dimensional cutting stock problem and application to the PVC, metal and public sector. “Inno-Venture 2007” Project Market, Tubitak, Ebiltem, Turkcell (Project Number BIT_208), İzmir, December 13–14.
- Nuriyev, U.G., Berberler M.E., Gürsoy A., 2008. Preparation of a software package for one-dimensional cutting stock problem and application to the PVC, metal and public sector. Project Market 2008, June 18. İzmir Institute of Technology, İzmir, p. 25.
- Yang, Chien-Tung, Sung, Tso-Chung, Weng, Wei-Chu, 2006. An improved tabu search approach with mixed objective function for one-dimensional cutting stock problems. Adv. Eng. Software 37 (8), 502–513.

Further reading

- [bbkp] (Bir Boyutlu Kesme Programı). <<http://sci.ege.edu.tr/~math/projects/bbkp/>>.
- [A] (1D Stock Cutter). <<http://www.astrokettle.com/pr1dsc.html>>.
- [B] (Optimum Cut). <<http://www.downloadjunction.com/product/software/30836/>>.
- [C] (Real Cut 1D). <<http://www.softplatz.com/Soft/Graphics/CADs/Real-Cut-1D.html>>.
- [D] (Pipe Cutting Suite 3.11). <http://www.shareup.com/Pipe_Cutting_Suite-download-7887.html>.
- [E] (Cut Logic). <<http://www.softpicks.net/software/CutLogic-1D-9970.htm>>.