Combinatorial optimization modeling approach for one-dimensional cutting stock problems

Ivan C. Mustakerov and Daniela I. Borissova

***Abstract*—**The paper describes an combinatorial optimization modeling approach to one-dimensional cutting stock problem. The investigated problem seeks to determine the optimal length of the blanks and the optimum cutting pattern of each blank to meet the requirement for a given number of elements with different lengths. Blanks of particular type are offered with equal size in large quantities and the goal is to find such optimal length of blanks that leads to minimal overall trim waste. To achieve that goal a combinatorial optimization approach is used for modeling of one-dimensional cutting stock problem. Numerical example of real-life problem is presented to illustrate the applicability of the proposed approach. It is shown that numerical example can be solved for reasonable time by Lingo Solver and MS Excel Solver.

***Keywords*—**one dimensional cutting stock problem, combinatorial optimization, optimization model, MS Excel Solver.

I. Introduction

THE cutting-stock problem has many applications in industry. This problem arises when the available material has to be cut to fulfill certain goals as cutting patterns with minimal material waste and cost efficient production, higher customer satisfaction, etc. In general, cutting stock problems consist in cutting large pieces (*blanks*), available in stock, into a set of smaller pieces (*elements*) accordingly to the given requirements, while optimizing a certain objective function.

In the last four decades cutting stock problems have been studied by an increasing number of researchers [1]-[6]. The interest in these problems is provoked by the many practical applications and the challenge they provide to researchers. On the first glance they are simple to formulate, but in the same time they are computationally difficult to solve. It could be summarized that: cutting and packing problems [7] belong to the class of NP-hard problems; solution of these problems extensively uses mathematical programming and combinatorial methods; many real-life problems are computationally hard and can be formalized only as NP-hard problems. The continuous growth of the prices of the materials and of the energy requires minimization of the production expenses for every element.

I. C. Mustakerov is with the Institute of Information and Communication Technology at the Bulgarian Academy of Sciences, Sofia – 1113, Bulgaria, Department of Information Processes and Decision Support Systems (phone: 3952 9793241; e-mail: mustakerov@iit.bas.bg).

D. I. Borissova is with the Institute of Information and Communication Technologies at Bulgarian Academy of Sciences, Sofia – 1113, Bulgaria, Department of Information Processes and Decision Support Systems (phone: 3952 9792055; e-mail: dborissova@iit.bas.bg).

Most materials used in the industry are supplied of standard forms and lengths, and direct use of such forms is most cases are impossible. They should be cut in advance to some size, expected to be optimal in the sense of trim waste. This can be done using various methods of cutting planning. The problem of optimal cutting is that different size elements have to be manufactured using blanks of single standard size. This demands development of methods for optimal cutting of source material. Cutting-stock problems can be classified by the dimensionality of the cutting as one-dimensional or two-dimensional problems.

The one-dimensional cutting stock problem (1D-CSP) is one of the crucial issues in production systems, which involve cutting processes. The classical 1D-CSP addresses the problem of cutting stock materials of length in order to satisfy the demand of smaller pieces while minimizing the overall trim loss. Industrial applications of 1D-CSP occur when cutting pipes, cables, wood and metal bars, etc. Kantorovich first formulates 1D-CSP [8], [9] and Gilmore and Gomory [10], [11] propose the first solution methodology for the cutting stock problems.

In most cases, cutting stock problem is formulated as an integer linear programming optimization problem that minimizes the total waste while satisfying the given demand [12]. In [13] a review of some linear programming formulations for the 1D-CSP and bin packing problems, both for problems with identical and non-identical large objects, is presented. It is investigated haw different ways of defining the variables and structure of the models affect the solvability of problems. Because of NP-hard nature of cutting stock problems finding an optimal solution in reasonable time is essentially difficult and often researchers turn to heuristic algorithms to deal with this kind of complex and large-sized problem [4], [14]. Some researchers look for solutions of 1D-CSP in which the non-used material in the cutting patterns may be used in the future, if large enough [5]. A two-stage decomposition approach for 1D-CSP is proposed in [15]. In the first stage is performed calculation of the total number of patterns that will be cut and generation of the cutting patterns through a heuristic procedure. On the second stage optimal cutting plan is determined. In [16] an approach to cutting stock problem is proposed where a ‘‘good’’ solution is seeking for consecutive time periods. It is adjusted to situations where useful stock remainders can be returned to the warehouse between time periods and used lately for other orders. A similar problem for wood industry is described in [17]. It is stated that cutting problems from the practice usually have its own specificity that do not allow the application of known models and solution algorithms. The difficulties in solution of cutting stock problems lead to using of approximate methods [18]. In many practical cases, proper modifications are needed or even completely new methods have to be developed on order to cope with real word requirements.

The current paper proposes an exact combinatorial optimization approach for one-dimensional cutting stock problem. A combinatorial optimization task is formulated to determine the optimal length of the blanks and optimal cutting patterns in sense of minimal waste. In contrast to other 1D-CSPs, the optimal length of the blanks and optimal cutting patterns are defined simultaneously as a result of solution of single optimization task. A proper algorithm for practical application of the proposed approach is defined and numerically tested by using real-life data. Numerical testing is performed by means of two popular solvers – Lingo and MS Excel Solver.

II. Problem Description

The blanks usually are supplied from the factory with some predetermined length. These blanks are used to cut out elements that differ in size and number that are specific for each particular project. The goal is to determine the optimal length of blanks (which are usually offered with equal size in large quantities) in order to satisfy the demand for all elements. Along with this, it is necessary to find the optimal cutting patterns that minimize the waste. The proposed approach to 1D-CSP will be explained by a real life example from the joinery manufacturing practice. It was found in [19] that the number of joinery types could be reduced to a certain number of unified modules. For example, in case of a middle size flat, these modules involve four modules: 1) Module 1 is used for 4 doors with dimensions 2200 mm x 730 mm; 2) Module 2 is used for 2 doors with dimensions 2000 mm x 650 mm; 3) Module 3 is used for 1 window with dimensions 1400 mm x 1400 mm; 4) Module 4 is used for 2 windows with dimensions 1700 mm x 2100 mm.

The problem can be described as follows: a factory that produces profiles for joinery manufacturing has to fulfill order of blanks with certain length needed to assemble a given number of joinery modules, consisting of elements with known length and number. For the sake of simplicity of the presentation only casement elements for the modules in the example above are summarized as a manufacturing order shown in Table I [1].

The length of the blanks for case of joinery is usually 6 meters. However, this is not mandatory requirement and if the order is sufficiently large it is possible to order blanks with different length than standard 6 meters – for example any length between 5 and 7 meters. So, the first step is to determine the length of the blanks which is optimal in the sense of overall trim waste. When the optimal length of blanks is determined, the next step is to define the optimal cutting patterns of joinery elements for each blank.

TABLE I
JOINERY ELEMENTS LENGTH AND DEMAND

Element *j*	**Length l_i, mm**	**Demand $k_{i,j}$**
1	l_1 = 650	4
2	l_2 = 730	8
3	l_3 = 1400	4
4	l_4 = 1700	4
5	l_5 = 2000	4
6	l_6 = 2100	4
7	l_7 = 2200	8

The problem of optimal blanks and cutting patterns determination for 1D-CSP can be approached by combinatorial optimization modeling.

III. Formulation of Optimization Model

The proposed combinatorial optimization model for 1D-CSP allows simultaneously determination of optimal length of blanks and optimal cutting patterns. To achieve this type of functionality of the model it is necessary to introduce inequalities for each of blanks. This in turn, requires the number *N* of the blanks to be known in advance. Number *N* can be calculated as overall demand of joinery elements divided by the length *L* of the blanks. On the other hand, the length *L* of the blanks will be determined after solution of the optimization task. This “recursive” property of the problem creates difficulty in the formulation of the model. To overcome this difficulty it is taken into account that length of the blanks *L* should have some value close to the standard length of 6 meters. Having this in mind, number of blanks *N* can be calculated as sum of lengths for all demanded elements divided by the length of 6 meters. The result is rounded to integer value because number *N* should have integer value. This value is used to formulate the proper optimization task as:

$$\min \rightarrow \sum_{i=1}^{N} \Delta L_i, i = 1,..., N \tag{1}$$

subject to

$$\forall i : \Delta L_i = L - x_{ij} l_j, \ j = 1,..., J \tag{2}$$

$$\forall j : \sum_{i=1}^{N} x_{ij} = k_{ij} \tag{3}$$

$$(6 - \Delta_{\min}) \leq L \leq (6 + \Delta_{\max}) \tag{4}$$

$$\forall i : \ \Delta L_i \geq 0 \tag{5}$$

$$\forall j : \ x_{ij} = \begin{cases} \textit{binary integer } 0 \textit{ or } 1, \text{ if } N \leq k_{ij} \\ \textit{integer}, \text{ otherwise} \end{cases} \tag{6}$$

where N is number of blanks; L is length of blanks; ΔL_i is waste of each blank; l_j is length of elements; x_{ij} are decision variables assigned to each element for particular blank; k_{ij} represents the demand of each element.

The objective function (1) minimizes the sum of trim loss for each blank. The optimal cutting pattern for each of the blanks is defined by decision variables x_{ij} in (2). Depending on the given particular project, the decision variables (6) could be binary integer variables or integer variables. For example, if the number of the blanks is less than the maximum demand of some element, then the decision variables x_{ij} are to be considered as integers. This statement allows the model to allocate more than one element within cutting pattern in the blank to satisfy the elements demand by relation (3). The deviation of optimal length of blanks from the standard length of 6 meters is given by the relation Δ_{min} and Δ_{max} in statement (4). The restriction (5) ensures that cutting pattern will not exceed the length of optimal blank L.

The formulated in this way combinatorial optimization task can be solved by means of any optimization solver. To demonstrate this, two different solvers as LINGO Solver and MS Excel Solver are used.

IV. Numerical Example

The applicability of the proposed one-dimensional cutting stock approach based on combinatorial optimization is illustrated using real life example based on data in Table 1.

The following steps are performed:

1) Determination of total length of all elements considering their demand $L_{sum} = 54840\ mm$;

2) Determination number of blanks N as rounded to integer result of the total elements length *54840 mm* divided by *6000 mm* as: $54840/6000 = 9.14 \Rightarrow N = 9$;

3) Setting of deviations $\Delta_{min} = \Delta_{max} = 1000$ mm;

4) Formulation of optimization task.

A. Optimization by LINGO Solver

The optimization task formulated for solving by Lingo Solver is:

$$\min\ (\Delta L_1 + \Delta L_2 + \Delta L_3 + \Delta L_4 + \Delta L_5 + \Delta L_6 + \Delta L_7 + \Delta L_8 + \Delta L_9) \qquad (7)$$

subject to:

$$\Delta L_1 = L - (x_{11}l_1 + x_{12}l_2 + x_{13}l_3 + x_{14}l_4 + x_{15}l_5 + x_{16}l_6 + x_{17}l_7) \qquad (8a)$$

$$\Delta L_2 = L - (x_{21}l_1 + x_{22}l_2 + x_{23}l_3 + x_{24}l_4 + x_{25}l_5 + x_{26}l_6 + x_{27}l_7) \qquad (8b)$$

$$\Delta L_3 = L - (x_{31}l_1 + x_{32}l_2 + x_{33}l_3 + x_{34}l_4 + x_{35}l_5 + x_{36}l_6 + x_{37}l_7) \qquad (8c)$$

$$\Delta L_4 = L - (x_{41}l_1 + x_{42}l_2 + x_{43}l_3 + x_{44}l_4 + x_{45}l_5 + x_{46}l_6 + x_{47}l_7) \qquad (8d)$$

$$\Delta L_5 = L - (x_{51}l_1 + x_{52}l_2 + x_{53}l_3 + x_{54}l_4 + x_{55}l_5 + x_{56}l_6 + x_{57}l_7) \qquad (8e)$$

$$\Delta L_6 = L - (x_{61}l_1 + x_{62}l_2 + x_{63}l_3 + x_{64}l_4 + x_{65}l_5 + x_{66}l_6 + x_{67}l_7) \qquad (8f)$$

$$\Delta L_7 = L - (x_{71}l_1 + x_{72}l_2 + x_{73}l_3 + x_{74}l_4 + x_{75}l_5 + x_{76}l_6 + x_{77}l_7) \qquad (8g)$$

$$\Delta L_8 = L - (x_{81}l_1 + x_{82}l_2 + x_{83}l_3 + x_{84}l_4 + x_{85}l_5 + x_{86}l_6 + x_{87}l_7) \qquad (8h)$$

$$\Delta L_9 = L - (x_{91}l_1 + x_{92}l_2 + x_{93}l_3 + x_{94}l_4 + x_{95}l_5 + x_{96}l_6 + x_{97}l_7) \qquad (8i)$$

$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{71} + x_{81} + x_{91} = 4 \qquad (9a)$$

$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} + x_{82} + x_{92} = 8 \qquad (9b)$$

$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} + x_{73} + x_{83} + x_{93} = 4 \qquad (9c)$$

$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} + x_{74} + x_{84} + x_{94} = 4 \qquad (9d)$$

$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} + x_{75} + x_{85} + x_{95} = 4 \qquad (9e)$$

$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{66} + x_{76} + x_{86} + x_{96} = 4 \qquad (9f)$$

$$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} + x_{67} + x_{77} + x_{87} + x_{97} = 8 \qquad (9g)$$

$$\forall i:\ \Delta L_i \geq 0\ ,\ i = 1, \ldots, 9 \qquad (10)$$

$$5000 \leq L \leq 7000 \qquad (11)$$

$$x_{ij}\ \text{– binary integer: 0 or 1} \qquad (12)$$

The relations (8) in combination with inequalities (10) define optimal cutting patterns for each particular blank. The optimal cutting patterns are defined not to exceed the length of the blanks (10) and to satisfy the requested demand of elements expressed by (9). The objective function (7) seeks for solution that minimizes the waste of all blanks. The optimal length of blanks is to be defined within interval of 5 to 7 meters (11). In this example the decision variables for optimal cutting patterns are considered as binary integer variables (12).

The solution the optimization task (7) – (12) by Lingo Solver (Table II) determines the optimal length of blanks; total waste; waste for each blank; and used length of each blank.

TABLE II
Optimal Solution Results

Optimal length of blank L, mm	Total waste for order, mm	Waste for each blank, mm	Used length of each blank, mm
		220	6330
		220	6330
		520	6030
		520	6030
6550	4110	870	5680
		870	5680
		870	5680
		20	6530
		0	6550

The optimal cutting patterns defined by the values of the binary integer variables for each blank are shown in Table III.

TABLE III
OPTIMAL CUTTING PATTERNS FOR EACH BLANK

Blank	Element1	Element2	Element3	Element4	Element5	Element6	Element7
L_1	0	1	1	0	1	0	1
L_2	0	1	1	0	1	0	1
L_3	0	1	1	1	0	0	1
L_4	0	1	1	1	0	0	1
L_5	1	1	0	0	0	1	1
L_6	1	1	0	0	0	1	1
L_7	1	1	0	0	0	1	1
L_8	0	1	0	1	1	1	0
L_9	1	0	0	1	1	0	1

B. Optimization by MS Excel Solver

Microsoft Excel is part of the popular MS Office package. It has an add-in module for optimization problems solving. The MS Office Excel has the advantage to be widespread and widely used software and is well known general-purpose optimization modeling system. Because of the specifics of spreadsheet tables, it is easy to create models that contain explanatory texts and sometimes are more useful than other modeling languages such as GAMS and AMPl. In addition there are many useful Excel functions for statistical and mathematical calculation to express a wide range of mathematical relationships [20]. There is also other third party add-ins that expands the capabilities of Excel. For example, Lindo's What's Best add-in combines the power of building of large-scale optimization models in a free-form layout within a spreadsheet [21].

By combining graphical user interface with algebraic modeling language and optimizers implementing different algorithms for linear, nonlinear, and integer problems solving, the Microsoft Excel Solver can be good choice for many users [22]. To illustrate this, the described approach to one-dimensional cutting stock problem is implemented also as spreadsheet optimization model in MS Excel environment

Solving the linear program (7) – (12) in Excel requires creating a spreadsheet which describes the problem. It includes: specifying the cell which contains the objective function; specifying the decision variables; specifying the cells which define the constraints; solving the model. For the goal *Solver Parameters* dialog box is used to enter the optimization problem.

Before activating the solution process by button *Solve* it is very important to set *Options* for the solving. These options are essential for adjustment of the solution process and for solution time.

The optimal solution obtained by MS Excel Solver is shown in Fig. 1. As it is expected the solution coincides with Lindo solution but the solution time is greater than Lindo solution time – approximately 2 and a half hour versus 1 and a half hour.

	A	B	C	D	E	F	G	H	I	J	K
1		*Combinatorial optimization model for one-dimensional cutting stock problem*									
2		*l1, l2, ..., l7: length of the elements that are to be cut*									
3		*ΔLi: waste for each blank = optimal length of blank - utilized length of each blank*									
4		*Overall waste = sum of waste of all blanks*									
5		*Lmin, Lmax: lower and upper limits of optimal length of blank*									
6											
7		*Lmin* =	5000	*l1*	*l2*	*l3*	*l4*	*l5*	*l6*	*l7*	Optimal lenght of blank,
8		*Lmax* =	7000	650	730	1400	1700	2000	2100	2200	**6550**
9		Demand of elements		4	8	4	4	4	4	8	Waste of each blank, *ΔLi*
10		Cutting patterns for each blank		0	1	1	0	1	0	1	220
11				0	1	1	0	1	0	1	220
12				0	1	1	1	0	0	1	520
13				0	1	1	1	0	0	1	520
14				1	1	0	0	0	1	1	870
15				1	1	0	0	0	1	1	870
16				1	1	0	0	0	1	1	870
17				0	1	0	1	1	1	0	20
18				1	0	0	1	1	0	1	0
19										Minimal waste:	**4110**
20		Elements # in solution:		**4**	**8**	**4**	**4**	**4**	**4**	**8**	

Fig. 1. Optimal cutting stock solution

V. Result Analysis and Discussion

The defined optimal length of blanks to fulfill the order is 6550 mm and the overall minimum waste is 4110 mm. The graphical illustration of optimal cutting patterns for each of the blanks is shown in Fig. 2.

The proposed optimization approach determines the optimal length of blanks that is increased toward standard length with 550 mm. This reduces number of needed blanks to fulfill the requested order and waste and costs as compared to the case of standard length using.

Using of standard length of 6 m not only increases the trim loss but also increases the number of required blanks to execute the order.

Due to NP-hard nature of one-dimensional cutting stock problems, the computational time increases essentially with increasing the number of decision variables. Formulated optimization tasks are solved on PC with 2.93 GHz Intel i3 CPU, 4 GB RAM and MS Windows OS.

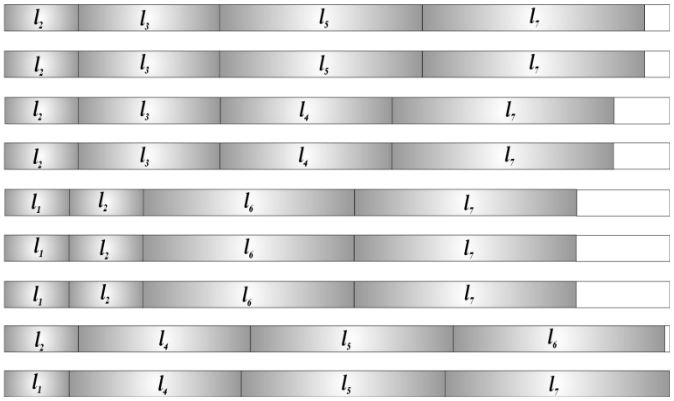

Fig. 2. Optimal cutting patterns for blanks
(L = 6550 mm, waste = 4110 mm)

The task solution report of Lingo is shown in Fig. 3.

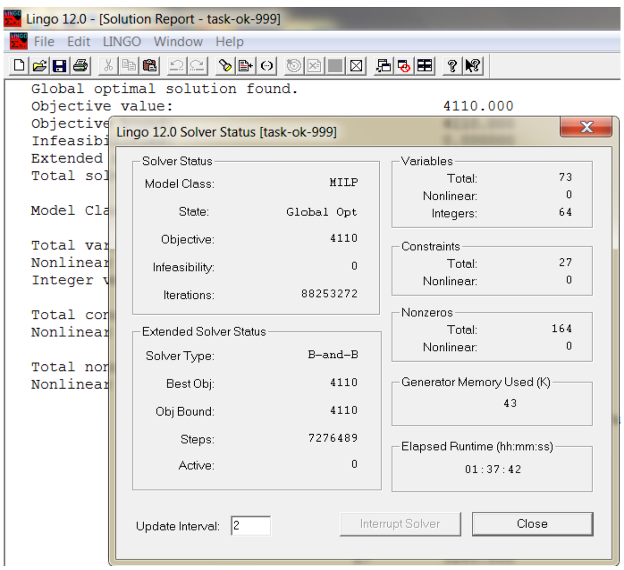

Fig. 3. Task solution report in Lingo environment

On the same computer the Excel Solver solution time is about 2.5 hours. This solution time depends of the computer load and of settings in window *Options* of the Solver. These computational times are quite acceptable having in mind that this is not case of real time optimization.

VI. Conclusion

In the paper, one-dimensional cutting stock problem is modeled by means of combinatorial optimization. The advantage of the proposed approach is the possibility to determine simultaneously the optimal length of the blanks and optimal cutting patterns for each blank. In contrast to heuristic approaches to this type of problems the described approach defines solution as a global optimum.

The reduction of cutting trim loss is one of the main problems in many manufacturing processes. It is very important especially for big projects when large numbers of elements are needed. Due to NP hard nature of the cutting stock problems, computational difficulties increase exponentially with dimensions of the problems.

Future investigations are to be done with different large scale problems to reduce computational times. One possible approach is to use decomposition and parallelization of the formulated by this approach optimization tasks. Implementation of the proposed approach in a software tool for planning and design will help the practitioners to reduce costs thus contributing to their competitiveness.

References

[1] I. Mustakerov, D. Borissova. One-dimensional cutting stock model for joinery manufacturing. *In Proc. Advanced Information Science and Applications – Volume I, 18th Int. Conf. on Circuits, Systems, Communications and Computers* (CSCC 2014), July 17-21, 2014, Santorini Island, Greece, pp. 51-55.

[2] A. Mobasher and A. Ekici, “Solution approaches for the cutting stock problem with setup cost”. *Computers & Operations Research*, vol. 40, 2013, pp. 225-235.

[3] A. C. Dikili, E. Sarioz and N. A. Pek, “A successive elimination method for one-dimensional stock cutting problems in ship production”. *Ocean Engineering*, vol. 34, 2007, pp. 1841-1849.

[4] Y. Cui and Y. Lu, “Heuristic algorithm for a cutting stock problem in the steel bridge construction”. *Computers & Operations Research*, vol. 36, 2009, pp. 612-622.

[5] C. Cherri, M. N. Arenales and H. H. Yanasse, “The one-dimensional cutting stock problem with usable leftover – A heuristic approach”. *European Journal of Operational Research*, vol. 196, 2009, pp. 897-908.

[6] A. C. Dikili, A. C. Takinaci and N. A. Pek, “A new heuristic approach to one-dimensional stock-cutting problems”, *Ocean Engineering*, vol. 35, no. 7, 2008, pp. 637-645.

[7] E. A. Mukhacheva and A. S. Mukhacheva. “L. V. Kantorovich and Cutting-packing problems: New approaches to combinatorial problems of linear cutting and rectangular packing”. *Journal of Mathematical Sciences*, vol. 133, no. 4, 2006, pp. 1504-1512.

[8] L. V. Kantorovich, Mathematical methods of organizing and planningproduction. *Management Science*, vol. 6, 1960, pp. 366-422.

[9] L. V. Kantorovich and V. A. Zalgaller, *Rational Cutting of Stock* [in Russian], Nauka, Novosibirsk, 1971.

[10] P. Gilmore and R. Gomory, “A linear programming approach to the cutting stock problem”. *Operations Research*, vol. 9, no. 6, 1961, pp. 848-859.

[11] P. Gilmore and R. Gomory, “A linear programming approach to the cutting stock problem, part II”. *Operations Research*, vol. 11, 1963, pp. 863-888.

[12] S. M. A. Suliman, “Pattern generating procedure for the cutting stock problem”. *Int. Journal of Production Economics*, vol. 74, 2001, pp. 293-301.

[13] J. M. Valerio de Carvalho, “LP models for bin packing and cutting stock problems”. *European Journal of Operational Research*, vol. 141, 2002, pp. 253-273.

[14] M. HMA Jahromi, R. Tavakkoli-Moghaddam, A. Makui and A. Shamsi, “Solving an one-dimensional cutting stock problem by simulated annealing and tabu search”. *Journal of Industrial Engineering International*, vol. 8, no. 24, 2012, doi:10.1186/2251-712X-8-24.

[15] T. Aktin and R. G. Ozdemir. “An integrated approach to the one-dimensional cutting stock problem in coronary stent manufacturing”. *European Journal of Operational Research*, vol. 196, 2009, pp. 737-743.

[16] P. Trkman and M. Gradisar. “One-dimensional cutting stock optimization in consecutive time periods”, *European Journal of Operational Research*, vol. 179 , 2007, pp. 291-301.

[17] S. Koch, S. Konig and G. Wascher. “Linear Programming for a Cutting Problem in the Wood Processing Industry – A Case Study”. *FEMM Working Paper* no. 14, 2008.

[18] M. M. Malik, J. H. Taplin and M. Qiu. “Variants of the cutting stock problem and the solution methods”. *Int. Journal of Economics and Finance Studies*, 2013, vol 5, no. 2, pp. 45-54.

[19] Ch. Korsemov, Hr. Toshev, I. Mustakerov, D. Borissova and V. Grigorova. "An optimal approach to design of joinery for renovation of panel buildings". *International Journal of Science and Engineering Investigations*, vol. 2, no. 18, 2013, pp. 123-128.

[20] D. Borissova. "Optimal scheduling for dependent details processing using MS Excel Solver". *Cybernetics and Information Technologies*, vol. 8, no.2, 2008, pp.102-111.

[21] Lindo Systems, http://www.lindo.com

[22] A. J. Mason and I. Dunning. "OpenSolver: Open Source Optimisation for Excel". *In Proc.of the 45th Annual Conference of the Operations Research Society of New Zealand*, November 2010, pp. 181-190.

Dr. Ivan C. Mustakerov is Associated Professor in the Instutute of ICT at Bulgarian Academy of Sciences, department of Information Processes and Decision Support Systems.
Office address: Bulgaria, Sofia – 1113, Acad. G. Bonchev St., Block 2, room 509.
E-mail: mustakerov@iit.bas.bg.
Major fields of scientific research: operations research, systems modelling and optimization, decision support systems, engineering systems design, software systems for information processing, e-learning and Web-based applications.

Dr. Daniela I. Borissova is Associated Professor in the Instutute of ICT at Bulgarian Academy of Sciences, department of Information Processes and Decision Support Systems.
Office address: Bulgaria, Sofia – 1113, Acad. G. Bonchev St., Block 2, room 515.
E-mail: dborissova@iit.bas.bg.
Major fields of scientific research: decision support systems, modeling and optimization, night vision devices, software systems for information processing, Web-based applications.