Journal of Industrial Engineering International
April 2007, Vol. 3, No. 4, 51-58

Islamic Azad University, South Tehran Branch

The trim loss concentration in one-dimensional cutting stock problem (1D-CSP) by defining a virtual cost

H. Javanshir*

Assistant Professor, Dep. of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

M. Shadalooee

M.Sc., Dep. of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

Abstract

Nowadays, One-Dimensional Cutting Stock Problem (1D-CSP) is used in many industrial processes and recently has been considered as one of the most important research topic. In this paper, a metaheuristic algorithm based on the Simulated Annealing (SA) method is represented to minimize the trim loss and also to focus the trim loss on the minimum number of large objects. In this method, the 1D-CSP is taken into account as Item-oriented and the authors have tried to minimize the trim loss concentration by using the simulated annealing algorithm and also defining a virtual cost for the trim loss of each stock. The solved sample problems show the ability of this algorithm to solve the 1D-CSP in many cases.

Keywords: One-dimensional cutting stock problem; Simulated Annealing; Trim loss concentration; Item-oriented; FDD algorithm; Virtual cost

1. Introduction

The one-dimensional cutting stock problem (1D-CSP) is used in many industrial processes [7,15,16] and recently has been considered as one of the most important research topic [1,3,21]. But most of the researches have focused on the problem which contains large objects by the same size and the same or a bit different with respect to standard lengths.

Most of the problems related to the 1D-CSP have been known as NP-Complete problems [2,8,13]. Although in many cases, these kinds of problems can be modeled by mathematical programming methods, their solutions could be found by the exact or approximate methods.

The purpose of this paper is presenting a method for solving the 1D-CSP, with specific number of large objects with different lengths and divides them to many items of relatively few small items in a way that the trim loss becomes minimized and also qualified to be used in future cutting plans. So, they must be focused on the minimum number of large objects with the largest possible lengths. According to the Dyckhoff typology [4], the CSP presented in this paper, when the number of the large objects is enough, can be classified as 1/V/D/R.

1, in this classification, shows the number of dimension of this problem. *V* means that all of the small items are a selection of large objects. *D*, means that there are many small items of relatively few dimensions that must be cut from large objects.

1D-SCP is mostly used in real world and the real problems are usually large and hard to solve with optimization methods. Many scientists have solved this problem by using heuristic and metaheuristic algorithms like Simulate Annealing, Genetic Algorithm, Ant Colony Optimization etc. in different cases. [5,6,17]

Dyckhoff also classifies the solution of CSP in two groups: Item-oriented and Pattern-oriented.

Item-oriented solution is characterized by individual treatment of every item to be cut. In the pattern-oriented solution, at first, ordered lengths are combined into cutting patterns and then the frequencies of the pattern that are necessary to satisfy the demands are organized based on an algorithm prepared with Gilmore and Gommory [9,10,11,12].

* *Corresponding author. E-mail: hjavanshir@yahoo.com*

Because the pattern-oriented method can be used only when the large objects have the same lengths or they are used in different standard lengths, and the item-oriented method is used when there is not any constant pattern for cutting process, according to the type of the solution method used for this problem, the authors of this paper have chosen the item-oriented solution method to be discussed.

2. 1D-CSP problem

In most industries, the cost of the materials consists of a high percent of total price. (More than 80%)

The CSP is one of the most famous operation research problems which is defined to improve the quality of using materials. This problem can be described as follow:

There is a collection of stock material object that we are going to divide them into smaller pieces of desired lengths in order to satisfy specific customer demands by facing the minimum trim loss or using the minimum number of stocks.

The very first formulation for CSP was produced for one dimensional CSP by Kantorvich in 1939 [19] In 1960`s, Gilmore and Gomory published their four popular papers for 1 and 2 dimensional CSP [9,10, 11,12]. Their first paper was published in 1961 and it was about using the linear programming to solve the 1D-CSP. This paper was a real start to introduce the techniques which are used to solve the real world problems.

CSP was developed during the development of using computers in solving the OR problems.

There are many definitions for 1D-CSP. In this problem, there are specific numbers of large objects and we want to satisfy the order list which is equal with the stocks in 2 dimensions so that the trim loss becomes minimized.

The dimension in CSP is the degree of freedom in decision making. If 2 dimensions of the large objects and small items be equal, the decision is made about the way of cutting the third dimension. So this cutting procedure is called one-dimensional.

The basic goal in solving the CSP is to minimize the trim loss. Of course, sometimes other parameters like the time or the cost of changing the fixations are applied In goal function [22]. In this paper, focusing the trim loss on minimum number of large objects is applied including minimizing the trim loss.

Gradisar, in his paper in 2002, made an evaluation between the 1D-CSP algorithms which was the main reference for the authors of this paper [14].

He presented 3 specifications for a suitable CSP model:

1. Ability to cut order lengths in exactly required number of pieces,
2. Ability to cumulate consecutive residual lengths in one piece which could be used later,
3. Ability to use non -standard stock lengths.

3. Trim loss definition

In CSP the final objective is to minimize the trim loss which is usually defined as minimizing the number of cutting patterns. But totally the trim loss on each stock can be formulated in according to this equation:

The trim loss appeared on a large object = the length of the large object – the sum of small items lengths cut from a large object.

This equation can be applied in both item-oriented and pattern-oriented methods in 1D-CSP. In item-oriented CSP solution, it is possible to model the trim loss according to these parameters:

n_k The number of the ordered small items by the length of l_k.

N The number of all small items.

l_i The length of the i^{th} ordered item.

M The number of all large objects.

L_j The length of the j^{th} large object.

w_j The trim loss appeared on j^{th} large object.

X_{ij} A binary variable which is equal to one if the i^{th} Item is cut from the j^{th} large object and zero otherwise.

The trim loss model is as follows:

$$w_j = L_j - \left(\sum_{i=1}^{N} l_i x_{ij}\right)$$

$$0 \le w_j \le L_j \qquad j=1,2,\ldots,M. \qquad (1)$$

The goal of the above equation is to present an algorithm in order to perform an item-oriented CSP solution while the trim loss is focused on the minimum number of stocks.

By changing the values of X_{ij}, the different states of the cutting patterns by different trim loss values would be appeared on a large object.

It must be mentioned that the different cutting patterns are created by using the FFD (Firs Fit Decreasing) algorithm.

4. FFD algorithm

In according to the FFD algorithm, if the length of the i^{th} small item is smaller than the remaining length j^{th} large object, the i^{th} item will be cut from the j^{th} large object $(X_{ij} = 1)$. Otherwise, the cutting process is applied on the next large object if possible $(X_{ij} = 0)$ [18]. And it is continued until the i^{th} item is cut. Here is the pseudo code of this algorithm:

For *i=1* to *N*
  For *j=1* to *M*
    If $l_i < L_j$ then
      $L_j = L_j - l_i$
      Go to NEXTI
    Else
      Go to NEXTJ
    End if
  NEXTJ: next *j*
NEXTI: next *i.*

In this algorithm, the order of cutting small items is important. By changing the order of these items, the trim loss concentration (and sometimes the trim loss value) will be also changed. Here is an example which defines the effect of small items cutting order on the trim loss concentration.

Example. It is assumed that the following table contains the customer order of the small items and the length of the stocks is 12 meters.

Table 1. The demand of small items.

Number	Length
11	2m
4	3m
4	5m

L =12 m

In Table 2, all of the items in the item list column are sorted ascending in according to their length. If one of the items can be cut from the remaining stock length, the new stock length after cutting should be written in the related column and if the cutting process could not be done on the remaining length of the previous stocks, a new stock is used. The trim loss of each stock is mentioned in the last row.

As it could be seen, the total trim loss of the Table 2 is 6 meter which is appeared on the stocks number 2,4 and 5. In Table 3, the order of the items will be changed and the table contents will be updated again.

It is obvious that by changing the order of items, the total trim loss will be obtained equal to 6 meters which is reduced from 3 stocks to only the stock #5. Therefore it is expected that, by using the FFD algorithm in different order of items the better concentration of trim loss will be achieved.

Among the different possible concentration of trim loss produced by changing the order of items, those order of items which have the better concentration of trim loss, should be searched. If the trim loss appeared on the least number of stocks with the longer length, the concentration of trim loss will be achieved to its desirable level and therefore these stocks will be used in the future cutting orders as input large objects.

Table 2. Cutting according to FFD algorithm in primary order of small items (in example).

Item list	**1**	**2**	**3**	**4**	**5**	**Stock no.**
	12	**12**	**12**	**12**	**12**	**Initial stock length**
2	10					Sum of trim loss
2	8					
2	6					
2	4					
2	2					
2	0					
2	0	10				
2	0	8				
2	0	6				
2	0	4				
2	0	2				
3	0	2	9			
3	0	2	6			
3	0	2	3			
3	0	2	0			
5	0	2	0	7		
5	0	2	0	2		
5	0	2	0	2	7	
5	0	2	0	2	2	
Trim loss	**0**	**2m**	**0**	**2m**	**2m**	**6m**

Table 3. Cutting according to FFD algorithm in Secondary order of small items (in example).

Item list	1	2	3	4	5	Stock no.
	12	12	12	12	12	Initial stock length
5	7					Sum of trim loss
5	2					
5	2	7				
5	2	2				
3	2	2	9			
3	2	2	6			
3	2	2	3			
3	2	2	0			
2	0	2	0			
2	0	0	0			
2	0	0	0	10		
2	0	0	0	8		
2	0	0	0	6		
2	0	0	0	4		
2	0	0	0	2		
2	0	0	0	0		
2	0	0	0	0	10	
2	0	0	0	0	8	
2	0	0	0	0	6	
Trim loss	**0**	**0**	**0**	**0**	**6m**	**6m**

In order to compare the different states of trim loss concentration and identify the better concentration, it is essential to define a parameter, explained below, as a comparison criterion.

5. Cutting virtual cost

It can be predicted that by considering a virtual cost and trying to minimize it, we will be able to obtain a better trim loss concentration by using a local search algorithm.

The virtual cost which is defined for each stock is a variable value which has the ascending trend. It is possible to use the natural series of number with a linear-ascending trend (from 1 to M) as a virtual cost. in order to assign the virtual cost to each stock, after using the FFD algorithm, they must be sorted in descend order in according to their own trim loss, and then the least virtual cost is assigned to the stock which has the most trim loss.

Finally the total virtual cost (TVC), will be computed as below:

$$TVC = \sum_{j=1}^{M} (VC_j w_j)$$

$$VC_j = j \qquad j=1,2,\ldots,M, \qquad (2)$$

where VC_j is the virtual cost assigned to the j^{th} stock and *TVC* is total virtual cost.

In other words, always the goal of search procedure is to find those solutions which use the entire stock. In this situation the amount of VC_j coefficient effect on total virtual cost computing *(TVC)* will be reduced.

By considering the previous example, after sorting the trim loss in descending order, their concentration in the first state will be achieved (Table 4).

After sorting the trim loss in descending order, their concentration in the second state will be achieved as below (Table 5).

It is obvious that the total virtual cost of the second state is lower than that of the first state. Therefore, it is possible to use the total virtual cost as a comparison criterion to distinguish the better concentration of trim loss.

In order to present another reason for this topic, the assignment of virtual cost will be generally defined:

Assume that we have *N* items which must be cut from *M* large items. The trim loss of this cutting procedure is sorted according to Table 6. So, *TVC* is calculated as follows:

$$TVC = \sum_{j=1}^{M} (VC_j w_j) \qquad w_1 > w_2 > w_3 > \ldots > w_M. \quad (3)$$

Table 4. Assuming virtual cost to the large objects.

Stock#	1	2	3	4	5
VC	1	2	3	4	5
w_j	2	2	2	0	0

TVC = 2*1+2*2+2*3+0*4+0*5= 12

Table 5. Assuming virtual cost to the large objects.

Stock#	1	2	3	4	5
VC	1	2	3	4	5
w_j	6	0	0	0	0

TVC = 6*1+0*2+0*3+0*4+0*5= 6

Table 6. Sorting trim loss on each large object and assignment of virtual cost.

Stock#	1	2	3	.	M
VC	1	2	3	.	M
w_j	w_1	w_2	w_3	.	w_M

Assume that in an order of small items, the amount of the first item trim loss (w_1) is increased x units. It means that an item (or some of the items) is cut from the first item instead of other items like the k^{th} item. Therefore the k^{th} item trim loss is decreased x units. The picture below shows the changing process of *TVC*.

It can be seen from Figure1, increasing the value of w_1 in extend of x units (causes the more concentration of trim loss), the value of *TVC* will be decreased in extent of *x(k-1)* units. Figure 2 totally demonstrates this subject.

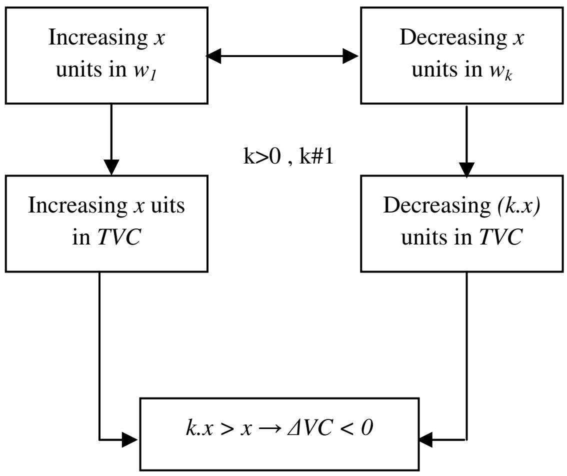

Figure 1. Changing process of *TVC*.

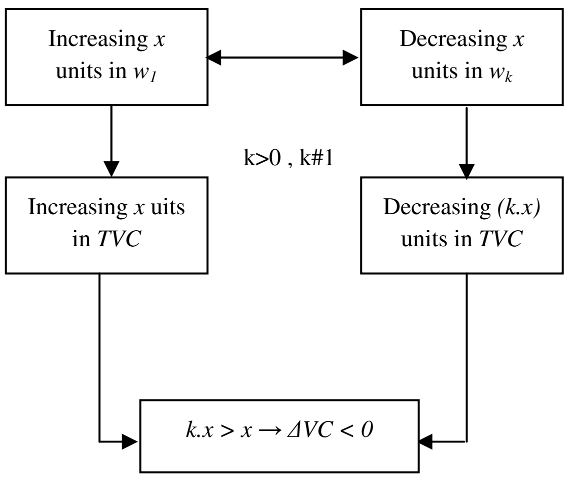

Figure 2. Changing process of *TVC*.

By considering the *TVC* as a comparison criterion for the different orders of cutting items, we are able to follow the search procedure and to achieve the better trim loss concentration.

Because the cutting problem is NP-Complete, especially in item-oriented method, it is essential to use a metaheuristic algorithm to solve the problem. In these special problems which are known as combinatorial optimization problems, one of the most useful metaheuristic algorithms is the simulated annealing.

6. Simulated annealing solution

Simulated Annealing (SA) algorithm is still being used to solve the general problems of cutting trim loss [2,6]. In this paper we try to extend the SA to solve cutting problem trim loss concentration.

6.1. Objective function definition

In this paper the objective function is defined as to minimize the *TVC* value:

$$\text{Min} \sum_{j=1}^{M} VC_j(L_j - (\sum_{i=1}^{N} l_i x_{ij})). \qquad (4)$$

At first, an N-Dimensional array, contents the ordered small items is produced. And also an M-Dimensional array, contains the length of the large objects is considered. The value of *M* can not be exactly determined initially. Therefore by the special manner of using *FFD* algorithm to determine the value of *M*, the required number of large items will be achieved. According to this change, when a small item can not be cut from the available large items, the value of M will be increased 1 unit and this procedure will be continued up to the end of cutting all of the items.

By the end of cutting process and determining the status and the amount of the trim loss on each stock, the stocks are sorted according to their own trim loss and then the virtual costs will be assigned to each stock. Therefore *TVC* will be obtained.

6.2. Initial feasible solution generation

After sorting the length of items in descending order, the *FFD* algorithm will be run in order to obtain the initial solution. Because in this process, the larger items are initially cut from the larger stocks, the above solution is usually suitable and we can consider as a desired initial solution.

6.3. Neighborhood generation

In order to generate a neighborhood solution, a random state will be used. In this state, by the use of random selection, the item orders are changed algorithmically. The steps of the above algorithm are explained below:

Step 1. Numerate the available small items in initial solution from 1 to *N*.

Step 2. Generate a Uniform Random Number in the range of [2,N] (*R1*) which defines the number of displacement of items.

Step 3. Generate *R1* uniform random numbers in the range of [1,N] *(R(i): i=1,…, N)* which defines the displaced items number.

Step 4. If the i^{th} and $(i+1)^{th}$ item have the same length, avoid displacement, otherwise, displace the i^{th} and $(i+1)^{th}$ items.

6.4. Simulated Annealing procedure

After determining the neighborhood solution, the amount of Boltzman function will be computed as below.

$$\Delta = TVC_1 - TVC$$

$$Boltzman = e^{-\frac{\Delta}{T}}. \tag{5}$$

So, the steps of the algorithm are shown below:

Step 1. Generate an initial solution.

Step 2. Calculate the TVC for the initial solution.

Step 3. Generate a neighborhood for initial solution and calculate the TVC_1 for that.

Step 4. Let $\Delta = TVC_1 - TVC$.

Step 5. If $\Delta < 0$, then $TVC = TVC_1$, go to Step 3. Else, $TVC = TVC_1$ with the Boltzman probability.

Pay attention that a proper solution must never be omitted [20]. Therefore to have a proper final solution and not to miss the proper solutions during the algorithm performance, the authors have considered a register memory which always keeps the best solution and finally presents it as the final solution.

By considering the diagram of searched points in the solved sample problems, it can be seen that the extensive range of solution area in high temperatures shown in Figure 3, is searched by SA. Moreover the SA algorithm is able to escape from local optima. In lower temperatures, the searching procedure in inclined to the near optima points and in final points it can be easily seen that the method inclination is to the optimum solution.

7. Computational results

For analyzing the computational results, 36 different random problems were generated by computer. These problems were modeled by operation research formulation of 1D-CSP and solved by Lingo-6 software. Then the problems were solved by Simulated Annealing method presented in this paper and the quality of the solutions was compared. In Table 7, there are some information about the solution results for each problem, like the trim loss and the number of large objects which contain trim loss. In most solutions of SA method, the quality of trim loss concentration were better than the OR results.

It shows the ability of the virtual cost technique to find the better trim loss concentration in CSP which can guide to reusing the trim loss in future cutting plans. The detail of the computational results is shown in Table 7.

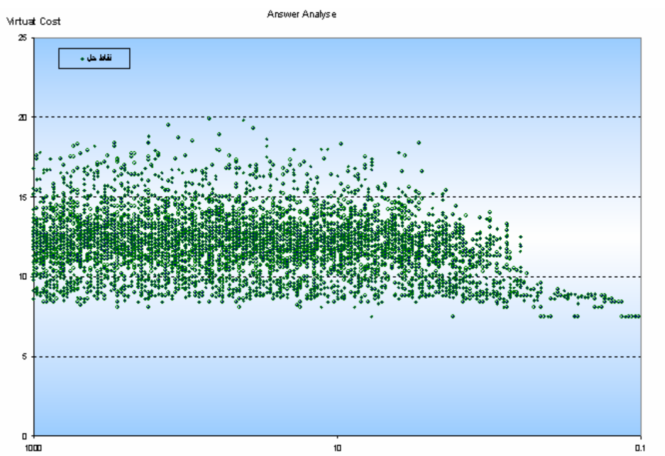

Figure 3. SA inclining to the near optimal solution.

Table 7. Solved problem information.

#	# Small items	# Stocks	Trim loss in SA	Trim loss focused stocks in SA	Trim loss in Lingo	Trim loss focused stocks in Lingo	#	# Small items	# Stocks	Trim loss in SA	Trim loss focused stocks in SA	Trim loss in Lingo	Trim loss focused stocks in Lingo
1	50	10	2	1	2	7	19	450	83	2	2	2	5
2	100	20	4	1	4	11	20	50	11	1.5	1	1.5	1
3	125	25	9	2	9	6	21	100	22	5	1	5	2
4	150	31	7	4	7	6	22	150	35	7	2	7	3
5	200	46	10	1	10	2	23	200	50	2.5	1	2.5	3
6	250	48	7.2	1	7.2	2	24	250	62	1.5	1	1.5	3
7	300	56	5	2	5	3	25	300	76	8.5	1	8.5	4
8	350	67	6	2	6	2	26	350	88	11	1	11	4
9	400	76	9	5	9	4	27	400	103	8.5	1	8.5	5
10	490	101	0	0	0	0	28	50	12	2.4	1	2.4	4
11	50	10	5	1	5	2	29	100	21	11.8	2	11.8	8
12	100	20	10	3	10	3	30	150	31	0.8	3	0.8	2
13	150	29	7	3	7	4	31	200	52	9	8	9	13
14	200	38	9	3	9	6	32	250	63	12	8	12	12
15	250	47	6	1	6	6	33	300	75	12.6	14	12.6	17
16	300	56	3	1	3	5	34	350	86	11	14	11	14
17	350	64	8	1	8	5	35	400	100	24	27	**Timeless**	
18	400	74	9	2	9	4	36	480	119	24	23	**Timeless**	

8. Conclusion

In this paper, in order to receive to the best concentration of trim loss between the 1D-CSP solutions, the authors defined an ascending trend cost and explained how to use this cost in the problem. Using this kind of cost in cutting problem, the authors received to a special type of virtual cost which could be used as a comparing parameter between the entire CSP solutions.

Finally a method was presented to minimize the trim loss concentration in 1D-CSP by using the Simulated Annealing algorithm and using the FFD cutting simulator.

The result was tested for 36 random generated problems and the solution results were compared with the Basic model of 1D-CSP, solved by Lingo which showed the best quality of solutions in SA results.

References

[1] Bishoff, E. E. and Waesher, G., 1995, Cutting and packing. *European Journal of Operational Research,* 84, 503-505.

[2] Chen, C., Hart, S. and Tham, W., 1996, A simulated annealing heuristic for the one-dimensional cutting stock problem. *European Journal of Operational Research*, 93, 522-535.

[3] Cui, Y., 2006, Generating optimal multi-segment cutting patterns for circular blanks in the manufacturing of electric motors. *European Journal of Operational Research,* 169, 30-40.

[4] Dyckhoff, H., 1990, A typology of cutting and packing problems. *European Journal of Operational Research,* 44, 145-159.

[5] Eshghi, K. and Javanshir, H., 2005, An ACO algorithm for one-dimensional cutting stock problem. *Journal of Industrial Engineering International,* 1(1), 10-19.

[6] Faina, L., 1999, An application of simulated annealing to the cutting stock problem. *European Journal of Operational Research,* 114, 542-556.

[7] Ferreira, J. S., Neves, M. A. and Fonseca, P., 1990, A two-phase roll cutting problem. *European Journal of Operational Research,* 44, 185-196.

[8] Fowler, R. J., Paterson, M. S. and Tatimoto, S. L., 1981, Optimal packing and covering in the plane are NP-complete. *Information Processing Letters,* 12, 133-137.

[9] Gilmore, P. C. and Gomory, R. E., 1961, A linear programming approach to the cutting stock problem. *Operations Research,* 9, 849-859.

[10] Gilmore, P. C. and Gomory, R. E., 1963, A linear programming approach to the cutting stock problem: part II. *Operations Research,* 11, 863-888.

[11] Gilmore, P. C. and Gomory, R. E., 1964, Multistage cutting stock problems of two and more dimensions. *Operations Research,* 13, 94-120.

[12] Gilmore, P. C. and Gomory, R. E., 1966, The theory and computation of knapsack functions. *Operations Research,* 14, 1045-1074.

[13] Gradisar, M., Kljajic, M., Resinovic, G. and Jesenko, J., 1999, A sequential heuristic procedure for one-dimensional cutting. *European Journal of Operational Research,* 114, 557-568.

[14] Gradisar, M., Resinovic, G. and Kljajic, M., 2002, Evaluation of algorithms for one-dimensional cutting. *Computers and Operation Research,* 29, 1207-1220.

[15] Haessler, R. W. and Vonderembse, M. A., 1979, A procedure for solving the master slab cutting stock problem in the steel industry. *AIIE Transactions,* 11, 160-165.

[16] Haessler, R. W. and Sweeney, P. E., 1991, Cutting stock problems and solution procedures. *European Journal of Operational Research,* 54, 141-150.

[17] Hinterding, R., Khan L., 1995, Genetic algorithms for cutting stock problems: with and without contiguity. In: Yao X, editor. Progress in evolutionary computation. *Lecture notes in artificial intelligence,* 956, 166-86.

[18] Jaliff, D. and Dagnino, A., 1995, An object-oriented tool-kit for building CSP decision support systems. *Systems, Man and Cybernetics, Intelligent Systems for the 21st Century, IEEE International Conference*, 4, 3201-3206.

[19]Kantorovich, L. V., 1960, Mathematical methods of organizing and planning production. *Management Science,* 6, 366-422.

[20] Karelahti, J., 2002, *Solving The Cutting Stock Problem in The Steel Industry.* Helsinki University of Technology, 1-39.

[21] Rietz, J., Scheithauer, G. and Terno, J., 2002, Families of non-IRUP instances of the one-dimensional cutting stock problem. *Discrete Applied Mathematics,* 121, 229-245.

[22] Umetani, Sh., Yagiura, M. and Ibaraki, T., 2003, One-dimensional cutting stock problem to minimize the number of different patterns. *European Journal of Operational Research,* 146, 388-402.