A Framework for Optimizing Lap Splice Positions within Concrete Elements to Minimize Cutting Waste of Steel Bars

Zahra S. Moussavi Nadoushani[1], Ahmed W. Hammad[2] and Ali Akbarnezhad[3]

[1] PhD candidate, School of Civil and Environmental Engineering, UNSW Australia, Sydney, NSW 2052, Australia;
[2] PhD candidate, School of Civil and Environmental Engineering, UNSW Australia, Sydney, NSW 2052, Australia;
[3] Lecturer, School of Civil and Environmental Engineering, UNSW Australia, Sydney, NSW 2052, Australia;
Email: z.moussavinadoushani@unsw.edu.au, a.hammad@unsw.edu.au, a.akbarnezhad@unsw.edu.au

Abstract

One of the material waste streams in the construction of concrete structures is the waste generated during cutting of steel bars to their final required length. The amount of waste generated to produce the required length of steel bars can be affected considerably by the cutting patterns adopted. Waste can, therefore, be minimized by optimizing the cutting pattern to be selected. Cutting patterns are a combination of bar lengths and, thus any change in the required lengths directly affects cutting patterns and the final amount of waste. This paper presents a framework for selection of the location of lap splices, through generating all possible lapping patterns, which define the bar lengths in cutting patterns. The focus is placed on steel bars used in concrete columns as their arrangement is affected by multiple parameters that make the manual preparation of shop drawings a tedious job. The developed framework is applied to a case study involving the construction of columns for an actual 6 storey building. The estimated waste produced after the adoption of optimal lapping patterns is compared with the actual waste generated based on the project shop drawings. The results indicate that considerable reduction in the steel bars wasted is achievable by optimizing the lapping patterns.

Keywords –

Minimizing rebar cutting waste, Lap splice, Cutting pattern generation, Linear Integer Programing

1 Introduction

Minimizing the material waste generated during construction is one of the major objectives in sustainable construction due to its economic and environmental benefits [1]. The economic benefits include reductions in material procurement and waste management costs. On the other hand, environmental benefits include decreasing the need for extraction of natural resources and reducing the carbon emissions and energy use associated with waste processing [2, 3]. One of material waste streams in the construction of concrete structures is related to reinforcing steel bars waste, accounting for up to 5 and 8% of the total waste in public and private residential construction respectively [4]. This waste is generated during the cutting of steel bars from standard lengths to required lengths [5-7].

The amount of the waste generated during cutting of steel bars can be affected considerably by the selected cutting patterns and thus may be minimized by selection of optimal cutting patterns. Cutting patterns are a combination of various steel bar lengths; therefore, any change in the required lengths directly affects cutting patterns and the final amount of waste. This is where the importance of engineering judgment, in terms of choosing alternative bar arrangements leading to different required lengths, becomes apparent [6-8]. However, due to the limitations of design software and Building Information Models (BIM) in producing comprehensive bar schedules, this process is mostly done by hand without using any special calculation tools, making the process a tedious and labour-consuming job [6, 7]. As Chen and Yang pointed out, even more recent developed software such as Tekla [9] which integrate design data and automatically generate design drawings still needs the engineer's experience and judgment. In their study, Chen and Yang picked concrete beams as one of the elements with the most complicated steel bar arrangement and developed an automated framework to generate shop drawings [6]. Porwal and Hewage also presented a framework integrating BIM with optimization techniques in order to minimize trim losses in the design stage [7]. Different optimization techniques such as Linear Programming (LP) [10], Integer Programming (IP) [11], Sequential Heuristic Procedure (SHP) [12] and Genetic Algorithm (GA) [5] have been proposed by previous studies to minimize the rebar trim loss referred to as one-dimensional Cutting Stock Problem (1D-CSP). Although the framework presented by Porwal and Hewage [7] can be the base of future development of BIM software, in the meantime a more practical methodology is needed to assist structural designers, in the task of generating steel bar arrangements that will lead to a reduction of trim loss.

The focus of this study is placed on steel bars used in concrete columns as their arrangement is affected by multiple parameters such as storey height, lap length and bar size that make manual preparation of shop drawings a tedious job. Including all these parameters in a model that yields the best steel bar arrangement in terms of the minimal final waste produced is a tedious job and time consuming to perform manually. In this study, a framework aiming at finding different cutting patterns and optimizing them based on variable lengths due to various possibilities for longitudinal lap splices positioning within structural columns is developed. The model is applied to a case project involving the construction of columns for an actual 6 storey building. The estimated waste produced after the adoption of optimal cutting patterns is compared with the actual waste generated based on the project's shop drawings.

2 Methodology

The proposed framework for trim loss analysis consists of four main modules; 1) Data collection module, a data entry interface for gathering all the required information needed to calculate the bar lengths, 2) Lapping pattern generation module that yields all the lap splice positioning within the permitted intervals specified by design codes, 3) Cutting pattern generation module that generates all the possible cutting patterns and 4) Trim loss optimization module that identifies the optimal cutting patterns to minimize the steel rebar waste generated during construction of concrete structures (Figure 1). Each module is described in the following sections. As shown in Figure 1, the trim loss analysis starts once the structural design phase is completed and all data regarding column sections, including the section dimensions, steel bar size and the number of steel bars, are finalized.

2.1 Data Collection Module

The data collection module is devised to provide the analyzer module with the data required to perform the trim loss analysis. As depicted in Figure 1, the information required in the data collection module comprises the number of stories of the building, storey heights, steel bar size and number of each column section, the overlap and hook length calculated based on design codes, the slab thickness of each storey and the foundation depth. These data are imported from the structural BIM model into a database which is used by the optimization code developed in MATLAB [13].

2.2 Lapping Pattern Module

Based on the ACI code, the lap splices of lateral load resisting columns are permitted only within the center half of the column length [14]. The overlap lengths are usually shorter than the columns half-length, therefore, the location of the center of lap can vary within the interval obtained from that difference (Figure 1). The aim of the developed framework is to identify the optimal rebar arrangements that will lead to a set of cutting patterns that will minimize the trim loss of the column steel bar and thus, minimize the steel trim loss of the whole project.

The lapping pattern generation module developed using MATLAB [13] yields all possible lapping patterns, considering the allowable intervals specified in design codes. The pattern generation code starts by importing the data made available by the data collection module. Then, the feasible region for the center of lap is identified based on the length of each column (storey height) and the overlap length required for each storey. It is assumed that bars are cut into lengths that are whole numbers. Thus, the length of the feasible region, which is basically a length in centimeter, represents the number of the center of lap positions. For instance, if the difference of the allowable upper and lower bound of the center of lap of a storey is 20 cm, there will be 20 different positions for the center of lap of that storey. Due to the difference in height and required overlap length of each storey of a column, the possible numbers of positions for the center of lap tend to be different. The role of the generation module is therefore to generate all the possible combinations of the center of lap positions for different stories of a column in a building. Based on the concept of permutations with repetitions [15], the total number of combinations for each column will be the product of the number of possible overlap positions for each storey height of that column. In other words, if there are 15, 20 and 30 allowable positions for storey 1, 2 and 3 of a three storey building respectively, there will be

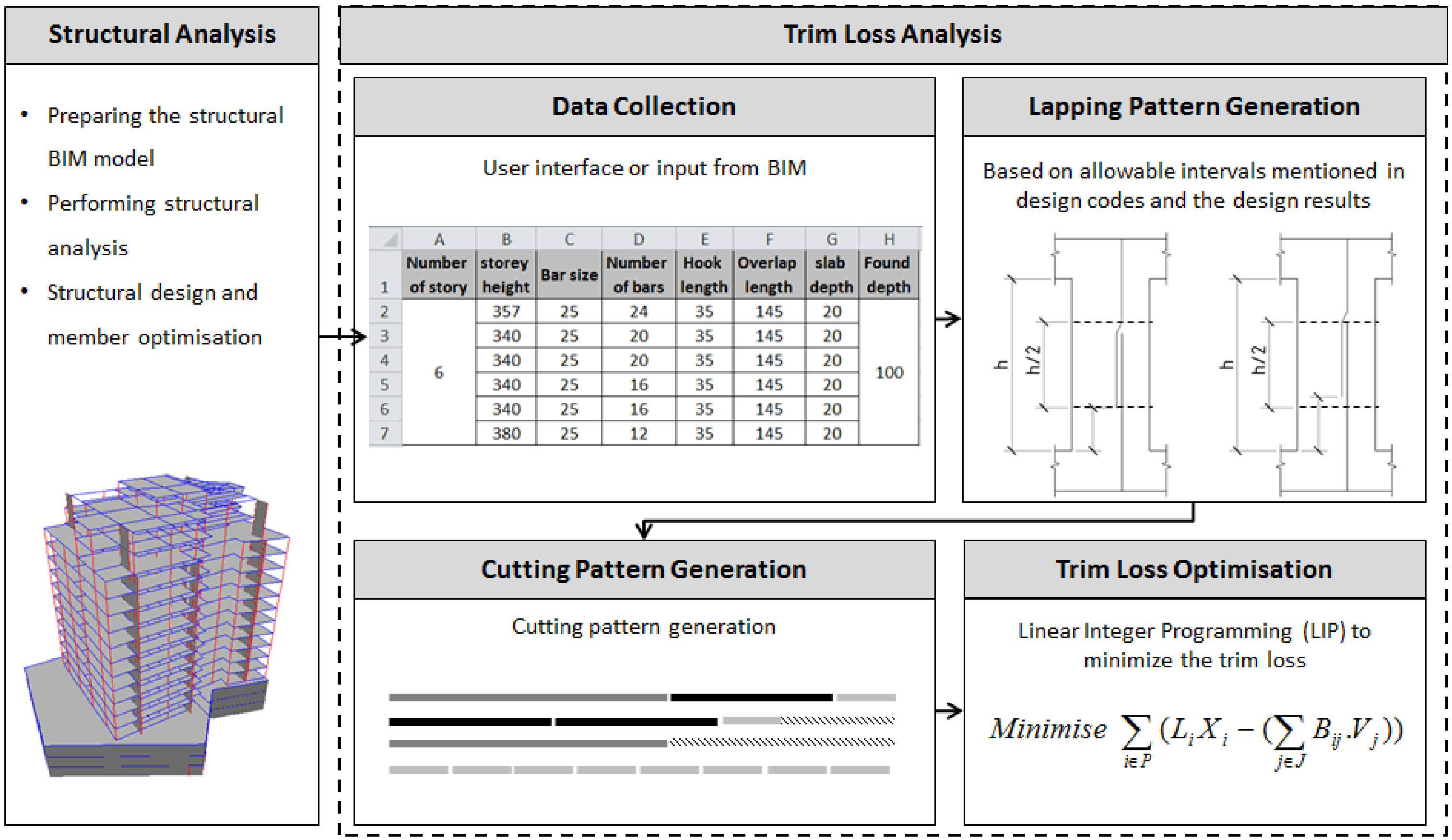

Figure 1. Framework for trim loss optimization of columns

$15 \times 20 \times 30$ equal to 9000 different arrangements for the lap splices of the columns of that building. The number of different arrangements for this simple example is indicative of the time and effort required for manual generation of all the possible arrangements for the columns in a typical building. To automate the process, once all the possible patterns are generated and stored in a matrix, the lengths of reinforcement bars of different stories are calculated based on the positions stored in that matrix. The length of dowels is calculated considering the depth of the foundation and hook length. Similarly, the length of steel bars of the last storey is calculated considering the slab thickness and hook length.

2.3 Cutting Pattern Generation Module

Cutting pattern generation module is responsible for the generation of all feasible cutting patterns for each set of lengths obtained in the previous step. In this study, this is achieved using the algorithm developed by Pierce [16]. First, the required lengths, l_1, l_2, l_3, … , l_N, are arranged in descending order. The required number of each length is shown by d_1, d_2, d_3, …, d_N. In the first pattern (i = 1), the number of bars corresponding to the longer required length (l_1) that can be cut from the standard length (L), A_{11}, is the minimum of the number of bars of that required length (d_1), and the quotient of the remaining length of the standard bar when divided by the required length. This is repeated for all other required bar lengths.

$$A_{i1} = \min\left(\left\lfloor \frac{L}{l_1} \right\rfloor, d_1\right) \tag{1}$$

$$A_{i2} = \min\left(\left\lfloor \frac{L - A_{i1} l_1}{l_2} \right\rfloor, d_2\right) \tag{2}$$

$$A_{iN} = \min\left(\left\lfloor \frac{L - \sum_{j=1}^{N-1} A_{ij} l_j}{l_N} \right\rfloor, d_N\right) \tag{3}$$

Efficient pattern number i is [A_{i1}, A_{i2}, . . . , A_{iN}], where A_{ij} is the number of demanded units of length l_j that are present in pattern number i. The algorithm proceeds as follows: If in pattern number i, there exists a j, $1 \le j \le N$, such that $A_{ij} > 0$ then let k be equal to the largest j and proceed as follow:

$$Set \;\; A_{i1} = A_{(i-1)1}, \ldots, A_{i(k-1)} = A_{(i-1)(k-1)} \tag{4}$$

$$A_{ik} = A_{(i-1)k} - 1 \tag{5}$$

$$A_{i(k+1)} = \min\left(\left\lfloor \frac{L - \sum_{j=1}^{k} A_{ij} l_j}{l_{k+1}} \right\rfloor, d_{k+1}\right) \tag{6}$$

This process continues until all the feasible patterns are generated. The Pseudocode of this pattern generation procedure can be found in Salem et al. [5].

2.4 Trim Loss Optimization Module

The next step is to select a combination of patterns that minimize the trim loss, thus generating the minimum waste possible. The optimization procedure selected is Linear Integer Programming (LIP). The model is formulated and solved using the linear solver CPLEX [17]. The objective function and constraints implemented are described next.

2.4.1 Objective Function

Equation below represents the objective function:

$$Minimize \;\; \sum_{i \in P} (L_i X_i - (\sum_{j \in J} B_{ij} . V_j)) \tag{7}$$

The set notation adopted is as follows: The set of patterns is indexed by i, such that the total number of patterns obtained from the cutting pattern generation module is P. The set of bar lengths demanded, which are cut from the standard steel bar lengths L_i, of each cutting pattern i, is indexed by the letter j. The total number of demanded bar lengths is thus represented by the letter J. Each pattern i, and each demanded length j is associated with a single value that represents the total number of bars of length j to be cut from L_i, and this value is represented by the notation B_{ij}. V_j denotes the length of demanded steel bar j. A single type of decision variable, an integer variable, X_i, specifies the number of times that pattern i is used. For instance, if in the final optimum solution, the value of X_3 equals to 8 and the value of X_6 equals to 5, the cuts that would minimize the generated waste are to use pattern number 3 a total of 8 times and pattern number 6 a total of 5 times. A total summation over the entire set of patterns is minimized to ensure that the solution produced is the one that corresponds to the minimum trim loss throughout the entire steel bar cutting phase in a project.

2.4.2 Constraints

Two constraints are defined to outline the shape of the feasible searching space for the algorithm. The first constraint requires that the total demand of bar length *j* is met by the solution. In particular, each demand d_j is set equal to the total number of instances a particular length is used within the solution pattern generated. This is to ensure that the set of lengths specified for each structural column as dictated by the design plans are met. The second constraint defines the domain of the decision variable. This constraint ensures the integrity of the decision variables, such that the variable x_i can take only integer values.

$$\sum_{i \in P} B_{ij} X_i = d_j \qquad \forall j \in J \tag{8}$$

$$X_i \in Z^+ \quad \forall i \in P \tag{9}$$

3 Case Study

The proposed framework was adopted to obtain the lapping patterns of the column's steel bars of an actual 6 storey building. The data collection Excel spread sheet was filled up based on the information displayed in Table 1, acquired from the structural drawings of the project (Figure 2). The standard length for the bars of this project is 12 meters. The analyzer code was then used to read the input data and perform the lapping pattern generation, cutting pattern generation and optimization operations.

Table 1. Structural data regarding stories and column sections obtained from the ETABS structural design file

Story height	Bar size	# of bars	Hook length	Overlap length	Slab depth	Found. depth
357	25	24	35	145	20	100
340	25	20	35	145	20	
340	25	20	35	145	20	
340	25	16	35	145	20	
340	25	16	35	145	20	
380	25	12	35	145	20	

Table 2 shows the cutting patterns and the number of times each pattern should be used (X_i) to fulfill the demand list while resulting in the minimum waste. A total of 1620 lapping patterns, and accordingly 49031 cutting patterns, were generated. Lapping pattern number 305 was found to produce the minimum amount of waste compared to the other 1619 lapping patterns. A total of 31 cutting patterns were generated for this lapping pattern. Among these 31 cutting patterns, only 6 were used in the optimum answer. As can be seen in Table 2, 48 bars with a standard length of 12 m, were cut to fulfill the required bar demand, shown in the Demand row of Table 2. The selected set of patterns and X_i resulted in 41.43 m of trim loss, generating 7.2% waste, considering the total length used (576 m).

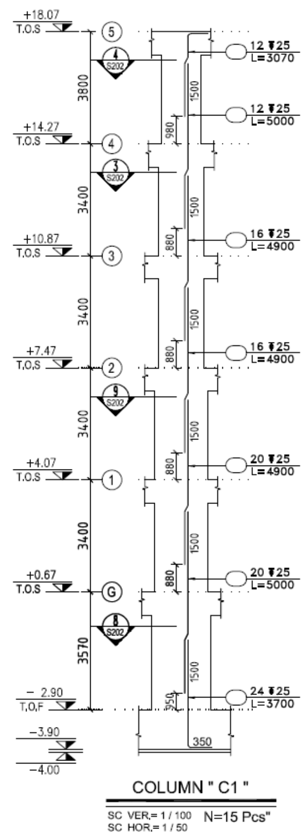

Figure 2. The original shop drawing prepared by the consultant

To show the effectiveness of the proposed framework, the bar demand list was prepared according to the structural drawings (Figure 2) and entered to the developed coded analyzer as the input for the third and fourth module, i.e. pattern generation and optimization modules. Table 3 shows the obtained cutting pattern along with the number of times that each of them is to be used (X_i) to fulfill the bar demand list. A total of 29 cutting patterns were generated, while only 7 patterns were included in the optimum solution. As can be seen in Table 2, 52 bars with a standard length of 12 m were cut to fulfill the required bar demand list. The selected set of patterns and X_i resulted in 83.56 m of trim loss, generating 13.4% waste, considering the total length used (624 m).

Comparing the total amount of material used and the

Table 2. Optimized cutting patterns based on lengths from actual shop drawing

	Patterns – number of bars (B_{ij}) of length V_j to be cut from the standard length (L)							Number of time each pattern is used in the final solution (X_i)	$X_i \times V_j \times \beta_{ij}$	$X_i \times L$
Patterns used in the final solution (i)	V_1 = 5.35m	V_2 = 5.18m	V_3 = 5.05m	V_4 = 4.65m	V_5 = 4.65m	V_6 = 3.6m	V_7 = 2.7m			
1	2	-	-	-	-	-	-	6	64.2	72
8	-	2	-	-	-	-	-	10	103.6	120
14	-	-	2	-	-	-	-	8	80.8	96
19	-	-	-	2	-	-	1	10	120	120
24	-	-	-	-	2	-	1	2	24	24
25	-	-	-	-	1	2	0	12	142.2	144
							Sum	48	534.8	576
							Trim loss	576-534.8=41.2		
							Total waste	41.2/576=7.2%		

Table 3. Optimized cutting patterns based on length generated from this framework

	Patterns – number of bars (B_{ij}) of length V_j to be cut from the standard length (L)							Number of time each pattern is used in the final solution (X_i)	$X_i \times V_j \times \beta_{ij}$	$X_i \times L$
Patterns used in the final solution (i)	V_1 = 5.0m	V_2 = 5.0m	V_3 = 4.9m	V_4 = 4.9m	V_5 = 4.9m	V_6 = 3.7m	V_7 = 3.07m			
1	2	-	-	-	-	-	-	6	60	72
8	-	2	-	-	-	-	-	10	100	120
14	-	-	2	-	-	-	-	8	78.4	96
19	-	-	-	2	-	-	-	8	78.4	96
23	-	-	-	-	2	-	-	4	39.2	48
24	-	-	-	-	1	1	1	12	140.04	144
26	-	-	-	-	-	3	0	4	155.4	48
							Sum	52	540.44	624
							Trim loss	624-540.44=83.56		
							Total waste	83.56/624=13.4%		

waste produced (Tables 2 and 3), reveals the effectiveness of the developed framework and the potential for reducing the rebar waste in construction projects through optimizing the cutting patterns. Optimizing the cutting patterns in the case project resulted in 49.5% waste reduction and a saving of 7.7% in the total amount of material used. Considering a unit weight of 3.85 kg/m for the bar with a diameter of 25 mm, used for these columns, and the fact that 15 of these columns were constructed in the case project, 2.77 ton of used steel bars and 2.43 ton of waste steel bars could be saved. This could in turn, result in considerable reduction in the cost of materials as well as the cost and energy consumed to recycle steel bars at the end of life of the structure. The proposed framework can also be extended to optimize the cutting pattern for other elements including shear walls.

4 Conclusion

Materials in the form of one-dimensional stocks such as steel bars generate a major fraction of the construction waste. This is due to the fact that these stocks are usually purchased in standard lengths and the required project lengths are shorter than the standard lengths. In this paper, a framework aiming at minimizing the trim loss, through automating the processes of lapping pattern generation, cutting pattern generation and trim loss optimization, was developed. A case study was presented to illustrate the effectiveness of the framework. The total amount of standard steel bars cut and the amount of generated waste produced after optimization were compared with those obtained from the actual shop drawings. A decrease of 7.7% and 49.6% in the total amount of material used and generated waste was observed respectively. The results highlight the potential for considerable reduction in rebar waste. Although, it should be noted that customizing the cutting pattern to minimize the rebar waste may in some cases add to the complexity of the rebar installation operation and thus reduce productivity. Therefore, careful consideration of costs and benefits by considering the project specific conditions is required.

References

1. Danatzko, J. and H. Sezen, *Sustainable Structural Design Methodologies.* Practice Periodical on Structural Design and Construction, 2011. **16**(4): p. 186-190.
2. Moussavi Nadoushani, Z.S. and A. Akbarnezhad, *Effects of structural system on the life cycle carbon footprint of buildings.* Energy and Buildings, 2015. **102**: p. 337-346.
3. Akbarnezhad, A. and Z.S. Moussavi Nadoushani, *A computational method for selection of optimal concrete recycling strategy.* Magazine of Concrete Research, 2014. **67**(11): p. 543-558.
4. Poon, C.S., A.T.W. Yu, and L. Jaillon, *Reducing building waste at construction sites in Hong Kong.* Construction Management and Economics, 2004. **22**(5): p. 461-470.
5. Salem, O., A. Shahin, and Y. Khalifa, *Minimizing Cutting Wastes of Reinforcement Steel Bars Using Genetic Algorithms and Integer Programming Models.* Journal of Construction Engineering and Management, 2007. **133**(12): p. 982-992.
6. Chen, Y. and T. Yang, *Lapping Pattern, Stock Length, and Shop Drawing of Beam Reinforcements of an RC Building.* Journal of Computing in Civil Engineering, 2013. **29**(1): p. 04014028.
7. Porwal, A. and K. Hewage, *Building Information Modeling–Based Analysis to Minimize Waste Rate of Structural Reinforcement.* Journal of Construction Engineering and Management, 2011. **138**(8): p. 943-954.
8. Benjaoran, V. and S. Bhokha, *Three-step solutions for cutting stock problem of construction steel bars.* KSCE Journal of Civil Engineering, 2014. **18**(5): p. 1239-1247.
9. Tekla, *Tekla Structures*. Kennesaw, GA.
10. Gilmore, P.C. and R.E. Gomory, *A Linear Programming Approach to the Cutting_Stock Problem.* Operation Research, 1961. **9**(6): p. 849-859.
11. Goulimis, C., *Optimal solutions for the cutting stock problem.* European Journal of Operational Research, 1990. **44**(2): p. 197-208.
12. Haessler, R.W., *Controlling Cutting Pattern in One-Dimensional Trim Problems.* Operations Research, 1975. **23**(3): p. 483-493.
13. MathWorks, I., *MATLAB*. 2011.
14. ACI, *Building Code Requirements for Structural Concrete (ACI 318-08) and Commentary*. 2008, American Concrete Institute: Farmington Hills, MI.
15. Gossett, E., *Discrete Mathematics with proof*. 2009, Hoboken, New Jersey: John Wiley & Sons.
16. Pierce, J.F., *Some large-scale production scheduling problems in the paper industry* in *Prentice-Hall*. 1964: Englewood, N.J.
17. IBM, *CPLEX optimizer*. 2009, IBM: NYC, USA.