Cutting optimization of structural tubes to build agricultural light aircrafts

Article *in* Annals of Operations Research · July 2009

DOI: 10.1007/s10479-008-0438-7 · Source: DBLP

2 authors:

Alexander Abuabara
Texas A&M University

Reinaldo Morabito
Universidade Federal de São Carlos

Ann Oper Res (2009) 169: 149–165
DOI 10.1007/s10479-008-0438-7

Cutting optimization of structural tubes to build agricultural light aircrafts

Alexander Abuabara · Reinaldo Morabito

Published online: 17 September 2008

Abstract In this study we deal with the one-dimensional cutting of metallic structural tubes used in the manufacturing of agricultural light aircrafts. The problem is modeled by mixed integer linear formulations aiming to minimize material trim losses and considering the possibility of generating remainders (leftovers) with enough size to reuse. To validate the application of the models in practice, we carried out experiments with real data of order lists from Ipanema, an agricultural airplane produced by a Brazilian aeronautical company. The models were solved using a modeling language and an optimization software. The computational results show that the models are useful in supporting decisions in this cutting process.

Keywords One-dimensional cutting problem · Agricultural light aircrafts · Tube cutting · Mixed integer linear programming · Aeronautics industry

1 Introduction

Neiva, a wholly owned Embraer subsidiary, is the unique manufacturer of agricultural light aircrafts in Brazil and one of the few companies worldwide in this sector. *Ipanema* is Neiva's best selling light airplane with more than 30 years of uninterrupted production and more than 1000 units sold. Neiva's national market share is currently approximately 80 percent, making it a dominant force in the domestic agricultural aviation market. Other agricultural light aircrafts are imported from the United States. A traditional illustration of Ipanema is shown in Fig. 1. The essential use of an agricultural airplane is the aerial application called crop dusting and involves spraying crops with fertilizers, pesticides and fungicides and spreading seeds (Neiva 2006).

In this work, we studied the one-dimensional cutting stock problem that arises in the manufacturing process of a light aircraft (i.e., less than 1 ton), such as Ipanema. The problem is how to cut a set of objects (stock lengths of metallic tubes) with known sizes and quantities

A. Abuabara · R. Morabito (✉)
Production Engineering Department, Federal University of Sao Carlos, 13565-905 São Carlos, SP, Brazil
e-mail: morabito@ufscar.br

Fig. 1 The agricultural light aircraft Ipanema

to make exactly a set of items (order lengths) with specified sizes and demands to be fulfilled. The objects are available in stock in different sizes and enough quantities to produce all ordered items (because they are imported in relatively large lots by the company to prevent a lack of material in the cutting process). The items cut are parts of the structure of Ipanema (e.g., fuselage, wings, etc.) and some of the aircraft systems manufactured by the company.

An immediate optimization criterion of this cutting problem would be to minimize the waste of the material cut. The minimum trim loss solution becomes particularly important for the present company for which the cost of the raw material of the structural metallic tubes represents at least one-fourth of the total costs of the final product. The company estimates that the cutting process produces material wastes of around 52 meters of tubes per day, which means 94 kg of material per day and about US$ 390 thousand per year. This waste consists of good-quality scraps, which are useless for practical purposes because of their small sizes. Part of this waste can be avoided by a more effective production planning of the cutting process, which does not imply additional investments in capacity.

The company does not use any cutting planning optimization tool and in practice the operators often cut the objects into the items without a more elaborate plan. An optimized plan can also be useful to support inventory management decisions. A peculiar aspect of this cutting process is the possibility of generating object remainders (leftovers) with enough size to fulfill future item demands. These leftovers can be maintained in the stock of objects for future usage. Therefore, the present problem can be seen as a one-dimensional cutting stock problem with possible leftover generation and reuse. The company does not consider anticipating the cut of items and keeping them in stock because there are difficulties with tracking issues and quality control.

According to Dyckhoff (1990)'s cutting and packing typology, this problem can be classified as 1|V|D|F, where 1 refers to a one-dimensional problem, V means that all ordered items must be cut, D means that the available objects may be of different sizes and F means that there are only a few items and they are of different sizes. More recently, Wäscher et al. (2007) extend this typology with the introduction of new criteria that define other new problems, beyond the ones defined previously. According to this extended typology, the problem can be classified as a residual cutting stock problem (RCSP) and categorized as a hybrid one-dimensional cutting stock problem (HODCSP).

Although there are many studies in the literature dealing with the classic one-dimensional cutting stock problem (see e.g. the surveys and special issues in Bischoff and Wäscher 1995; Mukhacheva 1997; Dyckhoff et al. 1997; Arenales et al. 1999; Wang and Wäscher 2002; Hifi 2002; Oliveira and Wäscher 2007; see also the references in ESICUP 2007), only a few of them consider the leftover generation and reuse (e.g., Dyckhoff 1981; Gradisar et al. 1997). For instance, Gradisar et al. (1997) studied the problem of minimizing trim loss in the

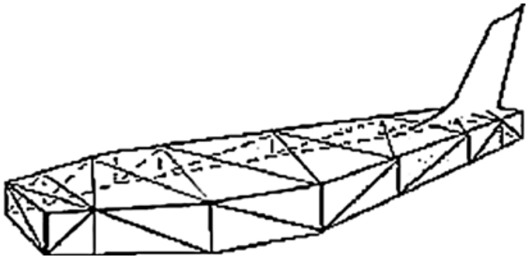

Fig. 2 The airplane structure

one-dimensional roll cutting of a clothing company. The rolls have different lengths and, as in the present cutting problem, it is also possible to produce roll remainders and keep them in stock for future usage. To the best of our knowledge, we are not aware of other studies in the literature dealing with cutting problems in agricultural light aircraft manufacturing.

In this paper, we present mixed integer linear models to cope with the structural metallic tube cutting problem to manufacture agricultural light aircrafts. The first model is based on the model proposed in Gradisar et al. (1997) and the second model is a simplification of the first one. These models are solved using a modeling language (GAMS) with an optimization software (CPLEX). To validate the models in practice, we did experiments using order lists from Ipanema provided by the company. The cutting plans generated by the models are compared to the ones used by the company, and the results show that the models are useful in supporting decisions in this cutting process.

This paper is organized as follows. In Sect. 2, we describe the present cutting problem in more details. In Sect. 3, we present mixed integer linear models to represent the problem. In Sect. 4, we analyze the computational results obtained by the models when applied to actual data provided by the company. Finally, in Sect. 5 we present our concluding remarks and discuss perspectives for future research.

2 Problem description

The production planning of agricultural light airplanes is mainly based on the product demand. The company MRP (Material Requirement Planning) determines a list of items to be produced by the cutting process as a function of the airplane demand. A typical planning horizon is one or more weeks and order lists are released daily or weekly. As mentioned, the company does not consider producing items for stock because of difficulties with tracking issues and quality control. After cutting the items, the next steps of the production flow are basically: assembling the wings, body (fuselage) and instruments, painting, flight testing and product delivery.

The airplane structure is formed by trussed-frame porticos, as illustrated in Fig. 2—a spatial truss is some timber framed together to bridge a space or form a bracket, to be self-supporting and to carry other timbers. These trussed-frames are special structures and the tube specifications are defined in the airplane project. The items that form the timber of the trussed-frame have determined sizes and demands by kind of material and tube thickness specification. These trusses are used either internally (e.g. fuselage and wings) or externally (e.g. landing system) in the airplane. Some of the reasons for using this structure in aviation are its lightness and resistance. The material consists of tubes which are metallic alloys like chromo-molybdenum and stainless steel 4130.

The structural metallic tube cutting problem can be defined as follows: how to cut a set of stock lengths of metallic tubes (objects), with known sizes and available quantities, to produce exactly a set of order lengths (items) with specified sizes and demands, so that the total trim loss is minimized (recall that the leftovers cut are not considered as waste), and the total object length cut is minimized—the idea is to favor the use of the leftovers available in stock and to avoid the generation of new leftovers.

We assume that objects are available in stock in sufficiently large quantities to produce all ordered items. The setup times of the cutting equipment are not relevant and are disregarded. Without loss of generality, we consider all object lengths and item lengths and demands as integer numbers. As mentioned, a peculiar aspect of this problem is the possibility of generating a small number of leftovers with enough length to be used for a future demand.

The company does not use an optimization software to determine the cutting patterns. The cutting planning is performed by the operators based on their experience and using electronic spreadsheets. The operators are always looking for the maximum area use of each object cut. They usually allocate the largest items first and the largest objects first, leaving the allocation of the smallest items to the end. They also try to allocate items of same length together. Note that their allocation rule is similar to the well-known heuristic FFD (First Fit Decreasing).

As pointed out by different authors, there are no general and efficient solution methods for cutting stock problems due to the their computational complexity (these problems are in general NP-hard; see e.g. Dyckhoff and Finke 1992; Dyckhoff et al. 1997; Wäscher et al. 2007; ESICUP 2007) and the diversity of cases in which they can appear. Three possible solution approaches for these problems are: (i) linear programming with column-generation and rounding procedures, (ii) mixed integer linear programming and (iii) heuristics and approximate methods.

In this study we did not use approach (i) because the demand of some items can be very small (e.g. there are items whose demand is only one), which does not encourage the application of linear relaxation, even considering rounding procedures. Another difficulty is that the availability of some stock lengths can be very small (e.g. the availability of a stock leftover is often only one), which also complicates the application of linear relaxation rounding. Moreover, we preferred approach (ii) to approach (iii) because we believed it was possible to find optimal integer solutions within a reasonable runtime for typical problem instances of the company. Initially, we focused the mathematical model proposed in Gradisar et al. (1997, 1999, 2002) for the cutting optimization of fabric rolls in the clothing industry, since these industrial settings have some similarities with the present study. In these studies, the objective is to minimize the trim loss and the generation of some few remainders to reuse is permitted.

3 Mathematical modeling

Each order list is planned individually and for each list there are different stock lengths available. The typical situation is that a large amount of stock consists of a few standard lengths (say, lengths 3000, 3500 and 6000 mm) of the suppliers. Only a small part of the stock is of various different substandard lengths, which are remainders of these objects generated in the cutting of previous order lists. Consider the following notations for the mathematical formulation:

Items:

m number of different types (sizes) of items
l_i length of item type i, $i = 1, \dots, m$
d_i required quantity (demand) of item type i, $i = 1, \dots, m$

Objects:

n number of stock objects
b_j length of stock object j, $j = 1, \dots, n$
N minimum length accepted as a leftover

where N, l_i, d_i and b_j are integers. We wish to obtain:

x_{ij} number of items of type i cut from object j.

The mathematical model presented in Gradisar et al. (1997) was not (explicitly) formulated as a mixed integer linear programming model. The following model (Model 1) is an adaptation of that model under the condition of not having limits on the number of different items allocated per object and having enough availability of objects in stock. The model involves determining the quantities x_{ij} so that the item demands are fulfilled and the overall trim loss is minimized (recall that the overall trim loss is all waste material with lengths smaller than a chosen value N).

The value of N is usually set arbitrarily between $\min\{l_i, i = 1, \dots, m\}$ and $\min\{b_j, j = 1, \dots, n\}$ by the decision maker. The lower setting of N means that any returned stock length remainder would be useless if future order lengths smaller than $\min\{l_i, i = 1, \dots, m\}$ are not expected. The upper setting of N would prevent the return of stock lengths larger than some stock lengths previously available. The model minimizes the total trim loss, that is, the sum of the trim losses t_j of each object j cut:

$$\min \sum_{j=1}^{n} t_j \tag{1}$$

subject to knapsack constraints—the total length of all items i allocated to object j must not exceed the object length b_j:

$$\sum_{i=1}^{m} l_i x_{ij} + \delta_j = b_j \quad \forall j \tag{2}$$

where δ_j is the non-negative slackness variable representing either a waste or a leftover of the cutting pattern of object j. The model also includes demand constraints—the total quantity of items i cut must be exactly equal to the demand d_i:

$$\sum_{j=1}^{n} x_{ij} = d_i \quad \forall i \tag{3}$$

and the following functions: z_j indicates if object j is used in the cutting plan ($z_j = 1$):

$$z_j = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} x_{ij} > 0,\ \forall j, \\ 0 & \text{if } \sum_{i=1}^{m} x_{ij} = 0,\ \forall j. \end{cases} \tag{4}$$

t_j is the trim loss of a used object j if its slackness δ_j is not a leftover, i.e., if $\delta_j < N$ ($t_j = \delta_j$):

$$t_j = \begin{cases} \delta_j & \text{if } z_j = 1 \text{ and } \delta_j < N\ \forall j, \\ 0 & \text{if } z_j = 0 \text{ or } \delta_j \geq N\ \forall j \end{cases} \tag{5}$$

and u_j indicates if the slackness δ_j of a used object j is a leftover, i.e., if $\delta_j \geq N$ ($u_j = 1$):

$$u_j = \begin{cases} 1 & \text{if } z_j = 1 \text{ and } \delta_j \geq N \ \forall j, \\ 0 & \text{if } z_j = 0 \text{ or } \delta_j < N \ \forall j. \end{cases} \tag{6}$$

The model also takes into account a constraint limiting the maximum number of leftovers generated to one (note that this limit prevents the model from overproducing leftovers in order to eliminate the trim loss of the cutting patterns—other limits could be used instead of one):

$$\sum_{j=1}^{n} u_j \leq 1 \tag{7}$$

and the domains of the variables:

$$x_{ij} \geq 0, \quad \text{integer } \forall ij, \qquad \delta_j \geq 0 \quad \forall j, \qquad t_j \geq 0 \quad \forall j, \qquad z_j \in \{0, 1\} \quad \forall j,$$
$$u_j \in \{0, 1\} \quad \forall j$$

This model can be stated as mixed integer program:

Model 1

$$\min \quad \sum_{j=1}^{n} t_j \tag{1}$$

$$\sum_{i=1}^{m} l_i . x_{ij} + \delta_j = b_j \quad \forall j \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = d_i \quad \forall i \tag{3}$$

$$z_j \leq \sum_{i=1,\dots,m} x_{ij} \quad \forall j \tag{8}$$

$$\sum_{i=1,\dots,m} x_{ij} \leq M z_j \quad \forall j \tag{9}$$

$$(\delta_j - N) \geq -M w_j + \varepsilon \quad \forall j \tag{10}$$

$$(\delta_j - N) \leq M(1 - w_j) \quad \forall j \tag{11}$$

$$t_j - M w_j \leq 0 \quad \forall j \tag{12}$$

$$t_j - M z_j \leq 0 \quad \forall j \tag{13}$$

$$-\delta_j + t_j \leq 0 \quad \forall j \tag{14}$$

$$\delta_j - t_j + M w_j + M z_j \leq 2M \quad \forall j \tag{15}$$

$$-z_j + u_j \leq 0 \quad \forall j \tag{16}$$

$$w_j + u_j \leq 1 \quad \forall j \tag{17}$$

$$z_j - w_j - u_j \leq 0 \quad \forall j \tag{18}$$

$$\sum_{j=1}^{n} u_j \leq 1 \tag{7}$$

$$x_{ij} \geq 0, \quad \text{integer}, \quad \delta_j \geq 0, \quad t_j \geq 0, \quad z_j \in \{0,1\}, \quad w_j \in \{0,1\},$$
$$u_j \in \{0,1\}, \quad \forall i, j. \tag{19}$$

Condition (4) is rewritten as restrictions (8) and (9), where M is a sufficiently large number (e.g., $M = \max_j\{b_j\} - \min_i\{l_i\}$).

To rewrite condition (5), we first define an auxiliary binary variable w_j to indicate if $\delta_j < N$ ($w_j = 1$) or $\delta_j \geq N$ ($w_j = 0$). Inequalities (10) and (11) make this relation, where ε is a small positive integer number to change the signals of the inequalities from "$\geq$" to "$>$" (this is a trick to code these inequalities in a modeling language). In the case where all input parameters are integer numbers, then $\varepsilon = 1$. Therefore, using (10) and (11), condition (5) is restated as:

$$t_j = \begin{cases} \delta_j & \text{if } z_j = 1 \text{ and } w_j = 1 \ \forall j, \\ 0 & \text{if } z_j = 0 \text{ or } w_j = 0 \ \forall j \end{cases}$$

which can be rewritten by constraints (12), (13), (14) and (15) (for more details, see e.g. Williams 1978). It should be noted that in order to guarantee that if either $z_j = 0$ or $w_j = 0$, then $t_j = 0$, we use the fact that the sum of all t_j is minimized in objective function (1).

To rewrite condition (6), we also use the auxiliary binary variable w_j to indicate if $\delta_j \geq N(w_j = 0)$ or $\delta_j < N(w_j = 1)$. Using (10) and (11), condition (6) is restated as:

$$t_j = \begin{cases} 1 & \text{if } z_j = 1 \text{ and } w_j = 0 \ \forall j, \\ 0 & \text{if } z_j = 0 \text{ or } w_j = 1 \ \forall j \end{cases}$$

which can be rewritten by constraints (16), (17) and (18) (see e.g. Williams 1978). Thus, constraints (10) to (18) together describe conditions (5) and (6) of the previous model. Restriction (7), which limits the number of possible leftovers to be generated, is considered in the same way as the previous model.

Note that there is a shortcoming in Model 1: there may be no feasible solutions for some instances, even if the lengths of the items are shorter than those of the stock objects. More precisely, if an object j used in the cutting plan has slackness $\delta_j \geq N$, then u_j must be 1 (considered as a leftover) according to (6) or (10)–(18). If there are two or more objects with slackness equal to or larger than N in the cutting plan, the model considers this solution infeasible due to (7). This is not a desirable behavior since this solution is feasible if one of the slackness is considered leftover and the others are considered trim losses. It should be mentioned that this shortcoming did not prevent Model 1 of finding a feasible solution in the computational experiments of Sect. 4. The model described below does not have this shortcoming.

Model 2

As mentioned, Model 1 is a mixed integer linear programming formulation for a particular case of the mathematical model proposed in Gradisar et al. (1997). In the following part, we present a simplification of Model 1, here called Model 2. The model is defined as:

$$\min \quad \sum_{j=1}^{n} t_j \tag{1}$$

$$\sum_{i=1}^{m} l_i x_{ij} \leq b_j \quad \forall j \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = d_i \quad \forall i \tag{3}$$

$$N u_j \leq b_j z_j - \sum_{i=1}^{m} l_i x_{ij} \quad \forall j \tag{20}$$

$$b_j z_j - \sum_{i=1}^{m} l_i x_{ij} \leq t_j + u_j M \quad \forall j \tag{21}$$

$$\sum_{j=1}^{n} u_j \leq 1 \tag{7}$$

$$x_{ij} \geq 0, \quad \text{integer}, \quad t_j \geq 0, \quad z_j \in \{0,1\}, \quad u_j \in \{0,1\}, \quad \forall i, j. \tag{22}$$

Note that the objective function (1) and the restrictions (2), (3) and (7) are the same as previously. As before, variable z_j indicates if object j is used ($z_j = 1$) or not ($z_j = 0$) in the cutting plan. If the difference between the length of object $j\,(b_j)$ and the sum of all item lengths allocated to object j is larger than or equal to N, then $u_j = 1$, otherwise $u_j = 0$. This is described in inequality (20). If an object j is chosen ($z_j = 1$) and $u_j = 0$ (by (20)), restriction (21) determines t_j as the difference between the length of object $j\,(b_j)$ and the sum of all item lengths allocated to object j. Otherwise, $t_j = 0$.

The objective function (1) of Models 1 and 2 can be slightly modified in order to also accomplish the criterion of minimizing the total object length cut (see Sect. 2). For alternative minimum trim loss solutions, the idea is to favor the ones that avoid the generation of new leftovers. In this case, (1) should be replaced by:

$$\min \quad \sum_{j=1}^{n} t_j + \frac{\sum_{j=1}^{n} b_j z_j}{\sum_{j=1}^{n} b_j}. \tag{23}$$

Given that the input parameters of the model are assumed to be integers, the first term of this expression (i.e. total trim loss) is integral. Therefore, since the second term always yields a value between 0 and 1, this term unties alternative minimum trim loss solutions in favor of solutions minimizing the total object length cut.

Model 2 with multiple periods

Although the company does not consider multiple production periods when planning the cutting patterns of the objects in each period, the models above could be extended to deal with multi-period cutting planning, taking into account the benefits of advancing the cutting of items and keeping them in stock, instead of generating and stocking leftover objects in order to satisfy future demands. For example, by defining k as the time period index ($k = 1, 2, \dots, q$), Model 2 with multiple periods can be stated as:

$$\min \quad \sum_{k=1}^{q}\sum_{j=1}^{n} t_{jk} \tag{24}$$

$$Nu_{jk} \le b_j z_{jk} - \sum_{i=1}^{m} x_{ijk} l_i \quad \forall jk \tag{25}$$

$$b_j z_{jk} - \sum_{i=1}^{m} x_{ijk} l_i \le t_{jk} + Mu_{jk} \quad \forall jk \tag{26}$$

$$\sum_{p=1}^{k}\sum_{j=1}^{n} x_{ijp} \ge \sum_{p=1}^{k} d_{ip} \quad \forall i, \forall k, k \ne q \tag{27}$$

$$\sum_{k=1}^{q}\sum_{j=1}^{n} x_{ijk} = \sum_{k=1}^{q} d_{ik} \quad \forall i \tag{28}$$

$$\sum_{k=1}^{q}\sum_{j=1}^{n} u_{jk} \le 1 \tag{29}$$

$$\sum_{j=1}^{n} b_j z_{jk} \le C_k \quad \forall k \tag{30}$$

$$x_{ijk} \ge 0, \quad \text{integer}, \quad t_{jk} \ge 0, \quad z_{jk} \in \{0, 1\}, \quad u_{jk} \in \{0, 1\}, \quad \forall i, j, k, \tag{31}$$

where C_k in (30) is the cutting capacity (in terms of total length cut) of period k and the notations $t_{jk}, u_{jk}, z_{jk}, d_{jk}, x_{ijk}$, etc., are defined in a similar way as before.

4 Computational results

To solve Models 1 and 2, we use the modeling language GAMS 19.6 and the optimization solver CPLEX 7.0 (Brooke et al. 1998) in a microcomputer with processor Intel Pentium III 550 MHz and 512 MB of RAM. We utilize the default values of the GAMS and CPLEX parameters, except for the optimal gap which was fixed to zero and the iteration limit which was relaxed. The execution time for each example was arbitrarily limited to one hour, considered reasonable to support the involved decisions in the company. In cases where it was not possible to find an optimal solution within this time limit, the best integer solution found was reported and marked with an asterisk.

Table 1 presents the optimal solution found by Models 1 and 2 when solving a one-dimensional fabric roll cutting with roll remainders presented in Gradisar et al. (1997). This is an example with $\sum_{i=1}^{m} d_i = 220$ ordered items of $m = 3$ different lengths ($l_1 = 239, l_2 = 188, l_3 = 134$) and with $n = 5$ available objects ($b_1 = 10\,280, b_2 = 10\,220, b_3 = 10\,160, b_4 = 10\,180, b_5 = 10\,100$), resulting in 41 variables and 65 constraints in Model 1 and 31 variables and 15 constraints in Model 2. Considering $N = 134$ and using GAMS/CPLEX, both models solved the problem in approximately 12 seconds. The values of the table correspond to the number of items of each type allocated to each object (x_{ij}).

Table 1 Solution of a fabric roll example presented in Gradisar et al. (1997)

Items	Objects					Total
	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	
$i=1$	22	34	36	20	28	140
$i=2$	26	9	4	5	11	55
$i=3$	1	3	6	5	10	25
Total items	49	46	46	30	49	220
Object length b_j [mm]	10 280	10 220	10 160	6390	10 100	
Leftover length [mm]	0	0	0	3790	0	

The cutting patterns of objects 1, 2, 3 and 5 fulfilled the object lengths (b_j), while the pattern of object 4 yields a roll remainder of 3790 mm to reuse.

The next experiment refers to 46 real data instances (order lists) of the aircraft Ipanema, provided by the company. We compared the solutions obtained by GAMS/CPLEX solving Models 1 and 2 and the solutions obtained by the company operators (obtained from the spreadsheets) and used in practice. In these examples, the number of total ordered items varies from 7 to 120, the number of item types (m) varies from 2 to 34, and the number of available stock objects (n) varies from 5 to 16. Note that the company instances are relatively small in terms of number of items and objects, if compared to one-dimensional cutting instances of other industrial settings, as e.g. in the paper and steel industries. The objects have standard lengths of 3000, 3500 and 6000 mm—each example uses one of these lengths. The company operators considered N as the length of the lesser item of each order list (i.e., $N = \min\{l_i, i = 1, \dots, m\}$). The complete description of these examples can be found in Abuabara (2006).

Table 2 resumes the results obtained by the methods for each example (columns Manual Programming, Model 1 and Model 2). For each method the table presents three sub-columns containing: the number of objects cut, $\sum_{j=1}^{n} z_j$ (Objects), the number of leftovers generated (Leftovers) and the trim loss percentage (Loss), calculated dividing the total discarded length, $\sum_{j=1}^{n} t_j$, by the total cut length, $\sum_{j=1}^{n} b_j z_j$, of each example. The two last lines of the table present the solution averages and the respective standard deviations.

Except for example *45*, all solutions of Model 2 obtained by GAMS/CPLEX within the time limit are proven optimal, unlike 6 solutions in Model 1, indicating that Model 2 has a better performance in this experiment. Note that the solutions of Model 1 for examples *7*, *14*, *19*, *21* and *35* are not optimal, however they are close to the optimal value. The trim loss average values of the models are almost the same and about 55% (from 2.32% to 1.04%) better than the values of the company solutions. The models generate approximately 36% (from 69 to 44) less leftovers than the manual programming and they use around 3.1% (from 163 to 158) less raw material (stock objects). These savings are significant in practice. Moreover, for all examples the model solutions meet the desired limit of at most one generated leftover (see constraint (7)), unlike the solutions from the company (see e.g. examples *13*, *19*, *20*, *21*, *23*, *24*, *31*, *32*, *40*, *42*, *44* and *45*).

Table 3 presents the total number of ordered items, $\sum_{i=1}^{m} d_i$ (Total of items), the total number of item types, m (Total of item types), the total number of available objects, n (Total of objects) and the execution times (in seconds) required by GAMS/CPLEX to solve Models 1 and 2 for each example. Notice that, in general, Model 2 requires less time to find an optimal solution than Model 1, possibly for involving a smaller number of variables and constraints (e.g., in example *3*, Model 1 contains 73 variables and 102 constraints, while

Table 2 Solution values of 46 problem instances (order lists) from the Ipanema aircraft

Ex.	Manual Programming			Model 1			Model 2		
	Objects	Leftovers	Loss	Objects	Leftovers	Loss	Objects	Leftovers	Loss
1	2	1	1.57%	2	1	1.57%	2	1	1.57%
2	3	1	0.69%	3	1	0.69%	3	1	0.69%
3	4	1	1.30%	4	1	0.24%	4	1	0.24%
4	2	1	1.83%	2	1	0.83%	2	1	0.83%
5	3	1	3.67%	3	1	0.67%	3	1	0.67%
6	4	1	6.04%	4	1	3.06%	4	1	3.06%
7	4	1	0.72%	4	1	*0.01%	4	1	0.00%
8	2	1	2.75%	2	1	0.04%	2	1	0.04%
9	3	1	0.67%	3	1	0.00%	3	1	0.00%
10	2	1	0.21%	2	1	0.00%	2	1	0.00%
11	2	1	0.25%	2	1	0.00%	2	1	0.00%
12	3	1	0.36%	3	1	0.00%	3	1	0.00%
13	4	2	0.31%	4	1	0.00%	4	1	0.00%
14	6	1	0.37%	6	1	*0.05%	6	1	0.00%
15	2	1	6.57%	2	1	3.14%	2	1	3.14%
16	3	1	8.95%	3	1	4.19%	3	1	4.19%
17	4	1	6.57%	4	1	6.57%	4	1	6.57%
18	4	0	10.00%	4	1	4.71%	4	1	4.71%
19	7	5	0.08%	6	1	*0.07%	6	1	0.00%
20	3	3	0.00%	3	1	0.05%	3	1	0.05%
21	5	4	0.00%	5	1	*0.06%	5	1	0.00%
22	3	1	1.39%	3	1	1.39%	3	1	1.39%
23	4	3	1.75%	4	1	1.75%	4	1	1.75%
24	4	3	1.63%	4	1	1.63%	4	1	1.63%
25	2	1	1.00%	2	1	0.13%	2	1	0.13%
26	4	1	2.14%	3	0	0.62%	3	0	0.62%
27	4	1	0.23%	4	1	0.00%	4	1	0.00%
28	3	1	0.47%	3	1	0.00%	3	1	0.00%
29	2	1	2.17%	2	1	0.00%	2	1	0.00%
30	2	1	0.07%	2	1	0.07%	2	1	0.07%
31	3	3	0.00%	3	1	0.10%	3	1	0.10%
32	4	4	0.00%	4	1	0.11%	4	1	0.11%
33	2	1	4.00%	2	1	0.00%	2	1	0.00%
34	4	1	4.48%	4	1	2.00%	4	1	2.00%
35	6	1	4.52%	5	0	*1.94%	5	0	1.94%
36	2	1	3.08%	2	1	3.08%	2	1	3.08%
37	2	1	3.33%	2	1	0.08%	2	1	0.08%
38	2	1	1.92%	3	1	1.28%	3	1	1.28%
39	3	1	1.22%	3	1	0.28%	3	1	0.28%
40	4	2	2.50%	4	1	2.00%	4	1	2.00%
41	3	1	0.69%	3	1	0.69%	3	1	0.69%
42	6	3	5.00%	5	1	1.39%	5	1	1.39%
43	4	1	1.25%	4	1	0.00%	4	1	0.00%

Table 2 (*Continued*)

Ex.	Manual Programming			Model 1			Model 2		
	Objects	Leftovers	Loss	Objects	Leftovers	Loss	Objects	Leftovers	Loss
44	5	2	3.63%	4	1	0.37%	4	1	0.37%
45	7	2	4.00%	6	1	*0.16%	6	1	*0.16%
46	6	1	3.18%	6	1	2.87%	6	1	2.87%
Total	**163**	**69**		**158**	**44**		**158**	**44**	
Average	**3.54**	**1.50**	**2.32%**	**3.43**	**0.96**	**1.04%** (**1.14%)	**3.43**	**0.96**	**1.04%** (**1.06%)
Std. Dev.	**1.41**	**1.03**	**2.40%**	**1.20**	**0.21**	**1.48%** (**1.54%)	**1.20**	**0.21**	**1.48%** (**1.49%)

*Best integer solution found within the one-hour time limit

**Considering only the examples that were optimally solved within the time limit

Model 2 contains 65 variables and 31 constraints). Considering only the examples that were optimally solved within the time limit, Model 2 was, on average, around 45% faster than Model 1 (from 92.07 to 50.34 seconds). Model 1 was a little faster than Model 2 in only 5 out of the 46 examples (see e.g. examples *10*, *15*, *25*, *26* and *33*).

In order to evaluate the effects of increasing the maximum number of leftovers (right hand side of constraint (7)) on the trim losses, another experiment was done using Model 2. For this experiment, we arbitrarily chose 13 examples of Table 2 (examples *7*, *19*, *25*, *27*, *28*, *29*, *40*, *41*, *42*, *43*, *44*, *45* and *46*, respectively). Table 4 presents the trim losses found, $\sum_{j=1}^{n} t_j / \sum_{j=1}^{n} b_j z_j$ (in percentage) and the number of objects used, $\sum_{j=1}^{n} z_j$ (in brackets) for each of these 13 examples. As in the previous experiment, the available stock objects have standard lengths of 3000, 3500 and 6000 mm (each example uses one of these lengths), and N is the length of the lesser item of each order list (i.e., $N = \min\{l_i, i = 1, \dots, m\}$). The results are presented varying the maximum number of leftovers for 1, 2 and 3. Note that, as we increase this number, on average, the total trim loss decreases. Table 5 presents the execution times (in seconds) required by GAMS/CPLEX to solve each example. Except for example *12*, all solution values presented in the table are proven optimal within the time limit of one hour.

A final experiment was done to evaluate the reuse of leftovers in a situation with constant demand in the next periods. This is common in the present company where the throughput rate (number of aircrafts assembled per period) usually is maintained constant for a given number of periods. Our motivation in this experiment is to investigate the possibility of cycles of generation and reuse of leftovers under the assumption of constant demand. We arbitrarily chose example *3* of Table 2 and we applied consecutively Model 2 for each period, one after the other. The item demand and length (d_i, l_i) of this example is: 6×320 mm length, 3×148 mm, 6×670 mm and 6×705 mm. All stock lengths (b_j) have initially 3500 mm. In each period, the output of the model, that is, the remainders of objects generated in the period, was used as available objects in the model input of the next period.

Figure 3 resumes the results obtained in 24 consecutive periods. For each period, the figure depicts: (i) the total length of the objects used, $\sum_{j=1}^{n} z_j$ (Total object length used), (ii) the total trim loss obtained, $\sum_{j=1}^{n} t_j$ (Total trim loss), and (iii) the length of the leftover generated (Leftover length generated), possibly null. In the first period, all available objects in stock are of equal sizes and, after the first period, the leftovers generated in each period

Table 3 Execution times (in seconds) of the solutions of Table 2

Ex.	Total of items	Total of item types (m)	Total of objects (n)	Model 1 Time [s]	Model 2 Time [s]
1	7	4	5	0.44	0.14
2	14	4	7	12.01	0.25
3	21	4	16	433.29	33.48
4	10	4	8	1.03	0.59
5	15	4	5	25.23	15.32
6	20	4	8	835.88	24.37
7	36	20	5	*3600.00	1009.52
8	13	7	7	0.83	0.53
9	26	7	5	35.95	5.22
10	16	5	7	0.08	0.10
11	24	5	5	0.14	0.11
12	32	5	7	0.91	0.15
13	40	5	10	9.22	0.26
14	62	30	7	*3600.00	285.28
15	8	2	5	0.14	0.17
16	12	2	7	1.18	0.62
17	16	2	7	4.87	1.89
18	20	2	7	13.03	10.07
19	29	17	7	*3600.00	5.59
20	13	7	10	47.71	10.81
21	26	7	10	*3600.00	0.22
22	20	4	7	4.59	2.40
23	24	4	7	6.65	4.76
24	28	4	8	50.81	37.91
25	10	5	5	0.15	0.18
26	20	5	7	0.10	0.11
27	58	33	7	85.41	55.99
28	40	27	8	8.32	2.47
29	9	5	5	0.15	0.10
30	8	4	5	0.25	0.15
31	16	4	7	5.85	2.29
32	14	4	7	95.42	24.06
33	8	5	5	0.10	0.12
34	16	5	7	1322.15	14.65
35	24	5	10	*3600.00	394.06
36	10	3	5	0.17	0.14
37	15	3	5	1.23	0.65
38	20	3	7	15.56	6.47
39	25	3	8	243.24	75.87
40	32	5	10	47.84	23.04
41	16	4	10	11.16	2.04
42	100	4	10	254.15	198.74

Table 3 (*Continued*)

Ex.	Total of items	Total of item types (m)	Total of objects (n)	Model 1 Time [s]	Model 2 Time [s]
43	120	6	5	61.25	0.18
44	39	7	5	0.22	0.01
45	62	34	10	*3600.00	*3600.00
46	20	6	6	46.17	14.34
Average	**26.39**	**7.37**	**7.20**	**549.63** (**92.07**)	**127.51** (**50.34**)

*Best integer solution found within the one-hour time limit

**Considering only the examples that were optimally solved within the time limit

Table 4 Solution values of 13 problem instances varying the maximum number of leftovers

Ex.	Model 2		
	$\sum_{j=1}^{n} u_j \leq 1$	$\sum_{j=1}^{n} u_j \leq 2$	$\sum_{j=1}^{n} u_j \leq 3$
1	0.00% (4)	0.00% (4)	0.00% (4)
2	0.00% (6)	0.00% (6)	0.00% (6)
3	0.13% (2)	0.00% (2)	0.00% (2)
4	0.00% (4)	0.00% (4)	0.00% (4)
5	0.00% (3)	0.00% (3)	0.00% (3)
6	0.00% (2)	0.00% (2)	0.00% (2)
7	2.00% (4)	0.00% (4)	0.00% (4)
8	0.69% (3)	0.10% (3)	0.00% (3)
9	1.39% (3)	0.39% (3)	0.00% (3)
10	0.00% (4)	0.00% (4)	0.00% (4)
11	0.37% (5)	0.00% (5)	0.00% (5)
12	*0.16% (6)	0.03% (7)	*0.01% (7)
13	2.87% (6)	1.13% (6)	0.15% (6)
Average	**0.58%** (**4.0**)	**0.13%** (**4.1**)	**0.01%** (**4.1**)

*Best integer solution found solution within the one-hour time limit

became available in stock as an object for the next period. Note that in the 24th period, all stock leftovers were used and the cycle restarted. Considering all 24 executions, a total of 73 objects of 3500 mm were used, 4 objects in the first execution and 3 objects in the remaining. There are other examples in Table 3 for which cycles (of different periods) were found, as well as examples without cycles (these results are detailed in Abuabara 2006).

Table 5 Execution times (in seconds) of the solutions of Table 4

Ex.	Model 2		
	$\sum_{j=1}^{n} u_j \leq 1$	$\sum_{j=1}^{n} u_j \leq 2$	$\sum_{j=1}^{n} u_j \leq 3$
1	1009.52	17.92	0.30
2	5.59	141.73	0.10
3	0.18	0.12	0.12
4	55.99	5.39	6.49
5	2.47	0.86	0.03
6	0.10	0.14	0.14
7	23.04	0.32	0.10
8	2.04	23.17	0.05
9	198.74	520.91	18.04
10	0.18	0.20	0.10
11	0.01	13.16	15.10
12	*3600.00	492.16	*3600.00
13	14.34	32.32	18.13
Average	**377.86**	**96.03**	**281.44**

*Best integer solution found within the one-hour time limit

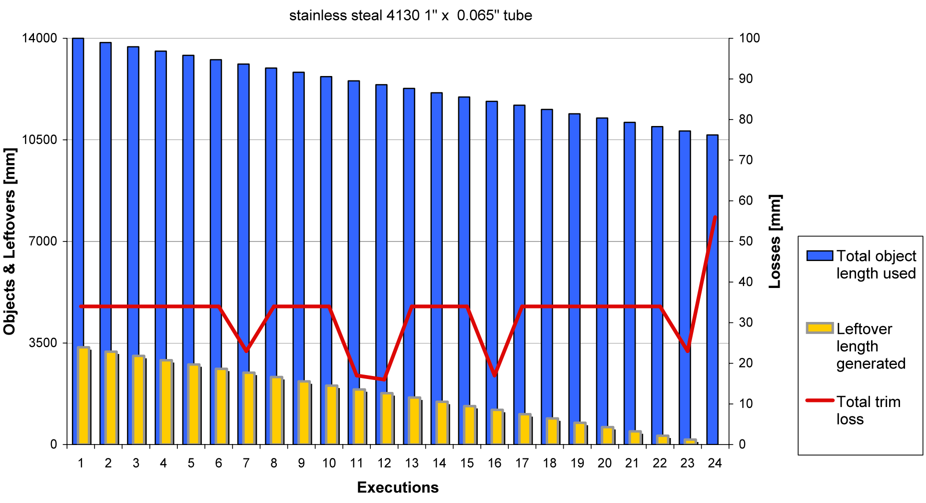

Fig. 3 Consecutive applications of Model 2 in example *3* of Table 2

5 Concluding remarks

In this work we study a one-dimensional cutting stock problem of structural metallic tubes that arises in the manufacturing of agricultural light aircrafts. Two models are presented; Model 1 is a mixed integer linear programming formulation of a special case of the mathematical model in Gradisar et al. (1997). Model 2 is a simplification of Model 1, it does not have the shortcoming of Model 1 discussed in Sect. 3 and it outperforms Model 1 in the empirical experiments of this study. Both models minimize the total trim loss considering the possibility of generating leftovers for future reuse.

Using a modeling language (GAMS) with an optimization solver (CPLEX), computational experiments with actual data provided by the company were performed comparing the solutions of the models and the solutions used by the company. The results show that the two models outperform the company's method and Model 2 performs better than Model 1. On average, the model solutions are about 55% better than the company solutions in terms of trim loss, requiring 3.1% less raw material (objects) and generating approximately around 36% less leftovers. These savings are significant in practice and show the usefulness of mathematical programming techniques in a practical application via computational experiments. Other results are presented illustrating the trade-off between the total trim loss and the maximum number of leftovers allowed, and the possible cycles of the lengths of the leftovers generated in consecutive periods under the assumption of constant demand.

An interesting perspective for future research would be to study the definition of the minimum length N which defines a leftover, considering the cost-benefits between the inventory management of object remainders and their real value for reuse in the cutting process of the company. Recall that parameter N is an input data of the models simply defined by the company as the smallest item length of the order list. This is the current practice of the cutting process of the company.

In case the company increases its production rate of airplanes, another interesting extension of this research would be to study the trade-off between the productivity of the cutting equipment and the waste of material discarded in the cutting process. The company could speed up the production (cutting) of items if the cutting patterns were simpler to be executed by the cutting equipment, at the expense of increasing the trim loss. Another interesting line of research would be to analyze the application of extended models to deal with multi-period cutting planning (as Model 2 with multiple periods presented in Sect. 3), taking into account the benefits of advancing the cutting of items and keeping them in stock, instead of generating and stocking leftover objects in order to meet future demands.

Acknowledgements The authors would like to thank the anonymous referees for their useful comments and suggestions, and Silvia Morales and Neiva/Embraer for their help with this study. This research was partially supported by CAPES and CNPq.

References

Abuabara, A. (2006). *Otimização no corte de tubos estruturais: aplicação na indústria aeronáutica agrícola*. Dissertation, Departamento de Engenharia de Produção, Universidade Federal de São Carlos, Brazil.

Arenales, M., Morabito, R., & Yanasse, H. (Eds.) (1999). Special issue on cutting and packing problems. *Pesquisa Operacional*, *19*(2), 107–299.

Bischoff, E., & Wäscher, G. (1995). Special issue on cutting and packing. *European Journal of Operational Research*, *84*(3), 503–505.

Brooke, A., Kendrik, D., Meeraus, A., & Rosenthal, R. (1998). *GAMS: a user's guide*. Washington: GAMS Development Co.

Dyckhoff, H. (1981). A new linear programming approach to the cutting stock problem. *Operations Research*, *29*, 1092–1104.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, *44*, 145–159.

Dyckhoff, H., & Finke, U. (1992). *Cutting and packing in production and distribution: typology and bibliography*. Heidelberg: Springer.

Dyckhoff, H., Scheithauer, G., & Terno, J. (1997). Cutting and packing. In M. Amico, F. Maffioli, & S. Martello (Eds.), *Annoted bibliographies in combinatorial optimization* (pp. 393–414). New York: Wiley.

ESICUP (2007). *Euro special interest group on cutting and packing.* Available in: http://paginas.fe.up.pt/~esicup/.

Gradisar, M., Jesenko, J., & Resinovic, C. (1997). Optimization of roll cutting in clothing industry. *Computers & Operational Research*, *10*, 945–953.

Gradisar, M., Resinovic, C., & Kljajic, M. (1999). Kljajic, M,. A hybrid approach for optimization of one-dimensional cutting. *European Journal of Operational Research*, *119*, 719–728.

Gradisar, M., Resinovic, C., & Kljajic, M. (2002). Evaluation of algorithms for one-dimensional cutting. *Computers & Operations Research*, *29*, 1207–1220.

Hifi, M. (2002). Special issue: cutting and packing problems. *Studia Informatica Universalis*, *2*(1), 1–161.

Mukhacheva, E. A. (Ed.) (1997). *Decision making under conditions of uncertainty: cutting–packing problems.* The International Scientific Collection, Ufa, Russia.

Neiva (2006). *Indústria Aeronáutica Neiva Ltda.* Available in: http://www.aeroneiva.com.br.

Oliveira, J. F., & Wäscher, G. (Eds.) (2007). Cutting and packing. *European Journal of Operational Research*, *183*(3), 1106–1108.

Wang, P. Y., & Wäscher, G. (Eds.) (2002). Special issue on cutting and packing. *European Journal of Operational Research*, *141*(2), 239–469.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, *183*(3), 1109–1130.

Williams, H. P. (1978). *Model building in mathematical programming*. New York: Wiley.