Computers & Industrial Engineering 67 (2014) 195–203

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Production scheduling of assembly fixtures in the aeronautical industry ☆

Bruno Jensen Virginio da Silva [a], Reinaldo Morabito [a,*], Denise Sato Yamashita [a], Horacio Hideki Yanasse [b]

[a] Departamento de Engenharia de Produção, Universidade Federal de São Carlos, Brazil
[b] Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, Brazil

ARTICLE INFO

Article history:
Received 3 April 2012
Received in revised form 5 September 2013
Accepted 11 November 2013
Available online 22 November 2013

Keywords:
Production scheduling
Assembly fixture
Aeronautical industry
Adjacency constraints

ABSTRACT

In this work, we study the production scheduling of a real world assembly problem present in the aeronautical industry. Parts of aircrafts should be produced on fixtures, which are commonly used in aircraft manufacturing and consist of several workstations. Due to a lack of physical space in the fixture, when a workstation is in use, other workers cannot use adjacent workstations in this fixture. These constraints are called here adjacency constraints. This assembly fixture scheduling problem is studied in the context of a workforce learning process including four main qualification stages (or epochs). Mathematical models are developed and implemented for each stage using a modeling language and an optimization solver. Computational experiments with this approach were performed in a case study of a Brazilian aeronautical company and they resulted in better solutions than those currently practiced in the company.

1. Introduction

In this paper, we address the production scheduling of assembly fixtures that arise in aircraft manufacturing. We are particularly interested in the production of a specific subset of aircraft parts that make use of fixtures with adjacency constraints in order to be assembled. A fixture is a workholding or support device built to fit a particular part or shape. The main purpose of a fixture is to locate and in some cases hold a workpiece during either a machining operation or some other industrial process (Niu, 1988; Drake, 1989, Howe, 2004). Due to a lack of space, adjacent workstations in the fixture cannot be occupied simultaneously, resulting in adjacency constraints. In order to assemble an aircraft part, at least two operations are necessary. One of them is done on the fixture and a second complementary assembly operation is carried out on a workbench. Depending on the part being assembled, these two operations should be repeated more than once.

In this study, the scheduling of these operations must take into account another relevant and rather practical issue faced by aircraft manufacturers: the development of the assembly process of the parts of a new aircraft model. It is well known that the repetitive production of airplane assemblies can result in different cost and time estimates, as pointed out by different authors since the seminal study in Wright (1936), who introduced the concept of the learning curve to the aircraft industry. In this paper, we are also interested in the impact of workforce learning or qualification process in the assembly of specific subsets of aircraft parts. This assembly learning process can be divided into four main stages (epochs), where the first stage is concerned with the development of the prototype and the elaboration of the production process. In the other three stages, the airplane is already in serial production and what differentiates these three stages from each other is the presence of assembly teams with different skills.

The literature on assembly fixture scheduling in the aeronautical industry is relatively scarce. To the best of our knowledge, we are not aware of other related papers that have studied this assembly problem considering the adjacency constraints. Structural assembly scheduling in the aeronautical industry was also analyzed in Heike, Ramulu, Sorenson, Shanahan, and Moinzadeh (2001) and Scott (1994). Other examples of applications of operations research techniques in the aeronautical industry can be found in Dale (2001), Chikong, Chang, and Lin (2006) and Abuabara and Morabito (2009). However, none of them considered the assembly fixture scheduling problem. Preliminary studies of this problem were presented in Silva, Morabito, and Yanasse (2011), where a mathematical model based on job shop scheduling was proposed for stage 1, and Silva, Morabito, Yamashita, and Produção (2013), where a project scheduling based model was investigated. The present paper can be seen as an extension of this line of research.

The problem of scheduling with adjacency constraints is present in other areas besides the aeronautical industry. Santos, Michelon, Arenales, and Santos (2008) studied the crop rotation

☆ This manuscript was processed by Area Editor Subhash C. Sarin.
* Corresponding author. Tel.: +55 16 33519516; fax: +55 16 33518240.
E-mail addresses: brunojen@gmail.com (B.J.V. da Silva), morabito@ufscar.br (R. Morabito), denisesy@dep.ufscar.br (D.S. Yamashita), horacio.yanasse@unifesp.br (H.H. Yanasse).

0360-8352/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2013.11.009

scheduling problem, and proposed a column generation procedure and a greedy constructive algorithm to deal with the problem. Weintraub et al. (2007) presented mixed integer linear models for the forest harvest scheduling problem, considering the constraint of harvesting adjacent areas. Gandham, Dawande, and Prakash (2008) studied adjacency constraints caused due to the existence of hidden nodes in telecommunication networks using algorithms to solve the problem. Duin and Van der Sluis (2006) dealt with the complexity of adjacent resource scheduling in the assignment of desks during airport passenger check in process. Irani and Leung (1996, 2003) used graph theory based algorithms to study adjacency constraints in the context of traffic light scheduling. The solution approaches used for these problems were developed specifically for them and therefore, cannot be directly applied to the assembly fixture scheduling problem with adjacency constraints studied here.

In the present work, we propose different mathematical models for this problem for specific subset of parts of an aircraft. The workforce learning/qualification effect on the assembly of the aircraft parts is considered in the models. They were implemented and solved using the modeling language GAMS (Brooke, Kendrickd, & Rosenthal, 1998) and the optimization software CPLEX (ILOG, 2011). In order to validate the models, computational tests were performed with real data from a Brazilian aircraft manufacturer. This work is organized as follows: Section 2 presents the literature review, Section 3 presents the mathematical models developed for each production stage, Section 4 presents the computational results of the models and Section 5 discusses the results and perspectives for future research.

2. Problem definition

For the sake of illustration, we took an example of a subset (or subassemblies) formed by eight parts, and for each part, there is a unique correspondent workstation in the assembly fixture. The parts are identified by numbers 1 to 8, following their assembly order in the final subset. Since each airplane needs two whole subsets in order to be assembled, it is necessary to have at least two parts of each of the eight parts that compose the subset. Fig. 1 shows the parts of the subset.

The fixture to assemble this subset has eight workstations, numbered 1 to 8, where the number of the workstation identifies the part that is assembled there. For example, part 1 is assembled at workstation 1, part 2 is assembled at workstation 2, and so on. Fig. 2 illustrates this fixture. An interesting feature of the assembling fixtures considered in this study is the presence of adjacency constraints, i.e., two workers are not able to use two adjacent workstations of the fixture due to a lack of physical space. For example, assume that part 4 is being assembled at workstation 4. Then, workstations 3 and 5 are unavailable until part 4 leaves the workstation.

In order to assemble a part, at least two operations are necessary. First, the part is processed in the fixture, and then, there is a second complementary assembly operation at the workbench. In this work, these pairs of operations are called *tasks*. We note that some parts require just one task, while others require that these tasks are repeated more than once. The number of times these tasks are repeated depend on the development of the productive process and it can change during the serial production of the airplane. Fig. 3 shows an example of the precedence relationships of the operations necessary to assemble the parts of the subset under consideration. The first number in the balloons indicates the task and the second number just after the dot indicates the type of operation (i.e., 1 for the operation in the fixture, 2 for the operation at the workbench), so that "1.2" indicates task 1 under operation at the workbench.

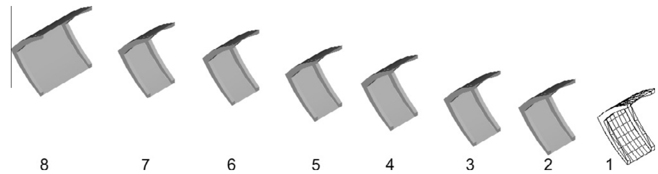

Fig. 1. Subset and its respective parts.

The assembly of the subset is considered in 4 stages, according to specific characteristics due to the different expertise levels of the assembly team. In stage 1, the production process is not well detailed yet, and the production processing times are estimated or simulated in computers based on the assembly time of few airplanes. In this stage, the main objective of the production manager is to determine the maximum capacity of the assembly fixture and the size of the teams for different production rates. In stage 2, the serial production starts. In this stage, there are only a handful of workers who are able to operate the assembly fixture. These are the same workers who took part in the development of stage 1, the prototype stage of the airplane. The other workers are limited to the workbench operations. Therefore, the team is formed by these two groups of workers: the specialized group, who is able to make operations only in the assembly fixture, and the non-specialized, who is able to work only at the workbench. In stage 3, the specialized workers can operate the assembly fixture and the workbench as well, but the non-specialized workers can only execute operations at the workbench. Finally, in stage 4, all workers are in the same level of expertise and are able to execute operations both in the fixture and the workbench as well. Below, we describe the mathematical models developed for representing each stage.

3. Modeling stages

3.1. Stage 1

The objective of stage 1 is to find out the maximum production capacity if just one assembly fixture is used, regardless of the number of workers available, i.e., to minimize the production *makespan* considering an unlimited number of workers. The following model is based on the job shop scheduling problem as presented in Coffman (1976), Baker (1974), Morton and Pentico (1993), Leung (2004) and Pinedo (2008); see also Potts and Strusevich (2009). Let J be the number of tasks, and for each task $j = 1, 2, \ldots, J$:

p_j: the duration of the operation of task j in the assembly fixture, and
q_j: the duration of the operation of task j at the workbench.

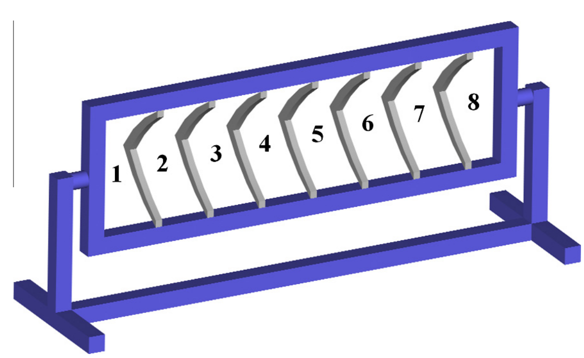

Fig. 2. Fixture with adjacency constraints used to assemble the subset.

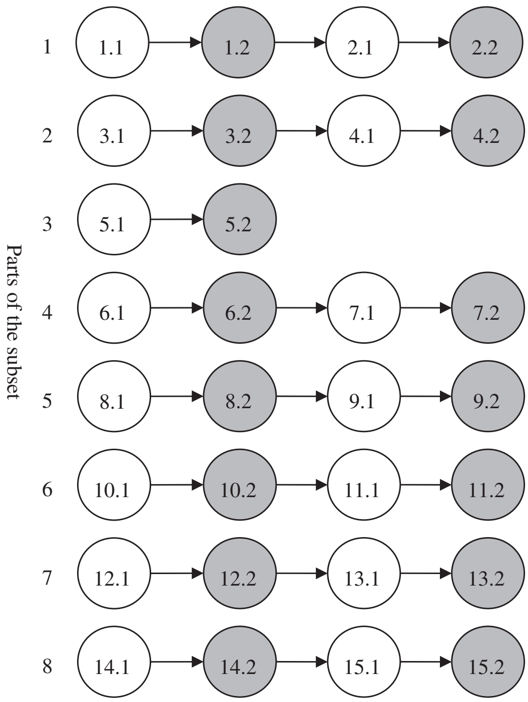

Fig. 3. Precedence relationships of the operations of the subset.

Moreover, the following sets for each pair of tasks $(j,k), j \neq k$, are defined:

$A = \{(j,k)|$task j has to finish in the fixture before task k starts$\}$;
$B = \{(j,k)|$tasks j and k are executed in adjacent workstations in the fixture$\}$;
$C = \{(j,k)|$tasks j and k are executed in the same fixture position$\}$.

The decision variables of the models are:

$$y_{jk} = \begin{cases} 1 & \text{if task } j \text{ is scheduled before task } k \\ 0 & \text{otherwise} \end{cases};$$

t_j starting time of task j;
t_F production makespan.

The mathematical model for stage 1 is:

$$\text{Min}\, t_F \tag{1}$$

$$t_j + p_j + q_j \leqslant t_F \quad j = 1, \ldots, J \tag{2}$$

$$t_k + M(1 - y_{jk}) \geqslant t_j + p_j \quad \text{For all } (j,k) \in C \tag{3}$$

$$t_j + \text{My}_{jk} \geqslant t_k + p_k \quad \text{For all } (j,k) \in C \tag{4}$$

$$t_k \geqslant t_j + p_j + q_j \quad \text{For all } (j,k) \in A \tag{5}$$

$$t_k + M(1 - y_{jk}) \geqslant t_j + p_j \quad \text{For all } (j,k) \in B \tag{6}$$

$$t_j + \text{My}_{jk} \geqslant t_k + p_k \quad \text{For all } (j,k) \in B \tag{7}$$

$$\text{tj} \geqslant 0, y_{jk} \in \{0,1\}, j = 1, \ldots, J, \quad k = 1, \ldots, J, \quad j \neq k \tag{8}$$

In model (1)–(8), the objective function (1) minimizes the production makespan t_F and constraints (2) ensure that the makespan is not smaller than the finishing time of any scheduled task. Constraints (3) and (4) avoid task overlapping in the same assembly workstation – note that these constraints are defined only for pairs of tasks j and k assembled at the same workstation. Note also that constraints (3) and (4) are disjunctive – when one is active, the other is redundant and vice versa (Williams, 1999). Parameter M is a sufficient large positive number that can be defined as: $\sum_{j=1}^{J}(p_j + q_j)$. Constraints (5) ensure that the precedence relationships between the tasks are respected. Constraints (6) and (7) ensure that tasks that use adjacent workstations in the fixture are not scheduled at the same time. Constraints (8) refer to the domain of the decision variables.

In cases where some of the tasks may have release times greater than zero, and/or cases where some of the tasks may have due dates (e.g., in cases where the tasks are related to more than one airplane with different due dates), we can define:

r_j: the release time when task j is available to be processed, and
d_j: the due date of task j

and the model above can be easily adapted including the following constraints:

$$t_j \geqslant r_j \quad \text{for some } j \tag{9}$$

$$t_j + p_j + q_j \leqslant d_j \quad \text{for some } j \tag{10}$$

where constraints (9) ensure that each task j is not scheduled before its release time r_j, and constraints (10) ensure that each task j is completed before its due date d_j. Note also that parameter M should be modified accordingly in order to consider the non-null release times r_j.

3.2. Stage 2

In this stage, two assembly teams are considered: the assembly fixture team and the workbench team. The objective is to minimize the total costs or the amount of workers so that the parts are finished before their given due dates. After some changes, it is possible to model this problem as a time constrained project scheduling problem. A project consists of activities where each activity needs one or more resources in order to be executed. An activity has a known duration and usually, there are precedence relationships among them due to technological constraints. The project scheduling problem studied in this paper considers limited and renewable resources, i.e., once used, it becomes available to be used again (examples of renewable resources are workforce, machines, etc.). The objective is to find a feasible schedule for the activities that minimize the number of resources used, respecting the precedence relationships among the activities, the amount of resources available, and a given due date for the project. In the literature of project scheduling, this problem is known as the time constrained project scheduling problem (Brucker, Drexl, Mohring, Neumann, & Pesch, 1999; Yamashita & Morabito, 2009). Reviews of project scheduling can be found, e.g., in Brucker et al. (1999), Hartmann and Kolisch (2000), Kolisch and Padman (2001) and Guldemond, Hurink, Paulus, and Schutten (2008).

It is not difficult to make a correspondence between a project scheduling problem and the scheduling problems with adjacency constraints studied here. Each assembly operation of a part corresponds to an activity, where the time spent to do the operation (either in the fixture or at the workbench) is the duration of the activity, and the sequence that these operations must follow (see Fig. 3) are the precedence relationships among the activities. Contrary to the model for stage 1, where two operations correspond to a task, an operation corresponds to an activity in this model. Ten resource types are used in the model:

(a) workers of the fixtures;
(b) workers of the workbench;
(c) each of the 8 workstations of the fixture.

In this stage, the number of workers of the fixture and in the workbench are variables of the problem. On the other hand, the resource types that correspond to the number of workstations available for use are parameters of the problem. In order to correctly represent the adjacency constraints in the model, it is artificially assumed that each resource type that corresponds to a fixture workstation has a resource availability of 2 units, and that an activity to be processed in this fixture workstation artificially requires two units of resources from this workstation, plus one unit of resource from each of the two adjacent workstations. Fig. 4 illustrates an example where activity 1 is assembled at workstation 2. It uses two resource units from workstation 2, one resource unit from the adjacent workstation 1 and one resource unit from the other adjacent workstation 3. Similarly, activity 2 needs 2 resource units from workstation 4, one resource unit from workstation 3 and one resource unit from workstation 5. Therefore, workstations 1, 3 and 5 would be blocked by the adjacency constraints and would not be used simultaneously with workstations 2 and 4 when processing activities 1 and 2. However, any activity that uses workstations 6, 7 or 8 can be processed simultaneously.

Furthermore, the total scheduling period is represented by multiple discrete time periods $t = 1,\ldots,T$, where T corresponds to the production cycle time of each airplane, according to the production rates. The additional parameters of the model are:

J: number of activities;
T: number of time periods (time horizon);
K: number of workstations in the fixture;
W: number of types of assembly crew;
j: activity, $j = 1,\ldots,J$;
t: time $t = 1,\ldots,T$;
k: resource type, corresponding to the workstations of the fixture, $k = 1,\ldots,K$;
w: type of assembly crew, $w = 1,\ldots,W$;
c_{jk}: amount of type k resource (workstations of the fixture) necessary to execute activity j. Note that $c_{jk} = 0$ means that activity j is not assembled in workstation k;
C_k: amount of type k resource (workstations of the fixture) available;
p_j: duration of activity j (for both the operations of the fixture and workbench);
n_{jw}: number of type w workers necessary to execute activity j;
υ_w: unitary cost of a type w worker of the assembly team.

Additionally, for each pair of activities (h,j) where $h \neq j$, the following sets are defined:

$H = \{(h,j)|\text{activity } h \text{ precedes activity } j\}$;
$G = \{(j)|\text{activity } j \text{ is executed in the fixture, no matter in which of the workstations of the fixture}\}$.

The decision variables of this model are:

$$x_{jt} = \begin{cases} 1 & \text{if activity } j \text{ finishes at time } t \\ 0 & \text{otherwise} \end{cases};$$

a_w: amount of type w resources (workers) required.
In particular, since the model is used to solve stage 2, $W = 2$, and

a_1: is the amount of workers required in the fixture;
a_2: is the amount of workers required at the workbench.

Then, the mathematical model for stage 2 is:

$$\text{Min} \sum_{w=1}^{W} \upsilon_w a_w \tag{11}$$

$$\sum_{t=1}^{T} x_{jt} = 1 \quad j = 1,\ldots,J \tag{12}$$

$$\sum_{t=r_h+p_h}^{d_h} t \cdot x_{ht} \leqslant \sum_{t=r_j+p_j}^{d_j} (t - p_j) x_{jt} \quad \forall (h,j) \in H \tag{13}$$

$$\sum_{j \in G} \sum_{b=t}^{\min\{t+p_j-1;T\}} c_{jk} x_{jb} \leqslant C_k \quad \begin{array}{l} k = 1,\ldots,K \\ t = 1,\ldots,T \end{array} \tag{14}$$

$$\sum_{j=1}^{J} \sum_{b=t}^{\min\{t+p_j-1;T\}} n_{jw} x_{jb} \leqslant a_w \quad \begin{array}{l} w = 1,\ldots,W \\ t = 1,\ldots,T \end{array} \tag{15}$$

$$x_{jt} \in \{0,1\}; \quad a_w \in Z^+, \quad j = 1,\ldots J; \quad t = 1,\ldots,T; \quad w = 1,\ldots,W \tag{16}$$

In this model, the objective function (11) minimizes the total cost of workers employed, constraints (12) ensure that each activity is scheduled exactly once during the time horizon considered, constraints (13) ensure that the precedence relationships among the activities are respected, constraints (14) ensure that the amount of resources used at each time period does not exceed what is available in each period, constraints (15) ensures a sufficient amount of workers to perform the activities at each time period, and finally, constraints (16) refer to the domain of the decision variables.

Note that for each time period t, the left hand sides of constraints (14) and (15) consider all activities j that start before period t and finish after (or at) period t, i.e., that finish between periods t and $t + p_j - 1$, as illustrated in Fig. 5. If an activity is not in process during period t, like activities j' and j'' in the figure, they are not accounted for in the left hand sides of constraints (14) and (15) for period t.

Consider the following parameters:

EFj: earliest finish time that activity j can finish, respecting the precedence relationships and without taking into account resource constraints;

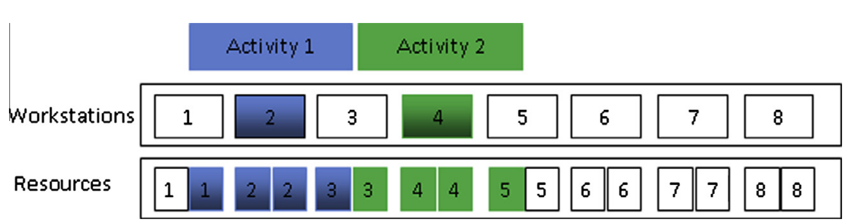

Fig. 4. Representation of resources and use of the workstations of the assembly fixture.

LF_j: latest finish time that at which activity j can be completed without delaying the project and respecting the precedence relationships, without taking into account resource constraints.

In cases where some of the activities may have release times greater than zero, and/or cases where some of the activities may have due dates smaller than the makespan (e.g., in cases where the activities are related to more than one airplane with different due dates), we can define:

r_j: the time when activity j is available to be processed ($r_j \geqslant EF_j - p_j$);
d_j: the due date of activity j ($d_j \leqslant LF_j$);

and the model above can be easily adapted deleting constraints (13) and including the following constraints:

$$\sum_{t=max(EF_h,r_h+p_h)}^{\min(LF_h-1,d_h-1)} t \cdot x_{ht} \leqslant \sum_{t=max(EF_j,r_j+p_j)}^{\min(LF_j-1,d_j-1)} (t-p_j)x_{jt} \quad \forall (h,j) \in H \tag{17}$$

$$x_{jt} = 0 \quad \begin{array}{l} t = 1,\ldots,EF_j - 2 \\ t = LF_j + 1,\ldots,T \\ \quad j = 1,\ldots,J \end{array} \tag{18}$$

where constraints (17) ensure that the precedence relationships among the activities are respected and constraints (18) ensure that the activities are scheduled after their release date and before their due date. These will eliminate the variables with a known value before the model is solved and improve the computational time spent to solve it.

Model (11)–(16) can easily be adapted to represent model (1)–(9), in discrete time. In this case, since in stage 1 there is not a constraint to the workforce, constraints (15) can be deleted and, the objective function (11) is written as:

$$min \sum_{t=1}^{T} t.x_{Jt}$$

where J is the last activity of the precedence net. Nevertheless, this model was not considered because the time continuous model presents a shorter computational execution time.

3.3. Stage 3

In stage 3, the workers are divided into two teams, the specialized team of workers who can operate in the fixture and at the workbench as well, and the second team of workers who can operate only at the workbench. Additionally to the parameters presented in the model of stage 2, consider the following parameters:

s_{jw}: amount of resources (workers) of type w necessary to execute activity (operation) j in the fixture;
υ_{jw}: amount of resources (workers) of type w necessary to execute activity (operation) j at the workbench.

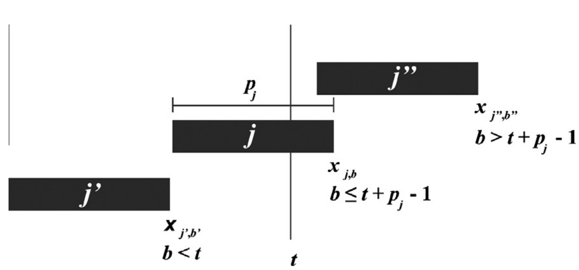

Fig. 5. Activities which are not in parallel during t.

All operations j of the problem studied in this paper use only one worker during the processing. Therefore, s_{jw} and υ_{jw} are equal to 1 for all operations. The decision variables for stage 3 are:

$$x_{jt} = \begin{cases} 1 & \text{if activity } j \text{ finishes at time instant } t \\ 0 & \text{otherwise} \end{cases};$$

a_w: amount of workers of team w required, where for stage 3, ($W = 2$):
a_1: amount of workers required that can execute operations in the assembly fixture and at the workbench;
a_2: amount of workers required that can execute operations at the workbench only.

Note that constraints (23) and (24) make use of a slack variable, y_t, that represents the slack of the specialized team at each time instant t that can be used at the workbench. The mathematical model for stage 3 is:

$$\text{Min} \sum_{w=1}^{W} \upsilon_w a_w \tag{19}$$

$$\sum_{t=1}^{T} x_{jt} = 1 \quad j = 1,\ldots,J \tag{20}$$

$$\sum_{t=EF_h}^{LF_h-1} t \cdot x_{ht} \leqslant \sum_{t=EF_j}^{LF_j-1} (t-p_j)x_{jt} \quad \forall (h,j) \in H \tag{21}$$

$$\sum_{j \in G} \sum_{b=t}^{\min\{t+p_j-1;T\}} c_{jk}x_{jb} \leqslant C_k \quad \begin{array}{l} k = 1,\ldots,K \\ t = 1,\ldots,T \end{array} \tag{22}$$

$$\sum_{j \in G} \sum_{b=t}^{\min\{t+p_j-1;T\}} s_{jw}x_{jb} = a_w - y_t \quad W = 1, t = 1,\ldots,T \tag{23}$$

$$\sum_{j \in J \setminus G} \sum_{b=t}^{\min\{t+p_j-1;T\}} \upsilon_{jw}x_{jb} \leqslant a_w + y_t \quad W = 2, \quad t = 1,\ldots,T \tag{24}$$

$$x_{jt} = 0 \quad \begin{array}{l} t = 1,\ldots,EF_j + p_j - 2 \\ t = LF_j + 1,\ldots,T \\ j = 1,\ldots,J \end{array} \tag{25}$$

$$x_{jt} \in \{0,1\}; \quad a_w \in Z^+; \quad y_t \geqslant 0; \quad j = 1,\ldots,J; \\ t = 1,\ldots,T; \quad w = 1,\ldots,W \tag{26}$$

The objective function (19) minimizes the total cost (or amount) of workers available, constraints (20) ensure that all activities are scheduled exactly once, constraints (21) ensure that the precedence relations among the activities are obeyed, constraints (22) ensure that the adjacency constraints and the workstations occupation are respected, constraints (23) ensure that the amount of fixture workers is enough to perform the activities at the fixture, constraints (24) ensure that the amount of workers at the workbench is enough to perform the activities at the workbench, constraints (25) fix the variables with known values, constraints (26) refer to the domain of the variables. Note that in constraints (23) and (24), x_{jb}, a_w, s_{jw} and υ_{jw} are integers, ensuring the integrality of y_t. Therefore, it is sufficient to impose that the y_t are non-negative real numbers in (26).

3.4. Stage 4

In stage 4, all workers of the assembly team can operate in the fixture and at the workbench as well. Since there is no distinction among the workers' skills, the mathematical model developed for stage 2 can also be applied to this stage by considering just one type of worker ($W = 1$) in model (11)–(16). This stage is the final stage of the learning curve, where all workers are multi-task. In this stage, great efforts are required to achieve a small increase in productivity. The optimization of the size of the teams is more important than in the previous stages since it is in this stage that most of the airplanes are produced.

4. Computational results

The mathematical models were implemented in the modeling language GAMS (version 23.0) and solved with the application CPLEX 11.2.1.0, with CPLEX option for using multiple kernels set to 4. The computational tests were performed in a PC Intel i7 (2.8 GHz, 12 GB). The computational time was limited to 10 h or 600 min for all instances.

This instance consists of 15 activities where the workbench operations are longer than the fixture operations, in general. In order to protect the data privacy of the company, the time unit (t.u.) of the duration of the operations is not revealed and the real data were multiplied by a constant. It should be noted that each airplane needs two whole subsets in order to be assembled, therefore the number of activities is duplicated resulting in 30 activities.

Table 1 presents the data of the instance studied in this paper. Note that since the activities are duplicated, all the data in Table 1, such as duration, are also duplicated. The first three columns identify the subset, the part number, and the activity number, respectively. The fourth column shows the workstation where activity (j) is performed, the next two columns refers to the duration of the activities in the fixture and workbench. Finally, the last column of Table 1 shows the activity that precedes activity j.

In the first set of experiments, the instance presented in Table 1 was solved using the model proposed for stage 1. It is worth noting that the solution obtained in stage 1, when compared to the solutions obtained by the models of the other stages, corresponds to the shortest time necessary to assemble two subsets. Since the smallest production rate, ever registered in practice, to execute the project was 288 time units (t.u.), then it was assumed that all instances solved in stages 2, 3 and 4 have a completion time between the smallest makespan, i.e., the solution found in stage 1, and 288 time units.

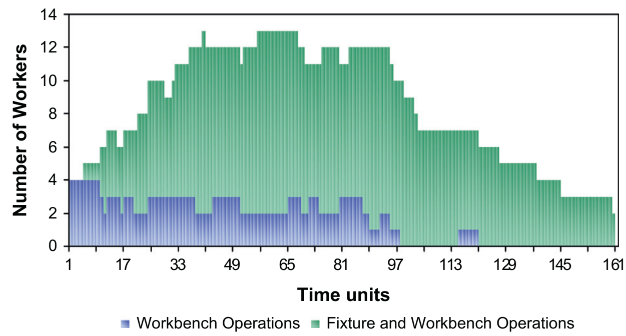

Fig. 6. Solution of model of the stage 1.

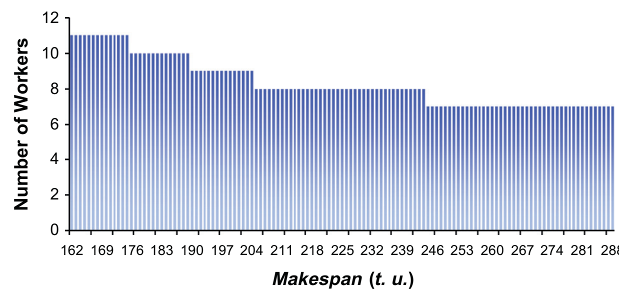

Fig. 7. Optimized size of the workers' team versus the makespan value, for stage 2 model (11)–(16).

Table 1
Input data of the instance studied.

Subset	Part of the subset	Activity (j)	Workstation at the fixture	Duration at the fixture (p_j)	Duration at the workbench (q_j)	Precedent activities ($A(j,k)$)
1	1	1	1	5	7	–
1	1	2	1	4	47	1
2	1	3	1	5	7	–
2	1	4	1	4	47	3
1	2	5	2	15	29	–
1	2	6	2	9	71	5
2	2	7	2	15	29	–
2	2	8	2	9	71	7
1	3	9	3	11	40	–
2	3	10	3	11	40	–
1	4	11	4	12	40	–
1	4	12	4	6	40	11
2	4	13	4	12	40	–
2	4	14	4	6	40	13
1	5	15	5	8	30	–
1	5	16	5	6	40	15
2	5	17	5	8	30	–
2	5	18	5	6	40	17
1	6	19	6	12	17	–
1	6	20	6	6	40	19
2	6	21	6	12	17	–
2	6	22	6	6	40	21
1	7	23	7	7	12	–
1	7	24	7	7	30	23
2	7	25	7	7	12	–
2	7	26	7	7	30	25
1	8	27	8	10	61	–
1	8	28	8	12	66	27
2	8	29	8	10	61	–
2	8	30	8	12	66	29

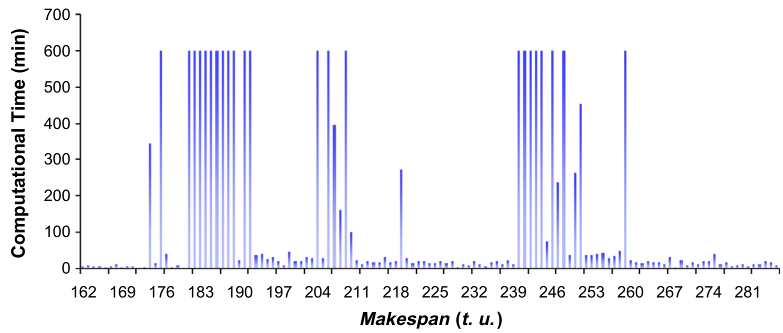

Fig. 8. Computational time obtained, for each makespan value, using stage 2 model (11)–(16).

4.1. Results for Stage 1

The model in stage 1 resulted in 278 integer variables, 310 real variables and was solved in 0.017 min. The smallest makespan was equal to 161 t.u., with 13 workers, as can be seen in Fig. 6. Note that this solution is not leveled because the objective function is to minimize the makespan while in the other stages, the objective function is to minimize the total number of workers. Therefore, using the models of stages 2, 3 or 4, depending on the skills of the team, it is possible to level the amount of workers to complete all the tasks using 161 t. u. In the case of stage 2, the total number of workers would be reduced from 13 to 11 workers.

In order to evaluate the performance of the models proposed for stages 2, 3 and 4, 127 instances were solved, with makespan value varying from 161 t.u. to 288 t.u., based on the data shown in Table 1.

4.2. Results for stage 2

In stage 2, the parts are assembled by two teams: one is responsible for the operations in the assembly fixture and the other is responsible for the operations at the workbench. The solution obtained considering the smallest makespan resulted in 11 workers, while the solution obtained considering a makespan value of 288 time units resulted in 7 workers. Fig. 7 shows the amount of workers necessary for each makespan value considered in stage 2. The resulting labor/makespan tradeoff curve shows the effect of various makespan values on the size of the team, presenting a pattern with discrete levels. This information can be used to quickly find out the minimum size of the assembly team for any makespan scenario. It is also helpful for analyzing the capacity reaction of the team, given by the distance of one step to the next, i.e., what is the risk when a certain size for the team is chosen. For instance, in Fig. 7, take the makespan value of 188 t.u. that needs 9 workers. If a variation in the production process calls for a shorter assembly time, i.e., a smaller makespan, then it will be necessary to hire an extra worker. However, considering a makespan value of 204 time units, if shorter assembling times were necessary, it would be possible to achieve them without the need to hire extra workers for makespan values as small as 188 time units.

Fig. 8 shows the computational time spent to obtain the solutions presented in Fig. 7. Note that, although some instances reached the maximum computational time limit of 600 min, most of them finished before 100 min. If an instance exceeded the maximum allowed computational time, it is possible to obtain upper and lower bounds for this instance examining proven optimal solutions of instances that have makespan values, respectively, smaller and larger than the current instance. Obviously, if the lower and upper bounds have the same optimal value, i.e., they need the same amount of workers, then this is the optimal solution for the instance.

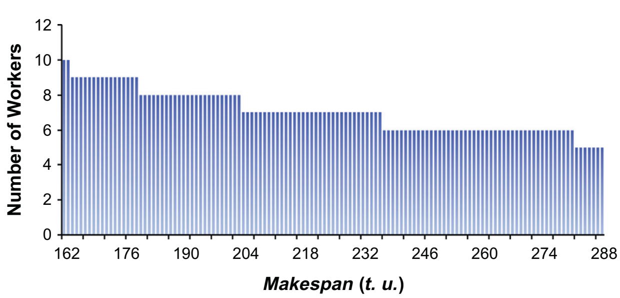

Fig. 9. Size of the team of workers versus the makespan value for stage 3 using model (19)–(26).

4.3. Results for stage 3

Two teams are considered in stage 3. The first team, formed by specialized workers, can operate in the assembly fixture and at the workbench as well, and the second team, made up of non-specialized workers, can operate only at the workbench. Therefore, we

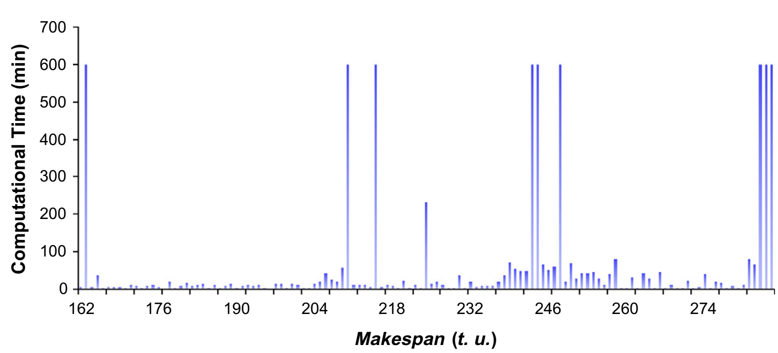

Fig. 10. Computational time for each makespan value, obtained with model (19)–(26) of stage 3.

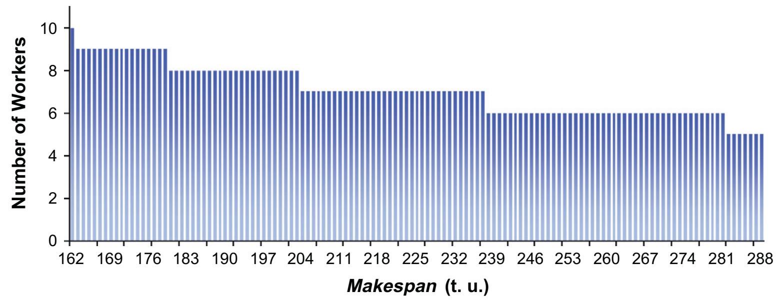

Fig. 11. Optimized size of the team of workers versus the makespan of the model (11)–(16).

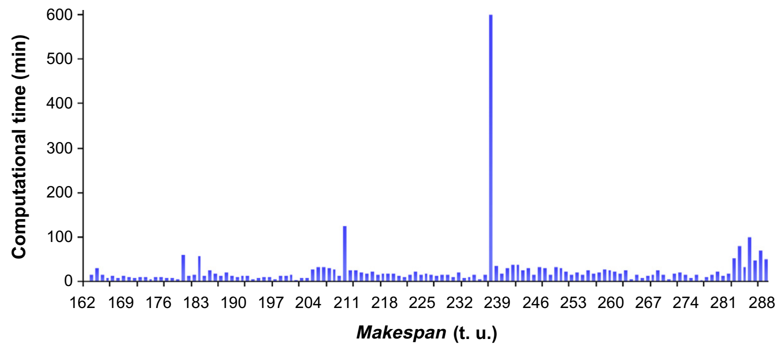

Fig. 12. Computational time for each makespan value, obtained with model (11)–(16) of stage 4.

expect to have a decrease in the amount of workers for the same makespan value when comparing stage 3 solutions with stage 2 solutions. For instance, considering a makespan value of 162 t.u., stage 2 found a solution with 11 workers while stage 3 found a solution with 10 workers. Fig. 9 shows the solutions of model (19)–(26).

The computational time spent to solve model (19)–(26) is presented in Fig. 10. Note that many instances reached the maximum allowed computational time limit. Note also that most instances were solved in less than 100 min, i.e., only a few of them reached the maximum computational time limit.

4.4. Results for stage 4

In stage 4, all workers are able to assemble parts both in the fixture and at the workbench. Therefore, the solutions obtained in stage 3 are upper bounds to the solutions of stage 4. Fig. 11 shows the labor curve for stage 4. For instance, by taking a makespan value of 162 time units, the solution of stage 4 results in 10 workers, i.e., the same amount of stage 3. Except for the makespan value of 163 t.u., where stage 4 found a better solution than stage 3, the other instances did not show a decrease in the amount of workers. Fig. 12 shows the computational time spent to find the solutions of stage 4.

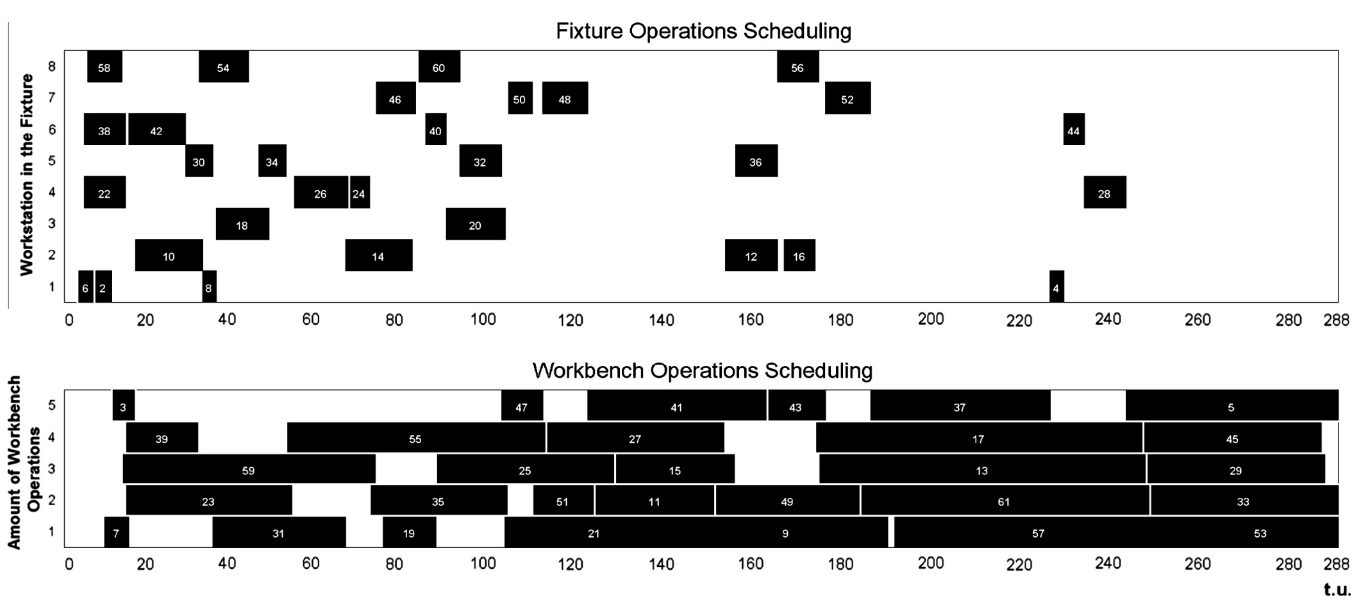

Fig. 13. Gantt chart of the production scheduling of stage 4 and makespan 288 t.u.

Fig. 13 shows an example of a schedule for makespan 288 t.u. Note that, the Gantt chart showing the schedule of the fixture has white areas, which corresponds to the time period where the workstations of the fixture are empty, due to adjacency or precedence constraints.

In Table 1, there are 30 activities to complete the aircraft assembly. Consider that each activity has two operations, one in the fixture and one in the workbench. Then, activity 1 in Table 1 is represented as operations 2 and 3 in Fig. 13, activity 2 as operations 4 and 5 and so on. Therefore, the total amount of operations is 62, with 30 operations in the fixture, 30 operations in the workbench and 2 artificial/fictitious operations, 1 and 62.

On the shop floor, the company used 8 workers with the makespan of 288, however the solution found, presented in Fig. 11, shows that it is possible to complete all activities with only 5 workers, i.e. 37% less workers than in practice.

5. Conclusion

In this work, we studied the production scheduling problem of assembly fixtures with adjacency constraints present in the aeronautical industry. Four different stages, concerning the four main workforce learning/qualification stages in the development of the assembly process, were analyzed and mixed integer linear programming models were proposed for each stage. The models were implemented with the optimization software GAMS/CPLEX, and solved in acceptable computational runtime. The numerical results were compared to real data obtained in practice. The proposed model found solutions with the number of workers up to 37% smaller than the original solution of the company.

Labor tradeoff curves were also generated for each stage. These curves can be useful as they show the effect of production plan changes, allowing a rapid response of production managers. This information is also useful for elaborating contingency plans, since information on how many people can be displaced or allocated in the assembly process is readily available.

An interesting line of research is to investigate other solution methods for this problem, based on heuristics and meta-heuristics, in order to solve larger problem instances which cannot be solved in a reasonable computer runtime with exact methods. Other objective functions, such as resource leveling, could also be incorporated into the models. Another interesting extension of this work would be to study the problem presented in this paper in the context of robust stochastic programming and robust optimization, for example considering uncertain durations for the assembly operations.

Acknowledgements

The authors wish to thank an anonymous reviewer for his/her useful comments and suggestions and the aeronautical company that cooperated in this research providing us with the necessary data collection and related information. This research was partially supported by research grants from CAPES, CNPq and FAPESP.

References

Abuabara, A., & Morabito, R. (2009). Cutting optimization of structural tubes to build agricultural light aircrafts. *Annals of Operations Research, 169*(1), 149–165.

Baker, K. (1974). *Introduction to sequencing and scheduling*. New York: John Wiley.

Brooke, A., Kendrickd, M. A. R. R., & Rosenthal, R. (1998). *GAMS, a useŕs guide* (1 ed.). Washington: GAMS Development Corporation.

Brucker, P., Drexl, A., Mohring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research, 112*, 3–41.

Chikong, H., Chang, H.-M., & Lin, C.-H. (2006). Simulation and analysis for layout of aerospace sheet metal process with lean concept. *Materials Science Forum, 505–507*, 907–912.

Coffman, E. G. Jr., (1976). *Computer and job shop scheduling theory*. New York: John Wiley.

Dale, R. (2001). Offline programming and simulation help Boeing use giant automated riveter on C-17 aircraft. *Industrial Robot: An International Journal, 28*(6), 478–482.

Drake, C. (1989). Proper fixturing. In *Metals handbook* (9th ed., Vol. 16, pp. 404–410). *Machining*. USA: ASM International.

Duin, C. W., & Van der Sluis, E. (2006). On the complexity of adjacent resource scheduling. *Journal of Scheduling, 9*, 49–62.

Gandham, S., Dawande, M., & Prakash, R. (2008). Link scheduling in wireless sensor networks: Distributed edge-coloring revisited. *Journal of Parallel and Distributed Computing, 68*, 1122–1134.

Guldemond, T. A., Hurink, J. L., Paulus, J. J., & Schutten, J. M. J. (2008). Time-constrained project scheduling. *Journal of Scheduling, 11*, 137–148.

Hartmann, S., & Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research, 127*, 394–407.

Heike, G., Ramulu, M., Sorenson, E., Shanahan, P., & Moinzadeh, K. (2001). Mixed model assembly alternatives for low-volume manufacturing: The case of the aerospace industry. *International Journal of Production Economics, 72*, 103–120.

Howe, D. (2004). *Aircraft loading and structural layout* (pp. 391–422). Reston, Virginia: American Institute of Aeronautics Inc. (AIAA).

ILOG (2011). <http://www-01.ibm.com/software/websphere/ilog/#> (Accessed 22.02.11).

Irani, S., & Leung, V. (1996). *Scheduling with conflicts and applications to traffic signal control*. Atlanta: ACM Symposium on Discrete Algorithms.

Irani, S., & Leung, V. (2003). Scheduling with conflicts on bipartite and interval graphs. *Journal of Scheduling, 6*, 287–307.

Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. *Omega: The International Journal of Management Science, 29*(3), 249–272.

Leung, J. Y. (2004). *Handbook of scheduling: Algorithms, models, and performance analysis*. Chapman & Hall.

Morton, T. E., & Pentico, D. W. (1993). *Heuristic scheduling systems: With applications to production systems and project management*. New York: John Wiley & Sons.

Niu, M. C. Y. (1988). *Airframe structural design: Practical design information and data on aircraft structures*. Hong Kong: Hong Kong Conmilit Press.

Pinedo, M. L. (2008). *Scheduling: Theory, algorithms, and systems*. New York: Springer.

Potts, C. N., & Strusevich, V. A. (2009). Fifty years of scheduling: A survey of milestones. *Journal of the Operational Research Society, 60*, 41–68.

Santos, L. M. R., Michelon, P., Arenales, M. N., & Santos, R. H. S. (2008). Crop rotation scheduling with adjacency constraints. *Annals of Operations Research, 190*(1), 165–180.

Scott, H. A. (1994). Modeling aircraft assembly operations. In *Proceedings of the 1994 winter simulation conference*.

Silva, B. J. V., Morabito, R., & Yamashita, D. S. (2013). *Otimização na programação de montagens na indústria aeronáutica. Gestão & Produção*. <http://dx.doi.org/10.1590/S0104-530X2013005000002>.

Silva, B. J. V., Morabito, R., & Yanasse, H. H. (2011). Programação da produção de gabaritos de montagem com restrições de adjacência na indústria aeronáutica. *Produção, 21*, 696–707.

Weintraub, A., Romero, C., Bjorndal, T., & Epstein, R. (Eds.) (2007). International series in operations research and management science. In *Handbook of operations research in natural resources* (Vol. 99). New York: Springer.

Williams, H. P. (1999). *Model building in mathematical programming*. New York: John Wiley & Sons Inc.

Wright, T. (1936). Factors affecting the cost of airplanes. *Journal of Aeronautical Science, 4*(4), 122–128.

Yamashita, D. S., & Morabito, R. (2009). A note on time/cost tradeoff curve generation for project scheduling with multi-mode resource availability costs. *International Journal of Operational Research, 5*(4), 429–444.