Dynamic Linear Assignment for Pairing Two Parts in Production
A Case Study in Aeronautics Industry

Clémence Bisot
Safran Aircraft Engines, Rond Point René Ravaud, 77550 Moissy Cramayel, France

Keywords: Dynamic Linear Assignment, Linear Optimization, Pairing, Myopic Assignment, Dynamic Assignment, Aeronautics Industry.

Abstract: In some manufacturing industries, the task of assembling two parts is a time-consuming step in production. Bonding can be more or less easy depending on the parts relative geometry. In this case, it becomes interesting to carefully choose the two pieces to be paired among available parts. As one does not know exactly the geometrical characteristics of the items that will be produced in the future, the problem of wisely choosing, over the long haul, the pairs to be bonded is dynamic. Minimizing the cost of pairing operation can be formulated as a dynamic linear assignment problem. This paper presents different heuristics used to solve the dynamic linear assignment problem in the framework of a specific application in the aeronautics industry. The article highlights how strong characteristics of the case study are used to choose adapted heuristics.

1 INTRODUCTION

In a plant, the task of assembling two parts can be more or less easy depending on the parts relative geometry. In the Aeronautics Industry, the problem of slotting two parts is encountered during the production of Composite Fan Blades. The two parts to be paired are: the Composite Fan Blade and its Metallic Leading Edge (MLE). These two parts are illustrated on Figure 1.

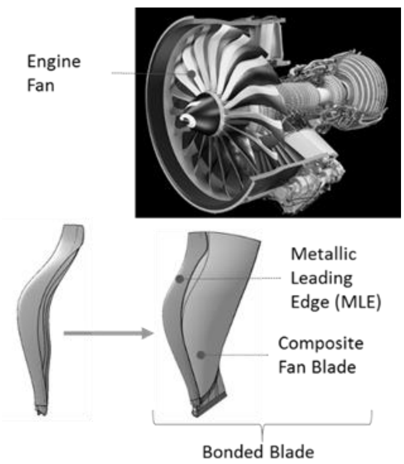

Figure 1: Engine Fan Blade and Metallic Leading Edge: the two parts to be paired.

If the two parts fit well, the pairing can be easy. On the other hand, if the two parts do not fit, it can be necessary to make some adjustment by benching the blade before pairing. Benching is a time-consuming step one wants to avoid. Thus, the following question is raised every day t: given the sets of blades and MLEs available in the stock which N_t pairs should be bonded so has to minimize, over the long haul, the cost of this production step? In our case, the static problem of choosing the best N_t pairs for an optimal cost at date t, is easy: the sets of available blades and MLEs is small and there is no need for a quick online computation (choice of pair is done only once a day). On the contrary, optimizing in the long term is hard: we do not know the characteristic of the items which will be produced in the future and choices made at date t have an impact on available choices at date $t+1$.

The static version of our problem, is in fact an extension of the Linear Sum Assignment Problem (LSAP): how to assign a number of tasks to a number of resources so as to minimize the total cost of the assignment where the global cost of the assignment corresponds to the sum of each individual assignment's cost. The small difference with the classical LSAP is that, here, the number of tasks and resources is higher than the number of pairs to be done (cf. part 3.1.1). Linear Assignment Problem and its extensions are linear optimization problems which

Bisot, C.
Dynamic Linear Assignment for Pairing Two Parts in Production - A Case Study in Aeronautics Industry.
DOI: 10.5220/0006627402540263
In *Proceedings of the 7th International Conference on Operations Research and Enterprise Systems (ICORES 2018)*, pages 254-263
ISBN: 978-989-758-285-1

have numerous applications in various fields: pairing weapons with targets (Ahuja et al., 2007), machine scheduling (Pinto and Grossmann, 1998), vehicle routing (Dantzig and Ramser, 1959) etc. It has been extensively studied and many algorithms have been proposed to solve it, among which the Hungarian algorithm (Kuhn, 1955) efficiently implemented in (Jonker and Volgenant, 1987). In our case, the dimension of the problem is small and computation time constraints are low, the static linear assignment is thus easily solved.

The problem of wisely choosing pairs in a long run is referred to as the Dynamic Assignment Problem: choosing the best assignment at date t without knowing which resources and tasks will enter the system in the future. The mathematical framework of a general class of dynamic assignment problems is established in (Spivey and Powell, 2004).

This paper explains how the framework of Dynamic Linear Assignment is applied to the specific case of pairing blades with MLE in a plant. We will highlight how some strong characteristics of the case study are taken advantage of to find satisfying heuristics for the optimization.

In a first part, the characteristics of the case study are described in details: cost, constraints, parts' flows in the plant and final cost function to be minimized. In a second part, different heuristics to solve the problem are proposed. In the last part the heuristics are tested on a set of real data coming from a plant so as to evaluate performances and compare strategies.

2 PROBLEM FORMULATION

In this part, the different costs and constraints of the case study are presented. Then the dynamic of the system is described. Finally, the total cost function function to be minimized is written. In a last parts, the specificities of our case study are highlighted.

2.1 Pairing Cost

The cost of pairing a MLE a with a blade b depends on two elements:

- How much material has to be benched to make the pair. The contribution of benching to the total cost is thus a function of a and b geometrical characteristic that we will note: G_{ab}. If the pair can be done without benching, $G_{\text{ab}} = 0$.
- The relative position of the bonded blade and MLE compared to nominal position. The relative position of blade and MLE is characterized by a few geometrical measures on the bonded blade noted X_{ab}. We are aiming at having pairs with relative position as close as possible to the nominal X_0. The distance to nominal is measured by a well-chosen norm (not detailed here): $\|X_{ab} - X_0\|$.

Thus, if a MLE a and a blade b are pairable, the cost of the pair is defined as:

$$C(a,b) = G_{\text{ab}} + \|X_{ab} - X_0\| \qquad (1)$$

For now, we suppose that given the geometrical characteristics of two parts a and b, we are able to predict both how much material will have to be removed and the relative position of the two parts on the bonded blade. Predicting cost is a challenge by itself which can be done using different technics. Here we can mention in particular S. Flöry's work on point clouds and surfaces matching (Flöry, 2010). In reality, cost prediction is imperfect and $C(a,b)$ is known with uncertainties: $C(a,b)$ is a random variable, we know only its expectation. This will limit the performance of any heuristics used to optimize pairs' choices.

2.2 Constraints on Production Flows

2.2.1 Production Rate

Most important constraint on production flow is the number of pairs which have to be done every day. Let N_t be the number of pairs to be done at date t. If the N_t pairs cannot be done at date t, the production is delayed. The cost of not being able to make a pair when we are asked to (there is not enough pairable parts available in the batch) is noted R. For example, on date t if only $N_t - 2$ pairs can be done, this will cost : $2 \times R$.

If MLE, a, and blade, b, are not pairable, as an artefact in the computation, we can say that the cost of the pair is:

$$C(a,b) = R \qquad (2)$$

Let K_{ab} be a Boolean giving the pairability of MLE a with blade b. For any pair nature (pairable or not pairable), the pair cost is:

$$C(a,b) = \begin{cases} R & if\ K_{ab} = 0 \\ G_{ab} + \|X_{ab} - X_0\| & if\ K_{ab} = 1 \end{cases} \qquad (3)$$

2.2.2 MLE Limited Life

Because of a surface treatment performed on MLEs to improve bonding quality, MLE cannot wait for too long in the batch at pairing post. If it stays more than

$\Delta t_{lim}^{LE} = 90\, days$, it will be scraped (i.e removed from the system).

The cost of scraping a MLE is noted S^{LE}.

2.2.3 Ordering of Blades Flow

For production engineers, it is better if blades production order is not shuffled too much. This is an important constraint because, among others things, it helps detecting production crisis.

This constraint was modeled as follows: if a blade stays more than $\Delta t_{lim}^{B} = 7\, days$ at the pairing post, we get a delay penalty of D^B. Unlike MLEs, when a blade stays more than Δt_{lim}^{B} at pairing post, it is not scraped and thus stays in the system.

Note that we can get a penalty only once in a blade life: for it doesn't cost more if a blades stays more than 7 days in the batch than if it spends exactly 7 days. It is also important to notice that in our application $\Delta t_{lim}^{B} \ll \Delta t_{lim}^{LE}$.

2.3 System Dynamic

The production and parts flows at the plant are modelled as follows:

- Each working day (5 days a week), n pairs have to be done.
- Every day, the pairs are chosen among the sets of MLEs and blades available at the pairing post. We call those parts "actionable parts". There is constant buffers of $5 \times n$ MLEs and $2 \times n$ blades actionable in the batch. A larger buffer of MLEs is needed since MLEs present more geometrical variability than blades.
- Every week, a batch of $5 \times n$ MLEs enters the plant. The 3D geometry of these MLEs is known immediately when it enters the plant. However, the MLEs are not instantly actionable because MLEs have to be inspected before entering the pairing post. These $5 \times n$ MLEs are progressively inspected during the week and become actionable little by little. A batch of known but not actionable MLE is always available. The number of MLEs in this batch varies from $10 \times n$, at the beginning of the week, to $5 \times n$ at the beginning of the week.

Following notation will be used later:

- $\mathcal{A}(t)$ is the set of actionable MLEs at date t.
- $\bar{\mathcal{A}}(t)$ is the set of known but not actionable MLEs at date t.
- $\mathcal{B}(t)$ is the set of actionable blades at date t.

Figure 2 gives an overview of the production flows described above. What is important to remember here is that MLEs are known before being actionable (from one to two weeks beforehand). This is a rich information to be used for long term optimization.

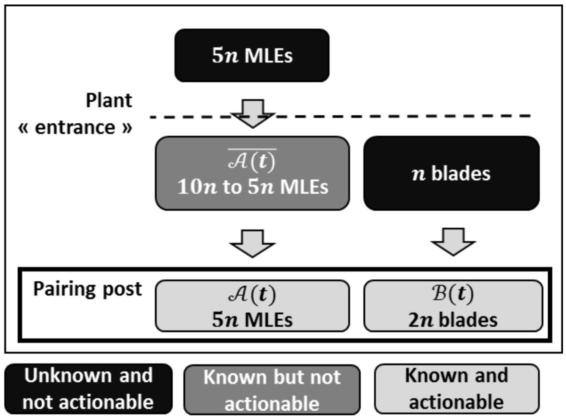

Figure 2: Blades and MLE flows at the plant. $n$is the number of pairs to be done at each working day t. $\mathcal{A}(t)$, $\bar{\mathcal{A}}(t)$and $\mathcal{B}(t)$are the different sets of available parts.

2.4 Total Cost Function

2.4.1 Total Cost of Pairing Operation

The total cost of pairing operation between dates $t = 0$ and $t = T$ (T is typically a value big in comparison with Δt_{lim}^{LE}, the limit time a MLE can stay in the batch before being scraped), $\mathcal{C}$, can now be written as follows:

$$\mathcal{C} = \sum_{t=0}^{T} \left(\sum_{n=1}^{N_t} C(a_n^t, b_n^t) \right) + n^s \cdot S^{LE} + n^D \cdot D^B \tag{4}$$

With:

- N_t, the number of assignments to be done at date t.
- (a_n^t, b_n^t), the n^{th} pair chosen at date t. MLE a_n^t and blade b_n^t are chosen among the actionable parts at date t.
- $C(a_n^t, b_n^t)$, the cost of the pair (a_n^t, b_n^t) as defined in equation (3).
- n^S, the number of MLE which had to be scraped (spent more than Δt_{lim}^{LE} days in the stock) between 0 and T.
- n^D, the number of blades which were delayed (spent more than Δt_{lim}^{B} days in the stock) between 0 and T.

$\mathcal{C}$ is the total cost to be minimized. Our problem is to find heuristics to choose the pairs (a_n^t, b_n^t) so as to minimize this total cost $\mathcal{C}$. The pairs (a_n^t, b_n^t) are chosen in the sets $\mathcal{A}(t)$ and $\mathcal{B}(t)$. Choice are made at date t given the information on the system available at date t, i.e. only knowing $\mathcal{A}(s), \mathcal{B}(s)$ and $\bar{\mathcal{A}}(s)$ for any date s before t.

2.4.2 Costs Hierarchy

The orders of magnitude of the different sources of cost in $\mathcal{C}$ are the following:

- R, the cost of a delay in production, and S^{LE}, the cost of scraping a MLE, are of the same order of magnitude.
- D^B, the cost of having a blade delayed, is about $R/2$.
- G_{ab}, the cost of benching varies between $R/100$ and $R/10$ (except for pairs feasible without benching for which $G_{ab} = 0$).
- $\|X_{ab} - X_0\|$ the cost of being away from nominal position varies between 0 and $R/1000$.

Costs are strongly hierarchical: it is much more important to avoid production delay or MLE scraping than to avoid benching which is also much more important than minimizing bonded blade distance to nominal. This hierarchical structure of costs will help a lot for choosing an adapted heuristic for long term optimization.

2.5 Case Study Important Properties

The constraints and costs of pairing for this specific problem has four strong characteristics which will help finding a satisfying heuristic to solve the problem:

- The $\mathrm{N_t}$ pairs to be done at date t are chosen once all together at the beginning of the day. There is no computation time constraints.
- Blades and MLEs play very asymmetrical roles: production flow properties and constraints are very different for the two parts.
- The total costs of the pairing step is made up of different components (cost of not being able to keep production speed, cost of benching, cost of scraps etc.). The cost structure is very hierarchical so that it is easy to know which events should absolutely be avoided without taking any risk and which events are acceptable.
- The MLEs entering the system are known in advance (before the MLEs become available for pairing). This helps taking decision at date t which will not badly impact the choices available at date $\mathrm{t+n}$.

3 HEURISTICS

In this part, we first present some of the basic blocks composing the different strategies proposed to solve the problem. Then, we present in details five different heuristics: one simple myopic strategy serving as a reference, three other more sophisticated myopic strategies and one non-myopic strategy. We call myopic strategies those in which decisions are made without using information given by the set of known but not actionable MLEs, $\bar{\mathcal{A}}(t)$.

3.1 Basic Blocks

In this part, we first present three basic bricks which are part of the heuristics presented later. Then, in part 3.1.4, the general structure of the heuristics described later is presented.

3.1.1 A Static Linear Assignment

The static linear assignment in our case, can be formulated as a generalization of the classical Linear Sum Assignment Problem. Given a set of N_A resources (MLE), a set of N_b tasks (blades) and a number of pairs to be done N, the goal is to find the set of N pairs which minimize total cost of the assignment.

This problem can be written in the form of a linear optimization problem:

$$\underset{X}{\operatorname{argmin}} \sum_{a=1}^{N_A} \sum_{b=1}^{N_B} x_{a,b} \cdot c_{a,b} \tag{5.1}$$

With

- $c_{a,b}$ the cost of pairing resource a with task b.
- $x_{a,b}$ the decision variables with $x_{a,b} = 1$ if resource (MLE) a is assigned with task (blade) b, 0 otherwise.

Under the constraints:

- The decision variables $x_{a,b}$ are Booleans:

$$\forall\,(a,b), \qquad x_{a,b} \in \{0,1\} \tag{5.2}$$

- Each resource is assigned at most once:

$$\forall\, a,\ \sum_{b} x_{a,b} \leq 1 \tag{5.3}$$

- Each task is assigned at most once:

$$\forall\, b,\ \sum_{a} x_{a,b} \leq 1 \tag{5.4}$$

- N assignments have to be done:

$$\sum_{a}\sum_{b} x_{a,b} = N \tag{5.5}$$

This linear optimization problem can be solved using any standard linear programing algorithms.

3.1.2 Correction of Cost Matrix

One other important block of our heuristics is the computation of a corrected cost matrix $\tilde{C}(t)$. The basic idea is to artificially reduce the cost of pairs containing old MLE or old blades. A static linear assignment (cf. part 3.1.1) will then be performed on the corrected cost matrix and oldest blades or MLEs will be favored. The goal is to anticipate MLEs' scraps and blades' delays.

We can correct cost matrix in a simple way that we will call a myopic correction.

At each time step t:

- The cost of any pair realized with a "too old" (i.e close from being scraped) MLE a is artificially reduced to favor this pair.
  Let age_a^t be the age of MLE a at date t and L^{LE} be an age limit close to Δt_{lim}^{LE}. Cost is corrected as follows:

$$\forall\, a \mid age_a^t > L^{LE},\ \forall\, b \mid K_{ab} = 1, \quad \widetilde{C_{ab}}(t) = C_{ab} - S^{LE} \tag{6}$$

This accounts for the risk that, if MLE a is older than L^B and not paired at time t, this will cost S^{LE} because the MLE will be scraped in the following days.

- Similarly, for each blade b which is too old, i.e close from Δt_{lim}^{B} limit (older than an age limit L^B), the cost of feasible pairs is reduced (so as to favor these pairs):

$$\forall\, b \mid age_b^t > L^B,\ \forall\, a \mid K_{ab} = 1, \quad \widetilde{C_{ab}^{t}} = C_{ab} - D^B \tag{7}$$

  This accounts for the risk that, if this blade b older than L^B and not paired at time t, this will cost D^B because the blade will be considered as delayed in the following days.

The values of L^{LE} and L^B are tuned based on the problem characteristics: Δt_{lim}^{LE}, Δt_{lim}^{B}, global proportion of not pairable pairs, sizes of blades and MLEs buffers etc. These parameters can be optimized by simulations similar to those presented in part 4.

For the non-myopic strategy, the cost matrix is corrected in a more sophisticated way which will be described in part 3.2.5 but basic idea stays the same: favor oldest blades and MLEs by reducing the cost of their pairs.

3.1.3 Maximum Cardinality Bipartite Matching

In our strategies, it is often useful to answer following question: given a set of MLEs and a set of blades, what is the maximal number of feasible pairs?

This can be done using a maximum bipartite matching algorithm like Ford-Fulkerson algorithm for example (Ford and Fulkerson, 1962). The sets of MLEs and blades are represented as a binary bipartite graph linking the set of MLEs with the set of blades. If the pair a, b is feasible, edge a, b exist. If the pair is not feasible, the edge does not exist.

3.1.4 Structure of the Strategies

The different strategies described in the following sections all have in common a four steps structure. At each time step t:

- First a cost matrix is calculated using all actionable blades and MLEs.
- Then we select a subset of blades and MLEs among all actionable ones so as to favor oldest blades and oldest MLEs (avoid MLE's scrap and blade's delay). We obtain a paring cost submatrix. In this part maximum cardinality bipartite matching algorithm plays an important role.
- The paring cost submatrix is then corrected so as to favor again old blades and old MLEs. For myopic strategies, cost correction is done as described in part 3.1.2. For the non-myopic strategy, cost correction is performed in a more subtle way described in part 3.2.5
- Finally a static linear assignment optimization (3.1.1) is performed on the corrected cost submatrix $\tilde{C}(t)$ to choose the N_t assignments which minimize total cost of the assignment.

3.2 Detailed Heuristics

In this part, the different heuristics imagined to help choosing which N_t pairs should be done at each time t are described in details. Each strategy is later evaluated on real data coming from the plant. Myopic strategies are strategies where no more future prediction than cost correction described in 3.1.2 is used.

3.2.1 Myopic Linear Assignment (MLA)

This strategy is the simplest strategy one can think of and will serve as a reference to evaluate efficiency of the others. Pairs are suggested in batch at each date t.

Among all actionable blades and MLEs at time t, we chose the N_t pairs so as to minimize the total cost of pairing (optimization is done solving a static linear assignment as described in 3.1.1). The cost matrix used for the assignment, is the corrected cost matrix as described in 3.1.2 (i.e. anticipating the cost of scraped MLE and delayed blades).

3.2.2 MLA with FIFO on Blades

At each time step, we select a subset of all actionable blades. The subset is the smallest and oldest set of blades so that it is possible to make N_t pairs between those blades and all actionable MLEs. This set is chosen so as to take oldest blades first and it is noted $\mathcal{B}'(t)$. The selection of $\mathcal{B}'(t)$ is done solving a succession of maximum cardinality bipartite matching (cf. part 3.1.3). In the set $\mathcal{B}'(t)$ and the set of all actionable MLEs $\mathcal{A}(t)$, we choose the N_t pairs minimizing the total cost of pairing (using corrected cost matrix as described in 3.1.2).

This algorithm combine advantages of assigning pairs by batch and keeping FIFO (First In First Out) lines on blades so as to avoid blades delays. However, this strategy doesn't favor oldest MLEs so that we do not avoid MLEs scrap (except through basic cost correction).

3.2.3 Myopic Linear Assignment with FIFO on Blades and MLE

A each time step, we first select $\mathcal{B}'(t)$, smallest and oldest set of blades so that it is possible to make N_t pairs with the set of all actionable MLE $\mathcal{A}(t)$. Then, we select $\mathcal{A}'(t)$ the smallest set of MLEs so that it is possible to make N_t pairs with the set $\mathcal{B}'(t)$. $\mathcal{A}'(t)$ is chosen so as to take oldest MLEs first. Then, amongst $\mathcal{B}'(t)$ and $\mathcal{A}'(t)$, we choose the N_t pairs so as to minimize the total cost of pairing.

This algorithm combines the advantages of assigning pairs by batch, assigning oldest blades and oldest MLEs first. An advantage is given to oldest blades over oldest MLEs since the time before delay of a blade is a lot shorter than time before the scrap of a MLE. The drawback of this strategy, is that the static linear assignment is performed on narrowed sets of MLEs and blades (see Figure 3) with reduced choices for the pairs.

3.2.4 MLA with FIFO on Blades and Partial FIFO on MLEs

We select the set $\mathcal{B}'(t)$. With this set $\mathcal{B}'(t)$ and the set of all actionable MLE $\mathcal{A}(t)$, it is possible to make a maximum of $m(t)$ pairs without benching the blades. We want to perform pairing on oldest MLEs without degrading the number of pairs which can be done without benching.

We select $\mathcal{A}''(t)$ the smallest set of MLEs so that it is possible to make N_t pairs and $m(t)$ pairs without benching with the sets $\mathcal{B}(t)$. $\mathcal{A}''(t)$ is chosen so as to take oldest MLEs first. Then, among $\mathcal{B}(t)$ and $\mathcal{A}''(t)$, we choose the N_t pairs so as to minimize the total cost of pairing.

This algorithm is a compromise between algorithms 3.2.2 and 3.2.3. It combines the advantages of assigning pairs by batch, assigning oldest blades first and giving advantage to oldest MLEs. With this strategy, we perform an optimal linear assignment on a larger sets of MLEs than with strategy 3.2.3 so that this gives more chance for optimization. However, more risk of MLE scrap is taken. This strategy takes advantage of the cost hierarchy to choose the set $\mathcal{A}''(t)$: the number of pairs done without benching is the same as the one for strategy 3.2.2, this implies that cost of the assignment is not degraded too much by the reduction of MLE set.

3.2.5 Non-myopic Strategy

In this non-myopic strategies, the goal is to correct cost matrix in a more subtle way than what was described in part 3.1.2. The goal is to favor pairing of MLEs and blades which are hard to pair over those which are easy. Most important contributors to final total price are MLE's scraps and blade's delays. Thus, we focused on anticipating those costs and avoiding it. This is why we try to pair blades and MLEs which are hardly pairable first (they have a higher risk to be delayed or scraped).

In this strategy, no sub-matrix is selected, a static linear assignment is performed on all actionable MLEs and blades using a cost corrected matrix. At each date t, the goal is to subtract from the initial cost of a pair, $C_{a,b}$, an estimation of how much it could cost if MLE a and blade b were not paired at t and were thus kept in the system. The expectation of the cost of keeping MLE a (blade b) in the system is estimated through the risk that the MLE a (blade b) will be scraped (delayed). Cost correction is done for blade and MLE separately as described below.

Cost Correction for MLE Scrap Anticipation

For each actionable MLE a of age age_a^t at time t, we denote:

- N_a^t the number of blades which will become actionable before this MLE gets scraped. In other

words, this is the number of blades entering the system in the next $\Delta t_{lim}^{LE} - age_a^t$ days.
- p_a^t the probability that this MLE is pairable with a blade. p_a^t is estimated based on MLE's pairability with blades which previously entered the system. It is re-estimated at each time step as new blades enter the system.

The probability that none of the N_a^t incoming blades will be pairable with MLE i is:

$$P_a^t = (1 - p_a^t)^{N_a^t} \tag{8}$$

The cost of keeping MLE a in the system, is mainly driven by the increase of the risk for the MLE to be scraped (because scrap is the most expensive source of cost). Then, the cost of each pairable sets of MLE and blade is corrected as follows:

$$\forall\, a, b \mid K_{ab} = 1, \quad \widetilde{C_{ab}^{0}}(t) = C_{ab} - S^{LE} \times P_a^t \tag{9}$$

Cost Correction for Blade Delay Anticipation

For each blade b, of age age_b^t, actionable at time t, we know which MLEs will enter the system before it gets too old. In other words, we know which MLEs will become actionable in the next $\Delta t_{lim}^{B} - age_b^t$ days. This is thanks to the important batch of MLEs known but not actionable described in part 2.3. The set of MLEs which will become actionable in the next $\Delta t_{lim}^{B} - age_b^t$ is called $\overline{\mathcal{A}_b}(t)$. It is a subset of $\bar{\mathcal{A}}(t)$.

The cost of keeping blade b in the system, is mainly driven by the increase of the risk for the blade to be delayed (because delay is the most expensive source of cost generated by the blade).

If there is no MLE in set $\overline{\mathcal{A}_b}(t)$ with which blade b is pairable we perform a cost correction. Otherwise no cost correction is done.

$$\begin{gathered} \forall\, a \mid (\nexists\, m \in \overline{\mathcal{A}_b}(t) \mid K_{ma} = 1) \\ \forall\, b \in \mathcal{A}(t) \mid K_{ab} = 1, \\ \widetilde{C_{ab}}(t) = \widetilde{C_{ab}^{0}}(t) - D^B \end{gathered} \tag{10}$$

Remarks on Flows Anticipation

Here it is important to highlight that practically, flows are not perfectly known:
- In reality, N_a^t will have to be estimated.
- In reality, we do not know precisely the order and when MLEs will become actionable: $\overline{\mathcal{A}_b}(t)$ is not perfectly known.

3.3 Summary of the Different Heuristics

The different myopic strategies presented above are all very similar: assignment is performed by batch using static linear assignment. The difference between those strategies is only the sets of MLEs and blades on which the static linear assignment problem (3.1.1) is solved. Figure 3 presents a schematic view of the sets on which linear assignment algorithm is applied for the different strategies. The best strategy will depend on the proportion of non-pairable pairs, the proportion of pairable pairs without benching and the balance between the different costs. Moreover, Table 1 gives an overview of the different strategies pro and cons.

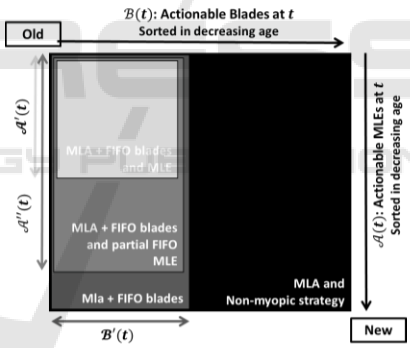

Figure 3: Graphical representation of the different strategies. The rectangles represent the sets of blades and MLEs on which static linear assignment is performed.

Table 1: Summary of the different heuristics properties. This table highlights the pros and cons of the heuristics proposed.

	MLA, 3.2.1	MLA + FIFO on blades, 3.2.2	MLA + FIFO on blades and MLE, 3.2.3	MLA + FIFO blades and partial FIFO MLA, 3.2.4	Non-Myopic, 3.2.5
FIFO blades	No	Yes	Yes	Yes	No
FIFO MLE	No	No	Yes	Partial	No
Favor pairing without benching	Yes	Yes	No	Yes	Yes
Improved prediction of MLE scrap	No	No	No	No	Yes
Improved prediction of Blades delay	No	No	No	No	Yes

4 HEURISTICS COMPARISONS

In this part, heuristics efficiencies are compared in two different cases:

- The case where the algorithm to estimate the cost of a pair is supposed to be perfect:
  - We know if a pair is feasible or not.
  - If the pair is feasible, we know if it can be done without benching.
  - The cost of the pair is exactly known.
- The case where algorithm to estimate the cost of a pair has uncertainties:
  - We know if the pair is feasible or not.
  - If the pair is feasible, we only know the probability that the pair is feasible with benching, p_1, or feasible without benching, $p_2 = 1 - p_1$.
    Let C_1 be the expected cost of the pair knowing that it is feasible with benching and C_2 be the cost of the pair knowing that it feasible without benching. The expected pair's cost is:

$$C = p_1 \times C_1 + p_2 \times C_2 \tag{11}$$

  - In this part simulated uncertainties are representative of uncertainties encountered in practice.

4.1 Simulation Description

Simulations are based on data from a three week production at the plant: we know the geometrical characteristics and, thus, the expected cost matrix and the real cost matrix for a large set of blades and MLE.

Strategies efficiencies are estimated by simulating 400 weeks of production for each strategy (enough for the operation average cost to converge). The production is simulated as follows:

- Ingoing and outgoing flows are simulated according to the flows described in part 2.3.
- Ingoing MLEs (blades) are simulated with a random sampling (with replacement) among the input MLEs set (blades set) with known geometrical characteristics issued from production. This means that we make the hypothesis that the shape of MLEs (blades) entering the system at date t is completely independent from the shape of MLEs (blades) entering the system at date $t + 1$. This hypothesis was roughly verified on the 3 weeks production dataset which was studied.
- Outgoing flows are the result of tested strategy.

4.2 Results with Perfect Cost Predictions

In this part, strategies efficiency are compared with the hypothesis that real cost (and real pair nature: non-feasible, feasible with benching or feasible without benching) is known.

The results are summarized in the Table 2. For each strategy, we have:

- The proportion of blades delayed: the ratio of the number of blades which stayed in the system more than Δ_{lim}^{Blades} over the number of pairs which were asked to be done ($\sum_t N_t$)).
- The proportion of inactivity: the ratio of the number of pairs which couldn't be done over the number of pairs which were asked to be done ($\sum_t N_t$).
- The proportion of scraped MLEs: the ratio of the number scraped MLEs over the number of pairs which were asked to be done.
- The proportion of benched pairs: the ratio of the number of pairs which were benched over the number of pairs which were asked to be done.
- The average cost of a week of production.

We see that every strategy enables to reach 0% of scraped MLEs and 0% of "inactivity". This is, amongst other things, related to the fact that proportion of non-feasible pairs are rather rare in our dataset (5% of the pairs). We also see that the best strategy is the non-myopic one which enables to avoid all sources of high costs: inactivity, scraping, delay and benching. However, given the fact that proportion of non-feasible pairs in our data set is low, the performances of the non-myopic strategy are not much higher than those of the myopic strategies.

4.3 Results with Uncertainties on Cost Predictions

In this part the difference is that decision are made based on expected cost matrix instead of real cost matrix.

Results are summarized in Table 3. We see that average cost of a week of production is much higher than when cost are known without uncertainty. With a perfect cost estimation, we can expect to bench 0% of the pairs whereas with uncertainties on cost estimation representative of real cost uncertainties, best strategy leads to 25% of benched pairs. This shows how important quality of cost prediction algorithm is. We also see that with cost uncertainties, the differences between strategies efficiencies are much smaller.

Table 2: Strategies comparison with exact cost predictions. This table summarises the results of the simulations used to compare the efficiency of the different heuristics. The simulation are done, in the theoretical case where we suppose that we are able to perfectly predict the cost of the pair before doing it.

		MLA, 3.2.1	MLA + FIFO on blades, 3.2.2	MLA + FIFO on blades and MLE, 3.2.3	MLA + FIFO blades and partial FIFO MLA, 3.2.4	Non-Myopic, 3.2.5
Algorithm parameters	Correction horizon MLE, L^{LE}	7 days	7 days	7 days	7 days	NA
	Correction horizon blades, L^{B}	1 day	1 day	1 day	1 day	NA
Results	Proportion of blades delayed	0,00475	0	0	0	0
	Proportion of inactivity	0	0	0	0	0
	Proportion of scraped MLEs	0	0	0	0	0
	Proportion of pairs benched	0	0,00181	0,0463	0	0,00012
	Average cost of a week of production	10,5	6,15	24,4	5,88	5,47

Table 3: Strategies comparison with uncertainties on cost predictions. This table shows the results of the simulations used to compare heuristics efficiency. The simulation are done in the case where algorithm to predict pairs' cost is not perfect: real cost of a pair can be different from what was predicted before pairing.

	MLA, 3.2.1	MLA + FIFO on blades, 3.2.2	MLA + FIFO on blades and MLE, 3.2.3	MLA + FIFO blades and partial FIFO MLA, 3.2.4	Non-Myopic, 3.2.5
Proportion of blades delayed	0,00881	0	0	0	0
Proportion of inactivity	0	0	0	0	0
Proportion of scraped MLEs	0	0	0	0	0
Proportion of pairs benched	0,26	0,256	0,256	0,254	0,262
Average cost of a week of production	120	110	110	109	111

5 CONCLUSIONS

This article shows how the framework of dynamic linear assignment was applied to the specific problem of pairing blades with MLEs in a plant. The strong characteristics of the studied system were taken advantage of so as to design a few adapted pairing strategies. Among the strategies, one was a simple myopic strategy serving as a reference, three were adapted myopic strategies and one was a non-myopic heuristic.

The different strategies were tested on a set of real data in the case where exact pairs costs are known before pairing and in the case where there is uncertainties on cost prediction. We highlighted the fact that strategies efficiency is strongly related to the quality of cost estimation. We also showed that the fours strategies proposed (three myopic, one non-myopic) enable to significantly reduce the cost of pairing operation. If costs are perfectly known, the non-myopic heuristic is the best one. However, this strategy is harder to implement in reality since more inputs (about the incoming flows of blades and MLEs) are needed.

Future work on the subject will include, influence studies to see how system reacts to changes on some key inputs of the model: buffer size for MLE and blade stocks, proportion of non-feasible pairs in the simulation, proportion of pairs feasible without benching in the simulation etc.

Some work should also be done to analyze the effect on the system to have time dependency between geometrical attributes of blades (MLEs) entering the system a t and those entering at $t+1$. The fact that time series of blades (MLEs) attributes are not completely random makes a lot of sense since two blades (MLEs) entering the system roughly at the same date will tend to come from the same batch of production and thus to share more similarities than two blades (MLEs) coming from different batches.

REFERENCES

Ahuja, R., Kumar, A., Jha, K. C. and Orlin, J. B., 2007. Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem. *Operations Research,* 55(6), p. 1136–1146.

Dantzig, G. and Ramser, J., 1959. The Truck Dispatching Problem. *Management Science,* 6(1), pp. 80-91.

Flöry, S., 2010. *Constrained Matching of Point Clouds and Surfaces,* TU Wien: PhD Thesis.
Ford, L. R. and Fulkerson, D. R., 1962. Flows in Networks. *Princeton University Press.*
Jonker, R. and Volgenant, A., 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing,* Volume 38, p. 325.
Kuhn, H., 1955. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quaterly 2,* pp. 83-97.
Pinto, J. M. and Grossmann, I. E., 1998. Assignment and sequencing models for the scheduling of process systems. *Annals of Operations Reasearch,* Volume 81, pp. 433-466.
Spivey, M. Z. and Powell, W. B., 2004. The Dynamic Assignment Problem. *Transportation Science,* 38(4), pp. 399-419.