A Decision Support System for Scheduling a Painting Facility in an Automotive Supplier

Jose Pedro García-Sabater, Carlos Andrés Romano,
Julio Juan García-Sabater, Cristóbal Miralles
Universidad Politécnica de Valencia, Camino de Vera, s/n, 46023 Valencia, Spain,
{jpgarcia, candres, jugarsa, cmiralles}@omp.upv.es

This paper deals with the problem of scheduling with setups of two types due to a closed loop configuration with a given quantity of positions. This is typical of closed painting facilities where products are scheduled in different loops. Products are defined by its geometry and its colour. When a change of colour is to be scheduled a setup -horizontal setup- is to be paid (either in terms of cost or in terms of lost capacity). But when in successive loops but in the same position a different geometry is going to be scheduled, a setup cost is also to be paid –vertical setup-. The paper models three variants of the problem. Moreover a greedy heuristic is proposed and it is evaluated through the resolution of a set of problems that covers many aspects of the reality considered.

Keywords: Applications, Production Scheduling, Real World Scheduling.

1. Introduction

Many production systems with closed loop facilities have to deal with the problem of scheduling batches in consecutive loops (e.g. electrolytic painting of automotive parts in closed conveyors, cyclic painting of metallic furniture…). Batch scheduling consists of grouping and scheduling jobs in batches. Such grouping is based on the existence of some similarity between jobs belonging to the same class or family (such as their colour, size or geometry) to minimize changeover or setup time incurred when starting a new batch (where setup time can be sequence dependent or not, see Allahverdi et al. (1999)). The objective of batch scheduling is to minimize the total changeover time or cost satisfying the customer's demand, Mosheiov et al. (2005). In most cases, a job is completed and released only when the batch to which it belongs has finished its processing. But in Just in Time (JIT) environments (such as the automotive sector or supply chain integrated manufacturers), the demand between supplier and customer imposes a cyclic demand, so that batches must be produced within a specific cycle time to minimize the parts and components inventory. Thus, a trade-off between minimizing the total setup time (achieved by reducing the number of batches) and satisfying the customer's demand under JIT perspective (achieved by reducing batch size) must be considered to solve efficiently batch scheduling problem.

Over the past two decades, batch scheduling has become a central topic for researchers, e.g. see Potts and Van Wassenhove, (1992) or Potts and Kovaliov (2000) for a survey. In the one machine problem, Albers and Brucker (1993) showed the complexity of the problem even for the simplest problems and Dobson et al. (1987), Naddef and Santos, (1988), Santos and Magazine (1985) and Webster and Baker, (1995) have defined some procedures for solve this problem in presence of setup times.

A basic assumption in the batching and scheduling problem is that the setup time appears between consecutive batches, but when we have a loop before the machines instead of a linear queue, setup between non consecutive batches can appear, and some concerns related to cyclic scheduling must be considered.

Cyclic scheduling appears in this problem because some times is preferable to repeat a sequence uniform process of supplying to the line by the manpower rather than changing batches in consecutive loop cycles. Cyclic scheduling problems have been studied by several authors. A more complete description of this field can be found in the surveys of Hanen and Munier (1997), Crama (1997) and Gentina (2001). However, these references are closed to their initial scope, and don't consider many additional constraints and situation that might be found in reality.

This paper shows a real productive system which combines cyclic and batching scheduling. The paint line consists of a moving train that forms a continuous loop and contains a fixed number of hollow spaces (positions) that are used to fix the so-called jigs where one or several products are to be hung. The problem of scheduling with setups of two types due to the referred physical configuration arises in real systems. This is typical of closed painting facilities where products are scheduled in different loops. Products are defined by its geometry and its colour. When a change of colour is to be scheduled a setup -horizontal setup- is to be paid (either in terms of cost or in terms of lost capacity). But when in successive loops but in the same position a different geometry is going to be scheduled, a setup cost is also to be paid –vertical setup-. The paper models three variants of the problem. Moreover a greedy heuristic is proposed and it is evaluated through the resolution of a set of problems that covers many aspects of the reality considered.

The remainder of this paper is organized as follows. In Section 2, a detailed description of the real production system will be given. Sections 3 is devoted to give a formal description and a mathematical model for the problem. Some solution procedures are proposed at section 4, and finally the results to implement the procedures at the real productive system are discussed at section 5.

2. Problem description

The problem of cyclic batch scheduling arises in manufacturing facilities with closed conveyors. At the following an example of a closed conveyor facility will be described. Then a broad classification of such systems will be presented. This classification let us to study the different problems that arise in such systems.

An example of closed conveyor facility can be found in a manufacturer of plastic components of varying shapes and colours. Each product is defined by its geometry and its colour. The manufacturer has a closed loop paint line that consists of a moving train that forms that contains a fixed number of hollow spaces so called positions. Products are fixed on each hollow using a special tool so called jig. Each jig might hold a specific and limited number of parts of a given geometry. For the sake of clarity, and without loosing generality we will consider that only one product is hold on each different jig, in real problem this can be achieved by dividing the number of products by the required demand. It is important to note that each model uses a different type of jig, but that the same jig can be used for multiple colours. This means that when the product model to be painted is changed, the jigs must also be changed, but when there is only an alteration in the colour then it is not necessary to change the jigs. A limitation comes with the number of available jigs. Each one has a relevant cost and storing too many of them is not a good option, so usually they are limited. If the colour to be painted in successive units is different, the application of solvent through the pipes that are used to paint might be required. Moreover in some cases, it requires more time than that available between two consecutive positions. In such cases empty positions have to be allowed between those two batches.

In some cases the PLC controlling the painting pipes need to be reset if different geometries have to be painted. Such changeover might require a quantity of time superior to that available between two consecutive positions. Therefore when a change of colour is to be scheduled a setup -horizontal setup- is to be paid (either in terms of cost or in terms of lost capacity). When a change of geometry is to be scheduled a setup cost might be paid in terms of lost capacity. But, an here is the novelty, when in successive loops but in the same position a different geometry is going to be scheduled, a setup cost is also to be paid (vertical setup).

The parts pass continuously pass through a painting area located at a fixed position on the line. In the case studied, batching scheduling is desirable to minimize setup time between consecutive batches of similar parts and giving good response to the customer, and cyclic scheduling is desirable to reduce work in progress inventory between the facility and the automotive customers.

Summarizing each type of jig supports a defined number of parts (in our case 1 is general enough). In the system, different quantities of jigs exist for each model and it is not possible to exceed the maximum number of jigs per model daily.

Two types of setups could be considered: Horizontal Setups and Vertical Setups

1) **Horizontal Setups:** are conventional changeovers between batches of consecutive batches. They are related to two types of changes geometry changes and colour changes. Geometry changes are required for instance when the software controlling the pipes need to be updated. Colour changes might require solvent to clean pipes or space between two colours.
2) **Vertical Setups:** as previously mentioned, when a geometry change is required in the same position but in the following loop, a jig change must be carried out. It is necessary to use worker capacity to do this change. The use of a worker has associated a cost and this is the reason why every geometry change has associated a setup cost.

Different situations might arise depending on the empty space that is required to be schedule between consecutive batches. The situation 1 happens when the conveyor moves slowly enough it is possible to do the changes without losing capacity. This means that it is possible to replace the colours or the jigs in a position when it remains at the same place. However, some times the cycle time of the conveyor is faster and it is not possible to perform a change in a specific position into the cycle time. Situation 2 and 3 arise when empty spaces have to be allowed between positions. Situation 2 appears when only one position is to be freed either for geometry change and for colour change. Situation 3 requires different quantity of empty positions depending if the changeover is due to a geometry changeover or a colour changeover. Loosing capacity due to the changeovers is specially relevant in this situations, because due to the continuous movement of the loop and the limited capacity of the workers to carry out such changes, the number of jig changes during each loop still remains limited.

In Figure 1, it can be seen that the change between consecutive blocks of different models makes it necessary to include empty jigs in the sequence. This setup is necessary if consecutive blocks have different model types. When carrying out a model change, one hollow in the first block must be empty and two hollows in the next block must also be empty (see Figure 1). If a product is going to be scheduled on a block which in the previous loop contained a product with a different model type, then the jigs must also be changed but usually without loose capacity.

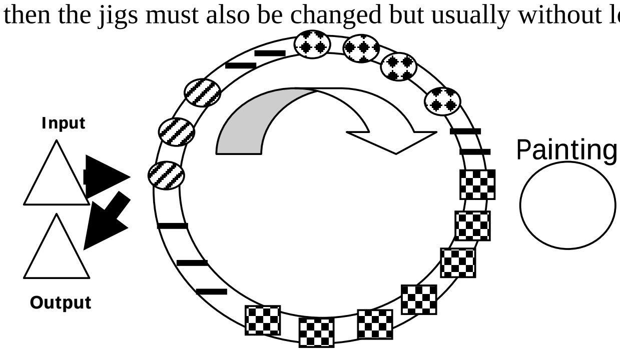

Figure 1. System layout

3. Problem statement.

3.1. Case 1. Problem description.

A better description can be stated using figure 2. The figure shows three cycles or loops with a number of different geometries (G1, G2, G3) colours (C1, C2,C3) in each loop. There are setup costs due to jig changes in consecutive loops and setup cost due to color in consecutive positions.

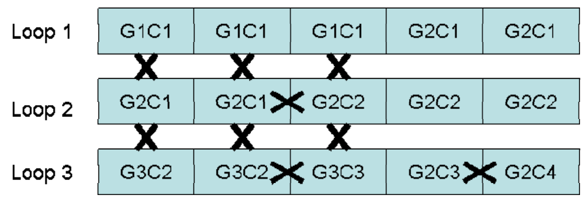

Figure 2. Schedules for Situation 1 at successive loops

The problem can be stated as follows: Sequence a set of products (defined by its geometry and its colour) in different quantities each one. Knowing that on each position only one unit can be scheduled. The number of units of a given model (in any set of BLOCS consecutive positions is limited by the availability of jigs to hang them. We have to minimize the number of colour changes (considering consecutive units) because each one has a cost (mainly in solvents), try to minimize the number of jig changes, between the same place of consecutive loops (mainly worforce cost), and try to minimize the number of empty positions (mainly energy cost). In this problem the geometry changeover in consecutive units is not relevant.

3.2. Case 2. Problem description.

The main difference between this case and the previous one is that when a geometry or colour change is scheduled an empty position should be scheduled. Note that we are considering consecutive positions and no consecutive loops. The objective will be then to sequence a set of products (defined by its geometry and its colour) in different quantities each one. At each position only one unit can be scheduled. The number of units of a given geometry (in any set of BLOCS consecutive positions is limited by the availability of jigs to hang themI. When a change in colour or model is to be scheduled an empty position has to be placed between those sets of units. We have to minimize the number of colour changes (between consecutive units) because each one has a cost (mainly in solvents), and to minimize the number of jig changes, the same position of consecutive loops (mainly worforce cost), and minimize the number of empty positions (mainly energy cost).

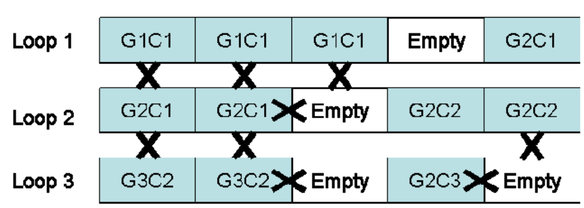

Figure 3. Schedules at successive cycles

A better description can be stated using figure 3. The figure shows three cycles or loops with a number of different geometries (G1, G2, G3) colours (C1, C2,C3) in each loop. There are capacity looses (one position) between consecutive colour changes, and setup costs due to jig changes.

3.3. Case 3. Problem description.

This case is a variant that arises when the geometry change needs a number of consecutive empty positions and that number is different considering colour changes.

Figure 4 tries to depict the problem. The figure shows three cycles or loops with a number of different colours and geometries in each loop. There are capacity loose (one position) between consecutive colour changes and geometry changes. Also jig change costs are considered.

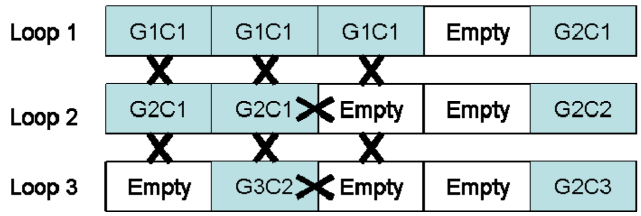

Figure 4. Schedules at successive cycles

3.4. General Problem definition

In every case, the problem may be stated as follows: given a set of products to manufacture (defined by their geometry, colour and required production quantity), the number of positions per loop (the system capacity), and the jig availability the objective is to schedule the products in the minimum number of blocks reducing the total cost given by setup cost (either horizontal and vertical) and the empty positions.

The objective function in this problem is to minimize the total number of empty positions in the blocks or maximize the capacity of the facility.

4. Solution procedures

The 1, 2 and 3 cases reflect variants of the same problem: to schedule the production taking into account not only the previously scheduled product but also the geometry of the product scheduled on the previous loop. Some herefore heuristic algorithms have been developed. When working with the real versions of the problem some questions arised that contributed to define the algorithms. The most important is that to maintain the previous loop geometry sequence is a very relevant point because is the only aspect that was considered by the workforce. Moreover "if every product is to be manufactured, if the geometry can be maintained, there is no need to change it" (as a foreman said). Of course the view from the board of directors point of view is slightly different because more gaps imply less productivity.

The following algorithm has been develop to solve the previous problems. It has to be noted that the following algorithms work for the third case (and therefore for the first and second).

The notation used for the models is kept but some new is added:

i is an index for products

[j]: represents the product sequenced at Position j

K[j] represents the colour of the product at Position j

H[j] represents the geometry of the product at Position j

R[j] represents the remaining quantity of units to be scheduled of product sequenced at position j

List is a list including the Code of the products to be scheduled. K is the index that will cover the ordered List. List(k) will be the product in place k. H(List(k)) is the Geometry of that product. K(List(k)) is the colour of that product. R(List(k)) is the remaining quantity of units of product List(k) to be scheduled. A product is to be removed if R(List(k))=0. Initially R(List(k))=Q(List(k))

JA(h) is the availability of Jigs for Geometry h

JU(h) is the Usage of Jigs for Geometry h during the previous BPL-1 positions

LastColour is the colour of the last scheduled product.

The implemented procedure is the following one

Step 1: List is ordered according to the chosen criteria.

Step 2: Initialize

Step 2.1 j:=1; k:=1

Step 2.2 JU(h)=0 for every h

Step 4: if **List** is empty then Finish

Step 5: ik k=0 then **Insert a Null Product**

Step 5.1 [j]:=0;
Step 5.2 j:=j+1
Step 5.3 JU(H([j-BLOCS]):= JU(H([j-BLOCS])-1
Step 5.4 Go to 8

Step 6: Insert the k-th product in the List

Step 6.1: [j]:=List[k]; LastColour:=K[j]
Step 6.2: R[j]=R[j]-1;
Step 6.3: JU(H[j]):= JU(H[j])+1;
Step 6.4: JU(H[j-BPL]):= JU(H[j-BPL])-1;
Step 6.5: j=j+1
Step 6.6: Go to 7

Step 7: Check if [j-1] can be reinserted

Step 7.1: if R[j]=0 then Remove [j] from the **List ;** Go to 8
Step 7.2: if JU(H(List(k)))=JA(H(List(k))) then Go to 8
Step 7.3: Go to 6

Step 8: Check if a Product with the same geometry H[j-BLOCS] can be inserted

Step 8.1: Count the number of empty positions before the position [j]
Step 8.2: if the number of empty positions is less than EH a change in model might be scheduled otherwise go to Step 8.4
Step 8.3: Look for the first Product in **List** with the same Geometry of H[j-BLOCS] and LastColour as colour. If the product exists set k to the place where it can be found and go to 5.
Step 8.4: if the number of empty positions is less than EK a change in colour might be scheduled otherwise go to Step 9
Step 8.5: Look for the first Product in **List** with the same Geometry of H[j-BLOCS]. If the product exists set k to the place where it can be found and go to 6.

Step 9: Look for the first Product in List to be inserted.

Step 9.1: Count the number of empty positions before the position [j]
Step 9.2: if the number of empty positions is less than EH a change in model might be scheduled otherwise go to Step 9.4
Step 9.3: Look for the first Product in **List** where LastColour is its colour and JU(H(List(k))))<JA(H(List(k))). If the product exists set k to the place where it can be found and go to 5.
Step 9.4: if the number of empty positions is less than EK a change in colour might be scheduled otherwise go to Step 9.7
Step 9.5 : Look for the first product in List where JU(H(List(k))))<JA(H(List(k))).
Step 9.6: If such a product has been found set k to the place where it is located in **List**; go to Step 6
Step 9.7: If the end of the list is reached then k:=0; go to Step 4.

With respect to the previous algorithm several remarks are to be noted. First, this algorithm tries to reduce intrinsically the number of jig changes (Step 8). The reason for that come from the experienced reality. Although for the company the main direct costs to be considered is the cost of a colour change and the excess of empty blocks, the only real effect over the manpower of a given schedule is the number of jig changes that they have to perform. Therefore a schedule improving the colour change level but having too many jig changes is never going to be understood neither by the workers nor by the foreman. Therefore if a jig can be maintained it should be maintained, except in the case that the same product is to be scheduled. Second, the jig limitation is considered with variables JU in Steps 7.2 and 9.5. Obviously it is not necessary to keep control of that if the geometry of the previous loop is to be repeated (Step 8). Third, for the first loop (j<BLOCS) some of the operations do not apply but for simplicity those controls have been avoided. Fourth, it is assumed that the colour change needs more empty positions than the model change. Our experience in reality never showed the opposite situation. Fifth, there is a variant (that has been tested) that always leaves empty the last two positions of a given loop. This variant intended to allow that in the following loop the same set of jigs could be placed, reducing therefore the number of jig changes. This variant only requires small changes mainly adding a step between 4 and 5, checking if the end of the loop has been reached and if so, include two empty positions and move to step 8. It has not been added on the previous algorithm for the sake of clarity.

5. Computational experiences.

As has been described, the problem presented is an innovative problem that can be found in different industrial environments. As there is no standard set of benchmark instances for testing solving procedures, a two-level five factors full factorial experimental study was constructed. The particular case that has been described includes several characteristics that have been taken as inspiration to generate this benchmark. In this sense every problem always has 1600 units to be grouped in batches and scheduled, but depending on the problem we have different number of geometries and colours involved. This is like saying that the daily demand is about 1600 positions what can be considered as an appropriated order of magnitude.

The problems were generated according to the following two factors: the number of geometries (NrGeometries) and the number of colours (NrColours).

For the first factor two levels were defined, High (40 geometries) and Low (20 geometries). On the other hand, for the second factor there were problems with High number of colours (20) and Low number of colours (10). For each kind of problem 40 instances were generated, so that the definitive benchmark is composed of 160 problems. All of them were run with the different variants of the heuristic algorithm described in section 4.

There are some parameters related to the physical configuration of the system. The two that are relevant for us are the number of positions per loop and the number of jigs per geometry. Together with the capacity of the systems (the 1600 units per day considered before) those are decisions that affect to the problem and the resolution procedure: Number of positions per loop (100-30), maximum percentage of geometries per loop (75 % of a loop can hold the same geometry, 25% of a loop can hold the same geometry), Criteria for sorting the batches (Sorting the list by the most demanded colour or Sorting the list by the most demanded geometry).

The 160 problems generated and they were run for the 4 variants of the system configuration and for the 4 variants of the heuristic, what supposes 2560 experiments. As it is not possible to have an optimal solution for the benchmark, in order to extract valid conclusions about which variant is more efficient for every kind of problem, the indicator used was the Relative Perceptual Deviation (RPD) for every experiment defined as:

$$RPD_{\exp} = \frac{Sol_{\exp} - BestSol_{ins}}{BestSol_{ins}} \cdot 100$$

From our ANOVA analysis we may summarize the main conclusions obtained by means of Fisher Test Graphics. The main statistically significant factors were the strategy and the direction, being indifferent the criteria applied both for selection and even results rules:

It is interesting to remark that the different problems have been built considering that the total demand is constant and varying the number of geometries and colors. But the real problem has another two aspects that are defined in reality: the length of the loop and the quantity of available jigs. Comparing the two ways of ordering the list (by color or by model) we can see that the difference between the two sorting methods for the RPD is favourable to ordering by colour, but not with a big difference (less than 30%). On the other hand the values of JPD have a bigger variance (nearly 100%) between both ordering methods. We can conclude that if the focus is in reducing Jig Changes there is one ordering method much better than the other (in our case ordering by geometry). If we try to measure the effect of leaving to empty positions at the end of the loop we can see that although it halves the quantity of jig changes, the number of empty blocks is strongly affected.

If we have to assess about the length of a loop again both metrics JPD and RPD had been checked. Figure 9 show that there is a strong relation between the length of the loop and the number of empty positions, as larger that number less overall empty positions. Data of Figure 10 shows that the number of jig changes is hardly affected by the length of the loop. If comparing the relation between the Jig availability and the quantity of different geometries we can easily see that the Jig availability is relevant only to the number of empty spaces if the number of different geometries is Low. If the number of geometries is low the different procedures are more relevant to the overall solutions for both the number of empty positions and the number of jig changes. By analysing the results we can say that ordering the products considering the most demanded model and leaving 2 empty positions at the end of the loop improves significantly the number of jig changes, but worsens the number of empty positions.

Considering the problem with a real perspective we have to make two additional considerations. First, the number of empty jigs and the number of jig changes is relevant considered against the total length of the schedule. Second, the number of colour changes is an absolute value that has to be taken into account. Having a look on the first aspect the relative extra time (i.e how much time would we have to work due to empty positions) moves around a 10% between the worst schedule and the best one. But if we have a look on the relative extra time that the workers will have to work changing jigs that relation (the worst against the best) is about a 35%. The number of total color changes is about 5 times worse between the best obtained solution and the worst one.

6. Conclusions

In this paper we have addressed a scheduling problem that can be considered innovative in the field of scheduling with setups. It does take into account two different types of changeovers, those due to two consecutive batches, that has been called (horizontal setup) and those due to the relation between consecutive loops in the same position. The objective is to schedule minimizing the cost of both types of setups. The cost of the setup is related to effective cost (either solvent or workforce) but also is an opportunity cost of lost capacity. The case treated in this paper does not fall into any category of the known scheduling problems in the literature, but it bears certain similarities with cyclic scheduling, batch scheduling or flowshop sequence dependent scheduling. Three different cases have been presented, each one incorporated a new feature. A simple direct heuristic has been developed with two variants, considering two alternatives for each variant. The heuristic has been used to analyze the effects of the values of the problem (such as number of geometries and colors), and the physical configuration of the system (such as the length of the loop and the quantity of available jigs for the same geometry).

Acknowledgments

This work was developed under research projects GESCOFLOW (DPI2004-02598) supported by the Spanish National Science&Technology Commission CICYT

References

[1] A. Allahverdi, J.N.D. Gupta, T. Aldowaisan, (1999), A review of scheduling research involving setup considerations, *Omega* **27**, 219 – 239.

[2] A. Allahverdi, C.T. Ng, , T.C.E. Cheng, and M.Y. Kovalyov (2007). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*. In press.

[3] G. Mosheiov, D. Oron, Y. Ritov (2005), Minimizing flow-time on a single machine with integer batch sizes, Operations Research Letters **33**, 497 – 501.

[4] C.N. Potts and L.N. Van Wassenhove, (1992), Integrating scheduling and batching lot-sizing a review of algorithms and complexity, Journal of Operational Research Society **43**, 395 – 406.

[5] C.N. Potts and M. Kovalyov (2000), Scheduling with batching a review, *European Journal of Operational Research* **120**, 228 – 249.

[6] S. Albers and P. Brucker (1993), The complexity of one-machine batching problems, *Discrete Applied Mathematics* **47**, 87 – 107.

[7] G. Dobson, U.S. Karmarkar and J.L. Rummel, (1987), Batching to minimize flow times on one machine, Management Science **33**, 784 – 799.

[8] D. Naddef and C. Santos (1988), One-pass batching algorithms for the one-machine problem, Discrete Applied Mathematics **21**, 133 – 145.

[9] C. Santos and M. Magazine, (1985), Batching single operation manufacturing systems, Operations Research Letters **4**, 99 – 103.

[10] S. Webster and K.R. Baker (1995), Scheduling groups of jobs on a single machine, Operations Research. **43**, 692 – 703.

[11] C. Hanen, A. Munier, (1997), Cyclic scheduling on parallel processors: an overview. In: Chretienne, P., Cofman, E.G., Lenstra, J.K., Liu, Z. (Eds.), Scheduling Theory and its Applications. John Wiley & Sons Ltd

[12] Y. Crama, (1997), Combinatorial optimisation models for production scheduling in automated manufacturing systems, European Journal of Operational Research **99**, 136 – 153,

[13] J.C. Gentina, O. Korbaa and H. Camus, (2001). Problèmes d'ordonnancement cyclique. In: P. Lopez, F. Roubellat (Eds.), (Chapter 7)**,** Ordonnancement de la Production, Hermès Science, IC2 Productique, 197 – 223.