Available online at www.sciencedirect.com
SCIENCE DIRECT
Computers & Industrial Engineering 50 (2006) 76–89
computers & industrial engineering
www.elsevier.com/locate/dsw

Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry

Cristovao Silva *, Jose M. Magalhaes

Mechanical Engineering Department, University of Coimbra, Polo II, Pinhal de Marrocos, 3030-201 Coimbra, Portugal

Received 6 February 2003; received in revised form 10 February 2005; accepted 20 January 2006
Available online 23 March 2006

Abstract

In this paper, we present an industrial problem found in a company that produces acrylic fibres to be used by the textile industry. The problem is a particular case of the discrete lot sizing and scheduling problem (DLSP). In this problem, lots of similar products must be generated and sequenced in ten unrelated parallel machines, in order to minimize tool changeovers and the quantity of fibre delivered after the required due date. The company problem is original because a changeover can occur between two lots of the same product due to tool wear. We analyse the problem in detail and present an adaptation of a heuristic found in the literature to solve it. Results obtained with the proposed heuristic are compared with results that used to be obtained by the production planner, using historical data.

Keywords: Lot size scheduling; Unrelated parallel machines; tool wear; heuristics

1. Introduction

In this paper, we present a problem found in a company producing acrylic fibres to be used either isolated or combined with natural fibres, by the textile industry. The acrylic fibres are obtained in three steps: (1) dope preparation, (2) spinning and (3) cutting and packing.

In the first step, a polymer is dissolved in an organic solvent to obtain the dope, which is stored in a silo. In the spinning unit the dope is forced to pass through a set of spinnerets, a metal plate with a large number of very small holes (in the order of tens of μm). When the dope passes through the spinnerets a set of filaments is generated. Those filaments are then washed, dried and submitted to a thermal treatment. At the end of the spinning unit the fibre forms a faggot of continuous filaments. At this stage, the fibre is called tow and can be sold in this form, after being pressed in the form of a pack. The tow can be transformed by cutting it into small segments (typically 30–40 mm) to obtain a product called raw. The raw is also pressed in the form of a pack to be expedited. The raw is usually used by the cotton industry while the tow is used by the wool industry.

In terms of composition the fibres produced can be of three types: brilliant, dull or black. The first type is the one obtained directly from the dope created by the dissolution of the polymer, while the other two result from the addition of a pigment to the dope. The fibres are also distinguished by their diameter. The final diameter depends on

* Corresponding author. Tel.: +351 239 790 757; fax: +351 239 790 701.
E-mail address: cristovao@gestao.dem.uc.pt (C. Silva).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2006.01.001

the diameter of the spinnerets' orifices and the stretch imposed on the fibre along the spinning machine. The same spinneret can thus generate fibres with different diameters by changing the stretch used in the production process.

The production planning problem in this company arises in the spinning unit, because this is where the bottleneck arises in the production process.

The company produces a large variety of finished products, but all of them can be obtained from about 60 different types of fibre, generated in the spinning unit, where production planning used to be on a monthly basis, as described bellow.

At the end of a month, client orders for finished products were analysed to generate production orders for the spinning unit. Different finished products are obtained from the same fibre generated in the spinning unit. For example, two raws of fibre X can give different finished products if they are cut to different lengths. The company's information system sorts client orders so that those requiring the same fibres generate a single production order for the spinning unit. At the end of a month all the production orders generated for the spinning unit were downloaded by the production planner, who had to sequence them in the ten spinning machines by the end of the next month. Thus, the planner used to plan a set of production orders, all having the same due date.

Since, the cumulative demand for the next month could be larger than the cumulative capacity available, part of the required production could not be manufactured during the planning horizon. Therefore, the planner used to generate a production plan, leaving part of the demand out of it. The planner elaborates this production plan making an effort to minimize the number of changeovers. This first production plan, called version 0, was sent to the commercial department to be approved. This procedure would lead to a negotiation between the commercial and the production department. During this negotiation, the commercial department would propose changes to the production plan, requiring higher production of some products or the early execution of other products. The negotiation between those two departments proceeded iteratively, until a plan satisfactory to both parties was arrived at. This process could lead to the preparation of 3–5 versions of the production plan for the next month. Obviously, the last version of the plan was the one followed on the shop-floor.

Recently, in order to increase service levels, the company decided to require due dates for spinning production orders weekly, instead of the usual monthly due date. The plan is still generated for the next month, but this planning horizon is now divided in four to five deadlines. Therefore, the planner defines a schedule for the next month, trying to produce the maximum quantity of fibre by the required due date and seeking to minimize changeovers. The negotiation phase with commercial department still exists, but it is now more complex because the discussion involves not only the quantities that must be produced, but also orders that cannot be delayed.

This new situation has made the problem even more complicated, thanks too, to an increase in the number of different fibres produced and an increase in machine capacities that allows the production of a larger amount of orders. This increase in complexity meant that the company managers felt the need to develop a decision support system for production planning that could generate sequencing plans suited to the spinning unit of the company, with the aim of:

- Meeting the due dates agreed with the commercial department, or at least minimizing the quantity of fibre produced after the required due date.
- Minimizing the number of changeovers required during the plan.

This paper is organized as follows. In Section 2, we describe the scheduling problem find in the company spinning unit. In Section 3, we make a literature review about the lot size scheduling problem, referring a heuristic, which we believe can be adapted to our case study. In Section 4, we describe the algorithm developed to solve our problem and Section 5 refers how this algorithm was implemented in a decision support system. Section 6 presents some results obtained with our algorithm, comparing them to result that used to be obtained manually by the production planner. We finish with some conclusions.

2. The scheduling problem in the spinning unit

In this section, we describe the scheduling problem found in the spinning unit. We define the production system and refer the constraints imposed to our problem.

2.1. The production system

As we refer in the previous section, the scheduling problem occurs in the spinning unit. In this unit, about 60 different types of fibres are produced in ten parallel machines. The machines are dedicated and therefore a given fibre can be processed only on a pre-specified set of machines. Moreover, the machines are not all identical and consequently the production rate of a given fibre depends on the machines where it will be processed.

2.2. Tool changeovers

During the production process, changeover costs occur whenever a spinneret has to be changed. To change a spinneret the machine must be stopped for about 2 h, and this time is irrespective of the type of transition between products. Thus, we can consider a fixed setup cost whenever there is a changeover.

A changeover will occur whenever there is a transition between two products requiring different spinnerets. The production of different products can, however, be sequenced without a changeover, because some different products use the same spinneret. The transition between those products may not imply a changeover, but simply a change of variety that can be accomplished without stopping the machine. Not all the products that share the same spinneret can be sequenced without stopping the machine. This will only occur if the transition is brilliant–brilliant or brilliant–dull. Dull–brilliant sequences always imply a changeover, even if the two products share the same tool. Transitions involving black products are like those for dull products. Therefore, in the rest of the paper when we refer to dull products we are considering all the products that use pigments, i.e dull and black fibres.

A changeover can also be necessitated by tool wear. The spinnerets used in the production process have a limited lifetime, ranging from 8–45 days, depending on the type of fibre processed and the machine used. After this time the orifices of the spinneret become blocked, and the tool must be changed. Therefore, in our particular case, a changeover may occur, even if there is no transition between two different products.

2.3. Planning horizon and time interval size

The planning horizon in the spinning unit is 1 month. This planning horizon is divided in days, the smallest production run considered in our case company. This means that only one product may be produced per day and the production uses the full machine capacity. As we refer previously we consider two problems. In the first one, all production orders have the same due date set to the last day of the planning horizon. The objective is to produce the maximum quantity of fibre during the planning horizon, minimizing the number of changeovers. In the second problem, the planning horizon is divided in 4 or 5 weeks. The production orders have due dates associated to the last day of one of the considered week. The objective is still to maximize the quantity of fibre produced during the planning horizon, minimizing the number of changeovers but in this new problem we also need to minimize the quantity of fibre produced after the required due date.

2.4. Backlog

The cumulative production capacity related to a given deadline can be equal, larger or smaller than the cumulative demand up to this deadline. This means that in our problem backlogging is allowed. If a production order or part of it cannot be processed in the required week due to lack of capacity, it can be produced using capacity available in subsequent weeks. On the other hand, the capacity left in a given week after having allocated all production orders required by this week can be used to process production orders with due dates in future weeks.

3. Literature review

From the description made in the previous section, we can conclude that the problem we want to solve consists of lot sizing and scheduling a set of N different production orders in ten dedicated and unrelated parallel machines. In our case, the smallest planning unit considered is one working day. The questions for a given production order are, therefore: what machine to produce it in, when to start to produce it, and for how many days.

This problem is a particular case of the simultaneous lot sizing and scheduling problem. The simultaneous lot sizing and scheduling of several products is a problem that has attracted the attention of a number of researchers. Salomon, Kroon, Kuik, and Van Wassenhove (1991) says that this problem is interesting because it combines lot sizing (medium range planning) and job scheduling (short range planning), and this may be important in many practical situations. Many variants of lot sizing and scheduling problems have been proposed by several authors. Kuik, Salomon, and Van Wassenhove (1994) present a classification of models and provide a list of references. More recently, Staggemeier and Clark (2001) have described the different aspects founded in this type of problem and note the most common method used to solve them.

The discrete lot sizing and scheduling problem (DLSP) is a particular case of this class of problems where the planning horizon is divided into micro periods. The fundamental assumption of the DLSP is the so called 'all or nothing' production: only one item may be produced per period and, if so, production uses the full machine capacity. Therefore, the periods in the DLSP model usually correspond to small buckets such as hours, shifts or days. For a survey of this class of problems see, for instance, Drexl and Kimms (1997).

The DLSP in its 'generic' form can be formulated as a mixed integer program, see, for example, Fleischmann (1990). However, many different features can be considered. The setup costs can be fixed, product-dependent or sequence-dependent. We also can consider the existence of multiple-machines, and in this case they can be parallel machines (in a single stage) or machines in sequence (i.e. multistage). Some authors also consider the possibility of backlogging; allowing the demand to be satisfied by production in subsequent periods.

Our literature review did not find any formulation for a DLSP containing all the particular features of our case study. We decided to concentrate our attention on the work of Pattloch, Schmidt and Kovalyov (2001) since we believed that the heuristic they propose to solve their problem could be easily adapted to our case study. The problem proposed by these authors is a special case of DLSP for identical parallel machines, where only fixed setup costs are considered. In this problem, the cumulative demand up to a period t, is always smaller than or equal to the cumulative capacity up to this period, and so no backlogging is allowed. A formulation for this particular case can be found in Blazewicz, Ecker, Pesch, Schmidt, and Weglarz (1996). Blazewicz et al. (1996) present algorithms to solve this problem for the single machine case and for the identical parallel machine case.

Between the problem proposed by Pattloch et al. (2001) and the one described in this paper, we can consider four major differences:

- The cumulative production capacity related to the deadlines can be equal, larger or smaller than the cumulative demand up to this deadline. Therefore, backlogging can be allowed;
- The machines are dedicated and unrelated and therefore a given product can be processed only in a restricted number of machines and it production rate depends on the selected machine;
- The spinnerets have a limited lifetime and consequently a changeover may be due to a change in product type or to tool wear;
- Finally, different products can be sequenced without changeovers.

Despite these differences we believe that the algorithm proposed by Pattloch et al. (2001) can be adapted to our case study. The algorithm we propose is presented in the next section.

4. The algorithm

To explain our algorithm we start by defining the concept of empty space. An empty space is a set of consecutive days, in a given machine, to which no production has been assigned. When we want to assign a production order I, requiring the production of n_{j1} ton of product $j1$, using a spinneret $s1$, to be concluded by due date DDI, we can do it in seven different empty spaces, described below and represented in Fig. 1. In the description of empty space we will consider that $j*$ is a product different from $j1$ and $s*$ is spinneret different from $s1$.

($j1$, 0) Product $j1$ is being processed and a set of idle days are available afterwards, until due date DDI (see Fig. 1a).

($s1$, 0) A product $j*$ produced using a spinneret $s1$ is being processed and a set of idle days are available afterwards, until due date DDI (see Fig. 1a).

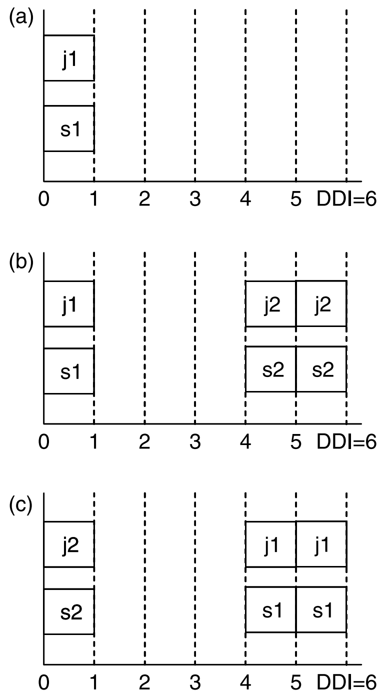

Fig. 1. Types of empty spaces.

($j1$, 0, j^*) or ($s1$, 0, s^*) The empty space starts with a product $j1$ or with a product j^* requiring the same spinneret than $j1$ ($s1$) and is limited at the end by a different product j^* using a different spinneret (s^*) (see Fig. 1b).

(j^*, 0, $j1$) or (s^*, 0, $s1$) The empty space starts with a product j^* using a spinneret s^* and is limited at the end by the product $j1$ or by a product j^* using the same spinneret than $j1$ ($s1$) (see Fig. 1c)

other The empty space is not limited, either at the beginning or at the end, by the product $j1$ or by a product using the same spinneret $s1$.

The algorithm we propose works in three steps: (1) selecting a production order, (2) choosing an empty space where the production order will be allocated and (3) allocating the production order to the selected empty space. Those three steps are described in the following sections.

Before the lot sizing and scheduling procedure, a list of all the production orders that must be sequenced is generated. This list is divided into various sub-lists, one for each due date in the planning horizon. For example, suppose that we have demand for due dates (*DD*) 1, 2 and 3. In this case, three lists are generated, one containing all production orders with *DD1*, a second one for all production orders with *DD2* and finally a third list containing all production orders with *DD3*.

The planning horizon is then divided into weeks, defined by the due dates referred to above. For the example given the planning horizon would be divided into three weeks: the first one extending from the first day of the desired plan to *DD1*; the second one extending from *DD1* to *DD2*, and the third one between *DD2* and *DD3*. Each of these periods is then divided into days, the shortest planning period considered by the planner.

The algorithm starts sequencing the production orders on the list with the shortest due date, trying to allocate them to the first week of the plan. If some production orders are not planned in this period (due to lack of capacity in the first week) they are kept in memory, and could occupy some capacity in future periods. On the other hand, if after sequencing all the production orders there is still some capacity available, it could be used by production orders with higher due dates. The above procedure is repeated for all the other lists, in order of lengthening due dates.

4.1. Selecting a production order

The production orders from a given list are allocated to the machines in the order in which they appear on the list. The production orders on a list can be sorted using various priority rules, described below.

4.1.1. Number of machines where the production order can be processed

This rule sorts the production orders by increasing number of machines where they can be produced. This rule is intended to avoid leaving products that can be produced on just a few machines out of the plan.

4.1.2. Size of the production orders

With this rule the production orders are sequenced in decreasing order of their size, measured in ton. With this rule we believe that the total amount of product processed will be maximized. On the other hand, this rule can have the disadvantage of leaving a large number of small-scale production orders, or production orders that can be produced only on a small number of machines, out of the plan.

4.1.3. Priority given to products from the previous plan

With this rule the production orders sequenced last in the ten machines from the plan of the previous month are checked. After this operation the algorithm chooses production orders of the same type to be sequenced first. This will make it possible to move from one plan to another with a minimum number of changeovers.

4.1.4. Priority defined by the user

The user can define the production orders to which he wants to assign priority. With this rule these production orders will be sequenced first. This rule is particularly interesting in the phase of negotiation between the production and commercial department.

The plan may be executed considering more than one priority. In this case, the production orders are selected in the order defined by the first priority, and the other selected priorities are used to untie production orders. The priority(ies) selected to execute the production plan will obviously influence the obtained sequence. Thus, different production plans can be obtained for the same set of production orders simply by changing the selected priorities and their order.

In the following sections, we present a pseudo code to explain how to allocate a production order I, requiring the production of product $j1$ to an empty space, and how to choose this empty space. The notation used in the pseudo code is:

- DDI due date for production order I;
- $n_{j1,I}$ is the quantity (in ton) of product $j1$ required to satisfy production order I;
- $s1$ is the spinneret required to process the product $j1$;
- u is the length of the empty space, i.e. the number of consecutive idle days;
- h is the first day of the empty space;
- $p_{j1,m}$ is the production rate of product $j1$ in machine m (in ton/day), machine m is the one where the empty space was selected;
- l_{s1} is the lifetime of the spinneret $s1$ expressed as a percentage; a new tool has a lifetime of 100%;
- $w_{s1,j1,m}$ is the wear, expressed as a percentage, of spinneret $s1$ when processing product $j1$ in machine m for 1 day.

4.2. Allocating the production order to the selected empty space

The selected production order is allocated to an empty space, using backward or forward scheduling. Figs. 2 and 3 give the pseudo code corresponding to both options. It can be seen from the pseudo code described above that tool lifetime is taken into account when the job is allocated to an empty space.

4.3. Selecting an empty space for production order allocation

After the selection of a production order the algorithm assigns it to a machine following the procedure described below. In a first step, called block 0 and shown in Fig. 4, the algorithm tries to find if the product ($j1$) corresponding to

```
DO
        Assign production of j1 to h
        u = u – 1
        h = h + 1
        n_{j1,1} = n_{j1,1} – p_{j1,m}
        l_{s1} = l_{s1} - w_{s1,j1,m}
LOOP UNTIL
        u = 0 OR n_{j1} = 0 OR l_{s1} = 0
```

Fig. 2. Pseudo code for forward scheduling.

```
DO
        Assign production of j1 to h + u - 1
        u = u – 1
        h = h - 1
        n_{j1,1} = n_{j1,1} – p_{j1,m}
        l_{s1} = l_{s1} - w_{s1,j1,m}
LOOP UNTIL
        u = 0 OR n_{j1} = 0 OR l_{s1} = 0
```

Fig. 3. Pseudo code for backward scheduling.

the selected production order is being processed in a machine. This procedure is designed to minimize the number of changeovers. If the product is not being processed in a machine the algorithm tries to find if a different product (j^*) using the spinneret required by $j1$ ($s1$) is being processed. If the above two situations are not found the product is assigned to the larger empty space occurring in the machines where the product can be produced, using backward scheduling.

If product $j1$ is being processed in a given machine, the algorithm plans it according to the pseudo code (block 1) given in Fig. 5. With this procedure, we intend to keep working with the same product for as long as possible, in order to avoid changeover costs.

If product $j1$ is not being processed in any machine, but a different product (j^*) is using the same spinneret $s1$, then the product is sequenced according to the pseudo code (block 2) given in Fig. 6.

```
IF there is an empty space of type [j1, 0] OR [j1, 0, j*] OR [j*, 0, j1]
        THEN BLOCK 1
ELSE IF there is an emptyspace of type [s1, 0] OR [s1, 0, s*] OR [s*, 0, s1]
        THEN BLOCK 2
ELSE select the empty space with maximum u
        Backward Scheduling
END IF
```

Fig. 4. Pseudo code for block 0 of the proposed algorithm.

IF there is an empty space of type [*j1, 0*] **OR** [*j1, 0, j**]

THEN Forward Scheduling

ELSE IF there is an empty space of type [*j**, *0, j1*]

THEN Backward Scheduling

END IF

Fig. 5. Pseudo code for block 1 of the proposed algorithm.

The pseudo code in Fig. 6 shows our concern to guarantee the correct utilization of the spinnerets.

To illustrate the developed algorithm let us consider a simple example. Consider that we intend to schedule 6 production orders in three machines. The production orders characteristics are presented in Table 1. To maintain simplicity we will assume that the production rate for any product in any machine is 10 ton per day and any spinneret is consumed at a rate of 10% of its available lifetime for each processing day. At time 0, the last day of the previous schedule, machine 1 is processing product *a*, brilliant, requiring spinneret *X*, which have 80% of its available lifetime, machine 2 is processing product *f*, dull, requiring a spinneret *Z* with 40% of its available lifetime and machine 3 is processing product *g*, brilliant, using a spinneret *X* having 70% of its available lifetime. We will consider that the production orders from the first list, the one containing all production orders having due date 7, are sorted in the following way: I, II and III. The production orders from the second list are sorted as follow: IV, V and VI.

Production order I is the first to be considered. Product *a* is being processed in machine 1, thus an empty space of type (*j*1, 0) with a length *u* of 7 days is available in this machine. We use forward scheduling to allocate 70 ton (7 days) of product *a* to machine 1. In the first week of the planning horizon product *a* is not being processed by any other machine. Nevertheless, spinneret *X*, required to produce *a*, is allocated to machine 3, originating an empty space of type (*s*1, 0). Spinneret *X* is processing a product brilliant and the product we intend to allocate (*a*) is also brilliant. Accordingly to Block 2 of our algorithm we have a condition (*B*; [*s*1, 0]; *b*). This means that the remaining 20 ton of product *a* from production order I are allocated to machine 3 using forward scheduling. Production order II is the second to be considered. Product *b* can only be allocated to machine 2. Product *b* is not being processed by machine 2 and spinneret *Y* is not allocated in it. Therefore, the 70 ton of product *b* are allocated to machine 2 using backward scheduling. We must now allocate 30 ton of product *c* required to satisfy production order III. Spinneret *X* is allocated to machine 3 processing a product brilliant. The product we want to allocate is dull. Thus, accordingly to Block 2 we have a condition (*B*; [*s*1, 0]; *d*) and product *c* is allocated to machine 3 using forward scheduling.

READ type of product being processed by *s1*- [D (dull); B (brilliant)]

READ available empty space type - ([*s1, 0*]; [*s1, 0, s**]; [*s*, 0, s1*])

READ type of product to be inserted in the emptyspace – [d (dull); b (brilliant)]

IF (D ; [*s1, 0*] **or** [*s1, 0, s**] ; d) **OR** (B ; [*s1, 0*] **or** [*s1, 0, s**]) ; d) **OR** (B ; [*s1, 0*] or [*s1, 0, s**] ;b)

THEN Forward Scheduling

ELSE IF (D ;[*s*, 0, s1*] ; d) **OR** (D ; [*s*, 0, s1*] ; b) **OR** (B ; [*s*, 0, s1*]) ; b)

THEN Backward Scheduling

ELSE select the empty space with maximum u

Backward Scheduling

Fig. 6. Pseudo code for block 2 of the proposed algorithm.

Table 1
Characteristics of example production orders

Production order	Product	Spinneret	Type	Quantity (ton)	Machines	Due date
I	*a*	*X*	*B*	90	1, 3	7
II	*b*	*Y*	*B*	70	2	7
III	*c*	*X*	*D*	30	1, 3	7
IV	*d*	*Y*	*D*	140	1, 2, 3	14
V	*c*	*X*	*D*	20	1, 3	14
VI	*e*	*Z*	*B*	80	1, 2	14

Having sequenced all the production orders with due date 7 we will now consider the production orders with due date 14. Production order IV is the first one to be considered. Product *d* is not allocated to any machine but spinneret *Y* is being used in machine 2 processing a product brilliant. Thus, we have in machine 2 a condition (*B*; [*s*1, 0]; *d*) and we allocate 30 ton of product *d* to this machine using forward scheduling. This allocation exhaust the spinneret available lifetime. From this point spinneret *Y* is not allocated to any machine and we choose the largest empty space, in machine 3, to allocate 90 ton of product *d* using backward scheduling. The remaining 20 ton of product *d* required to complete production order IV are allocated to machine 1 where the largest empty space is now available. 10 ton of product *c* from production order V are sequenced in machine 1, where we have a condition (*B*; [*s*1, 0]; *d*), exhausting the spinneret's available lifetime. The remaining 10 ton of product *c* from production order V are allocated, using backward scheduling, in machine 1 where the largest empty space is available. Finally, we allocate 30 ton of product *e* in machine 1 and 40 ton in machine 2 in the remaining available empty spaces.

The resulting schedule is presented in Fig. 7a. 6 changeovers are required, two of them resulting from tool lifetime constraint. 20 ton of product *d* are produced in week 1 but are only required by the end of week 2 and 10 ton of product *e* are kept out of the schedule due to lack of capacity.

As we refer previously, the order by which the production orders are sorted in the lists influence the obtained schedule. Let us consider, for example, that the production orders in our example are now sorted in the following way: list 1—II, I, III and list 2—VI, V, IV.

For this new situation the schedule obtained for the plan first week is equal to the one we obtained in the previous example. Nevertheless, the schedule for the production orders from second list is different. We start with production order VI requiring the allocation of product *e* produced with spinneret *Z*. Since, spinneret *Z* is not being used in any machine, we use backward scheduling to allocate 70 ton of product *e* to machine 1 and 10 ton in machine 2. The 20 ton of product *c*, from production order V, are allocated to machine 3, using forward scheduling, where an empty space of

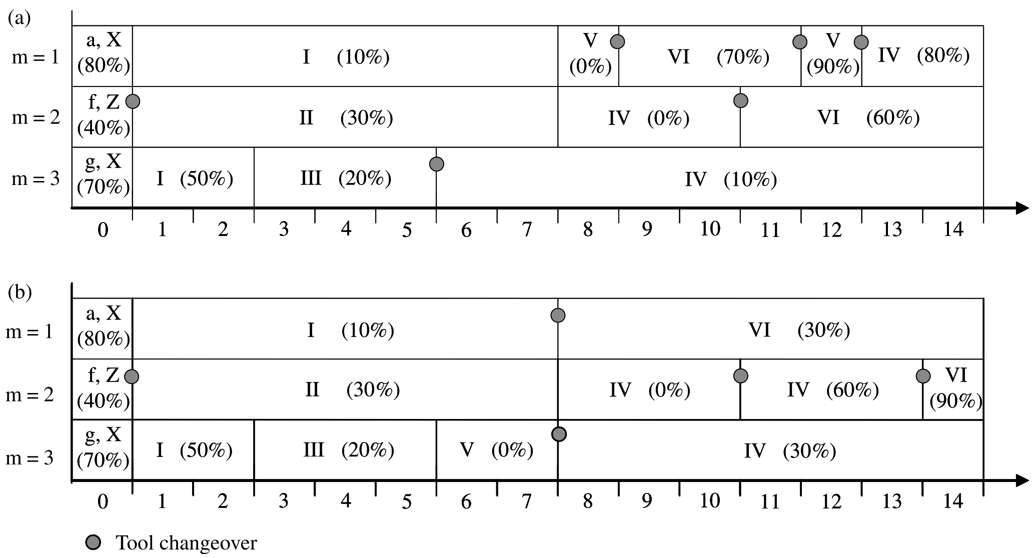

Fig. 7. Resulting schedules from the example.

type ($j1, 0$) is available. 30 ton of product *d*, from production order IV, are allocated to machine 2 where a condition (B; $[s1, 0]$; d) is present, exhausting the spinneret available lifetime. 70 ton and 30 ton of product *d* are backward scheduled in machines 3 and 2, respectively, where we have available empty spaces.

This new schedule is presented in Fig. 7b. 5 changeovers are required, 20 ton of product *c* are produced in week 1 but are only necessary by the end of week 2 and 10 ton of product *d* are kept out of the schedule due to lack of capacity.

5. The decision support system

The algorithm presented in the previous section was implemented in a scheduling decision support system (SDSS) to be used by the production planner. Before the implementation of the SDSS the detailed production plans for the spinning units were prepared, as explained in the next paragraph.

The company information system produces an aggregate production plan for each company department. For the spinning department this production plan indicates which fibre should be produced in which quantity by which deadline, for the next month. At the end of each month, the planner used to use this information to draw up a production schedule manually, indicating the batch size in which each of the required fibres should be produced, the machine it is assigned to, and the time interval in which production has to take place. It was for the preparation of this production schedule that the SDSS supporting the algorithm described in the previous section was developed.

The SDSS is composed of the above algorithm, a user interface and a data base. Fig. 8 shows simplified aspects of the data model, in particular the main data entities and attributes, along with the chief relationships. A *request* consists of a production order generated by the company information system, indicating the product required, in which quantity and for which due date. The required *product* has a spinneret, a type (tow or raw) and a dope (black, dull or brilliant) associated to it, and its processing time will depend on the *productivity* of the machine where it is to be produced. For each *machine* the data base keeps information about the product being processed and the corresponding spinneret lifetime at the time when the plan is to be prepared. The *spinneret lifetime* depends on the machine where it is used and on the product being processed. Finally, when a production scheduling *plan* is drawn up, information on it is kept on the data base.

To implement the SDSS in our case study company we had to connect it to the company information system. The connection between the SDSS and the company information system is shown in Fig. 9. A program to establish the interface between the two systems was designed. Each time the planner wishes to prepare a production plan for the spinning unit, he must activate the systems interface. This program downloads the spinning production orders (requests) and the corresponding product data for the spinning unit and updates the *request* and *products* tables from the SDSS database. After this step the user, through the SDSS interface, updates the *machine* table, indicating which product is being processed by each machine and the available lifetime of the corresponding spinneret. With this information, and using the algorithm presented in the previous section, the SDSS generates the detailed production schedule, presented in the form of a Gantt chart, as shown in Fig. 10.

The detailed production schedule is then uploaded to the system's interface, which translates it into a set of data readable by the company information system that will be used to control the execution of the plan.

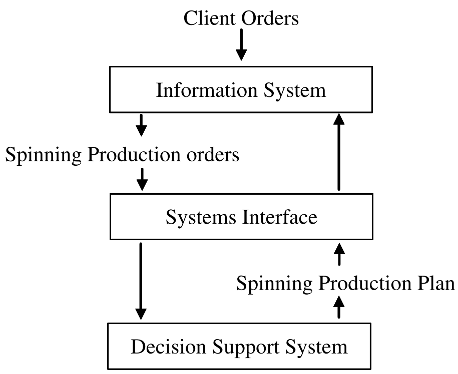

Fig. 8. Aspects of the developed data model.

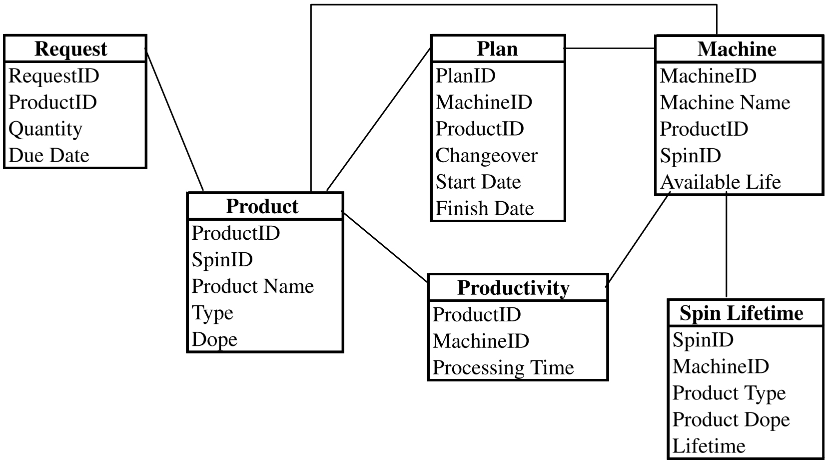

Fig. 9. Connection between the SDSS and the company information system.

6. Results

To test the decision support system referred to above we compared the plan obtained by the planner without the SDSS, which we shall call the real plan, with the plan generated using the SDSS, using historical data. The historical data chosen were the firsts 4 months of 2001, to test the SDSS for a monthly due date, and the first three months of 2002, for weekly due dates. Tables 2 and 3 give some details from the data sets used to test our algorithm. In Table 2, for the four plans with monthly due date, we indicate the number of production orders, the number of different products required, the number of different spinnerets needed to manufacture all the products. Note that in the case of a monthly due date each production order corresponds to a different product. Columns 5 and 6 show the percentage of products requiring each type of dope (dull or brilliant). The last two columns indicate the number of products versus the number of machines where they can be processed. For example, for plan 1, of the 19 products required 2 can be processed in only

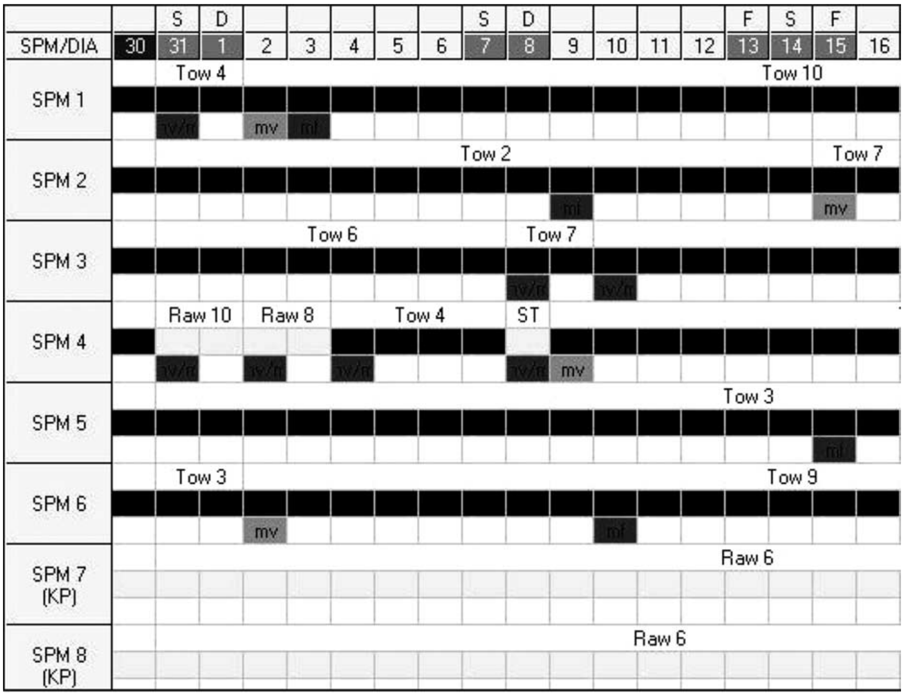

Fig. 10. Detail of the SDSS user interface showing the production plan for the spinning unit.

Table 2
Details of the data sets used to test the SDSS for monthly due dates

Plan	Production orders	Different spinnerets	Different spinnerets	Dope (% of products)		Machines	Products
1	19	19	9	*D*	38	2	2
						4	3
				B	62	6	10
						10	4
2	14	14	5	*D*	54	2	1
						4	3
				B	46	6	7
						10	3
3	25	25	8	*D*	58	2	3
						4	7
				B	42	6	12
						10	3
4	15	15	5	*D*	40	2	2
						4	4
				B	60	6	6
						10	3

2 machines, 3 products can be processed by 4 machines, 10 products can be produced in 6 machines and 4 products can be processed by all the available machines.

Table 3 gives the same information about the data sets used to test the algorithm when weekly due dates are considered, but in this case columns 7–10 have been added to give information about the division of work by due date. This additional information consists of the number of due dates to be considered, the length of the different planning periods, the number of production orders to be concluded by each due date and the corresponding percentage of production required. For example, for the first plan it can be seen that the production orders will be divided into four lists. The first one is composed of 5 production orders, corresponding to 8% of the total production required during the month, to be concluded by due date 1. The time elapsing between the start of the plan and the first due date is 3 days. The second list is composed of 9 production orders, corresponding to 18% of the total production required, to be concluded by due date 2. The time elapsing between due date 1 and due date 2 is 7 days. The third list is composed of 10 production orders, representing 29% of the total production required, to be produced by due date 3. The period between due date 2 and due date 3 is 7 days. Finally, the last list is composed of 9 production orders, representing 45% of total production required, to be produced by due date 4, corresponding to the last day of the production plan. The time elapsing between due date 3 and due date 4 is 13 days.

Results for both sets are presented in the following sections. We compare the results obtained with the DDS to version 0 of the plan produced by the planner.

Table 3
Details of the data sets used to test the SDSS for weekly due dates

Plan	Production orders	Different products	Different spinnerets	Dope (% of products)		DD	Length (days)	Production orders	Quantity (%)	Machines	Not produced
1	33	25	11	*D*	40	1	3	5	8	2	3
						2	7	9	18	4	5
				B	60	3	7	10	29	6	12
						4	13	9	45	10	5
2	46	24	9	*D*	54	1	6	9	8	2	2
						2	7	12	20	4	4
				B	46	3	7	12	20	6	14
						4	8	13	52	10	4
3	29	16	6	*D*	68	1	3	4	4	2	2
						2	7	5	18	4	3
				B	32	3	7	8	40	6	9
						4	7	7	38	10	2

Table 4
Results obtained for a monthly due date

		Real		Decision support system		
Plan	Production Orders	Changeovers	Not produced (%)	Changeovers	Not produced (%)	Priority rule
1	19	14	11.1	12	10.2	B
				13	11.8	c, a
				13	14.4	c, a, b
2	14	9	7.3	8	5.5	B
				10	4.6	c, a
				9	3.7	c, a, b
3	25	13	8.8	10	8.2	B
				12	6.9	c, a
				13	7.4	c, a, b
4	15	11	16.3	9	13.4	B
				10	15.4	c, a
				10	15.6	c, a, b

6.1. Results for a monthly due date

The results obtained for a monthly due date are presented in Table 4, which gives the number of production orders for each plan, the number of changeovers and the percentage not produced due to lack of capacity, for both the real plan and for the plan obtained with the SDSS, and the priority rule(s) used to generate the plan.

The results given in Table 4 show that the algorithm presented in this paper can help to improve the production plans of the company. The number of changeovers is usually less than or equal to those found in the plan that used to be implemented by the production planner. We can also confirm that the plans generated by the SDSS led to an increase of the quantity produced, essentially due to a better allocation of the products to the machines. The algorithm tends to allocate the products to the more productive equipment.

The best results in terms of tool changeovers are obtained using priority rule b. Nevertheless, during the test phase the production planner told us that the plans generated using this rule, were not the best, because a large number of products required in small quantities are left out of the plan. The preferred plans for the user are those obtained with rule c, followed by rule a. Those plans allow the production of a larger variety of products, and avoid tool changeovers at the beginning of the planning horizon.

6.2. Results for weekly due dates

Results obtained in the tests where weekly due dates were used are presented in Table 5. In this table, besides the data presented in Table 4, we also present the quantity planned with delay. It can be seen that for this second set of tests the number of production orders that must be sequenced is larger. This is due to an increase in the number of different fibres produced by the company from 2001 to 2002. This increase in the number of production orders is also due to the fact that

Table 5
Results for weekly due dates

		Real			Decision support system			
Plan	Production Orders	Changeovers	Not produced (%)	Delays (%)	Changeovers	Not produced (%)	Delays (%)	Priority rule
1	33	19	14.2	2.1	17	10.9	1.2	b
					17	13.8	2.4	c, a
					18	14.8	1.9	c, a, b
2	46	24	12.8	3.4	21	11.2	2.8	b
					22	12.9	2.9	c, a
					23	12.2	3.3	c, a, b
3	29	17	6.7	0.9	15	5.2	1.3	b
					17	6.9	1.1	c, a
					16	6.4	1.2	c, a, b

we are considering more due dates. In this case, the same product required for two different due dates generates two production orders. The information presented in Table 5 is the same as that in Table 4. A new column has been added (Delays) where the quantity of product planned to be produced after the required due date is given.

Once again it can be seen that the proposed algorithms can be useful to the company. A reduction in tool changeovers is found for almost all combinations of priority rules. It can also be seen that the percentage of demand not planned until the end of the planning period is similar to that obtained by the planner before the system's implementation. The results also show that there is a very small percentage of production concluded after the required due dates for both cases: the historical plans and SDSS plans. This is due to the fact that the demand is properly balanced around the different due dates. Thus, usually, a product that cannot be planned by its due date is not planned in future periods.

The results presented above show that the utilization of the scheduling decision support system can lead to better initial production plans for the spinning unit of the company analysed. Based on these results the company managers decided to implement the system in the company, and use it to generate the production plans. The utilization of the SDSS also provides other advantages.

One of these improvements is the possibility of generating a set of different plans in a short time, changing the priority rule(s) used to sort the production orders. Thus, the time the planner used to spend generating a production plan, is now used to generate a set of different plans, choosing the best one, and operating it in order to find further improvements.

The system is also very useful to the negotiation between the production and the commercial department. Cutting the time needed to produce the production plans reduces the time needed to the referred negotiation. Furthermore, when the production planner receives feedback from the commercial department, he can easily generate a new plan that satisfies the commercial requisites, choosing to sort the production orders by the importance given them by the commercial department. It is expected that, in the future, a version of the SDSS can be made available to the commercial department, in order to help them to understand the impact of their decisions on the production department. This could help the commercial department to negotiate with the clients.

7. Conclusions

In this paper, we have presented an industrial problem found in a company that produces acrylic fibres for the textile industry. This company needs to draw up production plans for the spinning units, which consist of ten dedicated and unrelated parallel machines. The objective of the production plan is to minimize the number of tool changeovers, while trying to meet the required due dates.

To address this problem we have adapted a heuristic method found in the literature to our case study.

A scheduling decision support system was developed to support the proposed algorithm and this was tested in the company. The results obtained during the test phase shows that improvements can be obtained. These results, plus the facts that the scheduling decision support system we developed can deliver new sequences in a very short time, and its usefulness to the negotiation phase with the commercial department led the company to decide to implement the SDSS. The SDSS has thus been implemented in the company and is being used to generate the required production plans for the spinning unit.

References

Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., Weglarz, J. (1996). *Scheduling computers and manufacturing processes*. Berlin: Springer.

Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling—Survey and extensions. *European Journal of Operational Research*, *99*, 221–235.

Fleischmann, B. (1990). The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, *44*, 337–348.

Kuik, R., Salomon, M., & Van Wassenhove, L. N. (1994). Batching decisions: Structure and models. *European Journal of Operational Research*, *75*, 243–263.

Pattloch, M., Schmidt, G., & Kovalyov, M. Y. (2001). Heuristic algorithms for lot size scheduling with application in the tobacco industry. *Computers and Industrial Engineering*, *39*, 235–253.

Salomon, M., Kroon, L. G., Kuik, R., & Van Wassenhove, L. N. (1991). Some extensions of the discrete lotsizing and scheduling problem. *Management Science*, *37*, 801–812.

Staggemeier, A.T., Clark, A.R. (2001). A survey of lot sizing and scheduling models. *Proceedings of the 23rd Annual Symposium of the Brazilian Operational Research Society (SOBRAPO), Brazil* (pp. 938–947).