Abstract — In this paper, a problem of scheduling tasks in uniform parallel machines with sequence-dependent setup times is presented. The problem under analysis contains a particular set of constraints, including equipment capacity, task precedences, lot sizing and task delivery plan.

The complexity of the mathematical programming model developed for this problem hasn't permitted to find one solution using optimising methods and so the authors have developed a heuristic based on the simulated annealing algorithm which allows obtaining nearly-optimal solutions.

Index Terms—Scheduling parallel machine, uniform parallel machine, simulated annealing, sequence dependent setup times.

I. INTRODUCTION

The scheduling problem for uniform parallel machines with sequence dependent setup times should be considered as an operational planning problem, occurring in different types of industries, namely in the textile industry.

In the first part of this work, the problem under analysis is defined, comparing it with other types of scheduling problems referred in literature and focusing the aspects that make them unique (see Table I).

The mathematical programming model developed by the authors for the problem of scheduling in uniform parallel machines with sequence dependent setup times will be later presented, defining the proposed objective function and the constraints for the problem.

The complexity of the studied model doesn’t permit to find the optimal solution. To solve this, the authors used the simulated annealing algorithm to obtain “nearly-optimal” solutions for the problem.

This heuristic has been incorporated in a tool to support decision makers and some computational results will later be presented to show the good performance reached by using such heuristic.

This work contributes to the existing literature by presenting a new tool to solve a problem that occurs in real industrial environments. This is the reason why a textile industry example was chosen to inspire the present work.

The results obtained through the application of this particular heuristic show the importance of using a structured approach in scheduling the production, in

F. Charrua Santos is with the Department of Electromechanical Engineering, University of Beira Interior, Calçada Fonte do Lameiro, Covilhã, Portugal, phone: +351 275 329 754; fax: +351 275 329 972, e-mail: Bigares@ubi.pt.

Pedro M. Vilarinho is with the Department of Economics, Management and Industrial Engineering, University of Aveiro, Campo Universitário de Santiago, 3810-193 Aveiro, Telf: +351 234 370 025; e-mail: pvil@ua.pt

Table I – Problems referred in literature

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Year</th>
<th>Parallel Machine Problem</th>
<th>Objective Function</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>1999</td>
<td>Identical</td>
<td>Earliness (E) Tardiness (T) Penalties (P)</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>[8]</td>
<td>2003</td>
<td>Uniform</td>
<td>MM</td>
<td>Optimal Lexicographic Search</td>
</tr>
<tr>
<td>[12]</td>
<td>2006</td>
<td>Identical</td>
<td>MM</td>
<td>SA</td>
</tr>
<tr>
<td>[14]</td>
<td>2006</td>
<td>Identical</td>
<td>MM</td>
<td>Lower Bounding Strategies</td>
</tr>
<tr>
<td>[16]</td>
<td>2006</td>
<td>Identical</td>
<td>MM</td>
<td>SA</td>
</tr>
<tr>
<td>[17]</td>
<td>2006</td>
<td>Identical</td>
<td>MM</td>
<td>Branch and Bound (BB)</td>
</tr>
<tr>
<td>[18]</td>
<td>2006</td>
<td>Single machine</td>
<td>E/T</td>
<td>Heuristic Tailored and SA</td>
</tr>
<tr>
<td>[19]</td>
<td>2007</td>
<td>Unrelated</td>
<td>MT</td>
<td>BB</td>
</tr>
<tr>
<td>[20]</td>
<td>2007</td>
<td>Generalized</td>
<td>MT</td>
<td>Hybrid Metaheuristic Approach,</td>
</tr>
<tr>
<td>[21]</td>
<td>2007</td>
<td>Unrelated</td>
<td>Minimizing the Weighted Tardiness</td>
<td>Six Tabu Search</td>
</tr>
<tr>
<td>[22]</td>
<td>2008</td>
<td>Identical</td>
<td>MT</td>
<td>BB</td>
</tr>
<tr>
<td>[23]</td>
<td>2008</td>
<td>Generalized</td>
<td>Review</td>
<td></td>
</tr>
<tr>
<td>[25]</td>
<td>2008</td>
<td>Unrelated</td>
<td>Due Dates and Weighted Jobs</td>
<td>Exact</td>
</tr>
</tbody>
</table>
comparison with the results obtained in a reference plant where mainly “ad hoc” actions were normally taken.

II. CHARACTERIZATION OF THE PROBLEM

The scheduling production problem has largely been analysed by the scientific community and these particular problems are frequently joined together according to the industrial environment characteristics found in each particular site.

In this work the classification used by Allahverdi et al. [1] will be used, subdividing the scheduling problems as follows: Single machine; Parallel machines; Flow Shop Job shop; Open shop.

The problem of scheduling parallel machines was studied by Mokotoff, (2001) [18]. In this review paper, this type of problems is presented according to the following criteria: Identical parallel machines; Uniform parallel machines and Unrelated parallel machines

Generally is accepted that in some types of industrial environments, particularly in the processes industry, sequencing is relevant, but on the other hand other researchers say the opposite stating that the optimization efforts are systematically denied by the reality found in each plant and enterprise.

In the following sections of this work, this is supported by the experience acquired in fabrics producers and its main purpose is to evaluate how important the scheduling problem is to the productive process efficiency in this particular case of industrial environment.

The study has allowed identifying a production lot sizing and scheduling problem associated with the planning stage in the weaving area of fabrics manufacturing.

According to the adopted classification referred above, the problem now being analysed fits entirely in the type of lot sizing and scheduling problem associated with the planning stage in the weaving area of fabrics manufacturing.

According to the search made in the specialized available literature, a lot of situations concerning identical parallel machines with setup times and sequence dependent, width restrictions and tardiness penalties.

According to the search made in the specialized available literature, a lot of situations concerning identical parallel machines with setup times and sequence dependent [18], have been found, but none of them considers the lot sizing and scheduling restrictions associated neither with the equipments nor to tardiness penalties. As a result of such situation, is well worth to make a more detailed analysis in the search of the best solution for such drawback.

It is now important to refer the problem concerning types of setups associated to scheduling: the problem deals with job sequences, material pieces, weaving equipments and looms. Equal material pieces, being sequenced in the same machine, are joined in batches, so the setup time between equal pieces, i.e., for pieces belonging to the same batch, is zero.

Nevertheless, when the amount of equal pieces reaches a certain established value, the batch limit, a setup should then take place. The batch limit varies according to the type of piece of material. We can now conclude, (this is one point that diverges from what has been found in literature), that for equal pieces, having a continuous sequence in the same machine, two preparation times can take place: a) it will be zero, if the limit dimension has not been reached; b) it will have a certain value (the set-up time) if any other situation occurs.

III. PROBLEM STATEMENT AND FORMULATION

The generalized version of the problem can be described in the same way as when occurs a problem of tasks sequence in parallel uniform machines, having different delivery times and setups variable times between tasks, and also depending on their sequence in the queue.

The problem under analysis has the following characteristics:

1) the throughput is stable along the time
2) the demand is known in the planning horizon.
3) the overall production can adequately be represented by a limited set of pieces (p=1, ..., P)
4) production is made in a set of machines joined together according to their characteristics in G groups of machines (g=1, ..., G) each one having NMg machines
5) to each piece p is associated a delivery date dp.
6) any piece delivered after the established delivery date, implies a certain penalty, expressed by a factor ρ for each previously established time unit of delay.
7) each piece is processed in one only machine of a defined group.
8) each piece p can be processed in machines belonging to any group which has the required characteristics for the processing, considering tppg the processing time of the piece p in one of the machines belonging to the group g (p=1, ..., P and g=1, ..., G).
9) the setup time of each machine that processes the piece j after having produced the piece i is know and represented by sij and is independent of the machine in which the piece has been processed , but always associated to the characteristics of the sequence of the pieces.
10) the setup time of the machine that produces the piece i in the beginning of the queue is known and represented by $s_{0i}$.
11) Lamax represents the maximum amount of pieces than can be joined and processed together continuously without any setup operation being required.
12) Each piece has its own assembly and colour and when pieces that are processed continuously in the same machine have the same assembly and colour there is no need for a setup operation. The amount of different assembly combinations is A, and Na (a=1,…,A) is the set of pieces having the same assembly and colour (having a as reference).

13) M is an large positive number

14) $\beta_{ig}$ = \begin{cases} 1 & \text{if the piece } i \text{ can be processed in the group } g \\ 0 & \text{otherwise} \end{cases}

Before providing a mathematical formulation, the following variables have be defined:

- cp date of conclusion of the piece p
- tp tardiness of the piece p
- s’ij setup time for the task j, when this is proceeded after task i.
The problem can then be modelled as one of mixed integer programming:

\[
\min \sum_{i} \sum_{k} y_{ikg} \cdot p_{ig} + \sum_{i} \sum_{j} x_{ijkg} \cdot s_{ij} + \sum_{i,j} s_{ij} \cdot R_{akg} + \rho \sum_{i} t_{i}
\]

\[
\sum_{j} x_{0jkg} = 1, k = 1, \ldots, M_{g}; g = 1, \ldots, G
\]

\[
c_{j} + M_{i}(1 - x_{0jkg}) \geq s_{ijg} + p_{ig}
\]

\[
\sum_{i} x_{ijkg} = y_{jk}, j = 1, \ldots, P; k = 1, \ldots, M_{g}; g = 1, \ldots, G
\]

\[
\sum_{i,i+j} x_{ijkg} = y_{jk}, i = 1, \ldots, P; k = 1, \ldots, M_{g}; g = 1, \ldots, G
\]

\[
c_{i} - d_{i} \leq t_{i}, i = 1, \ldots, P
\]

\[
\sum_{k} y_{ikg} \cdot \beta_{kg} = 1, i = 1, \ldots, P
\]

\[
y_{ikg} + y_{jkg} - 1 \leq z_{i}, i = 1, \ldots, P; k = 1, \ldots, M_{g}, g = 1, \ldots, G
\]

\[
s_{ij} = s_{ij}, z_{ij} \in N_{a}, a = 1, \ldots, A
\]

\[
\sum_{v_{ijk}} x_{ijkg} \leq R_{akg}, \forall i, j \in N_{a}, a = 1, \ldots, A
\]

In the objective function, 1) is the total production time and is the sum of the following four items: i) delay time; ii) setup time between different pieces; iii) setup time between equal pieces; iv) processing time.

- the delay time is given by the sum of all the delays occurred during the planning horizon, multiplied by a penalty factor.
- the setup time is given by the sum of all the setup times during the planning horizon.
- the processing time is given by the sum of all the pieces processing times included in the planning analysis.

The model constraints can be interpreted as follows:
2) constraints ensuring that only one initial preparation can take place in each machine k belonging to group g
3) constraints ensuring that the conclusion time of piece j, when processed in the beginning of the queue of machine k, belonging to group g, is equal to the initial setup time of piece j plus the processing time of piece j in the machine g.
4) constraints ensuring that if the setup time for piece j takes place in one k machine belonging to group g, then j is either preceded by another piece, or from the initial position and so will be processed in machine k of group g.
5) constraints ensuring that if the preparation for piece j occurs in one machine k of group g, then j either is preceded from another piece or is situated in the last position and so will be processed in machine k of group g.
6) constraints ensuring that the processing of each piece once started cannot be interrupted.
7) constraints ensuring that the delay time of piece i is given by the difference between the conclusion date and the delivery date.

From the general conditions of the problem can be concluded that \( t_{i} \) is a positive number and so delay only happens when \( c_{i} \) is larger than \( d_{i} \).
8) constraints ensuring that each piece is manufactured only once and in a compatible machine.
9) constraints ensuring that if one piece i is processed in the machine k belonging to group g, and another piece j is processed in the machine k of the same group g, then both pieces are processed in the same machine.
10) constraints ensuring that the preparation time taken by changing from piece i to piece j in group g is zero if i and j have the same type of assembly and colour, assuming a known value \( S_{ijg} \) if the previous conditions are not carried out.
11) constraints ensuring that when the number of equal pieces, continuously sequenced, exceeds the maximum batch dimension, then, the resulting setups are taken in account.

IV. SIMULATED ANNEALING ALGORITHM

The simulated annealing (SA) is a no deterministic search local technique used when trying to find solutions in combinatorial optimization problems, based on the analogy of the minimum state energy of a physical system compared to the minimum cost in a combinatorial optimization problem [26], [27].

Simulated annealing is an extension of the local search algorithm, based on a possible initial solution. For that initial solution is attached a certain cost, \( Z_{x} \), and from that initial solution on is reached a neighbouring solution with a cost represented by \( Z_{y} \). The difference between \( Z_{y} \) and \( Z_{x} \) is represented by \( \Delta Z_{xy} \).

If the cost decreases, i.e., if \( \Delta Z_{xy} = Z_{y} - Z_{x} < 0 \), the neighbour solution \( Z_{y} \) will then be accepted.
If the opposite happens, solution Zx will be maintained. This procedure will keep being repeated till new improvements stop emerging, what means that a local minimum was attained. The local search algorithms are very easy to apply, but, as can be understood, they may converge to a unique local minimum, what implies that significant deviations from the optimal solution can take place.

The simulated annealing turns up in order to improve the performance of this type of algorithm once it permits to overtake one local minimum with a certain probability, i.e., it is possible to accept one solution having a certain probability when \( \Delta Z_{\text{opt}} < Z_{\text{opt}} - Z_{\text{new}} > 0 \). In the SA the probability of the solution to be accepted is determined by the accepting function given by the expression \( \exp\left(\frac{-\Delta Z}{T}\right) \), where \( T \) is the control parameter which, by analogy, is equivalent to the factor temperature in the controlled metals cooling process. A random number is generated from a uniform distribution \((0, 1)\) and compared with the value of the accepting function, if not it won’t be accepted.

This function has the following effect: minor increases in the value of the accepting function have less probability to be accepted, once the larger the value of the temperature is, the bigger is the probability of an increase of the accepting function to be accepted. While the temperature is decreasing, less is the probability of a worst solution to be accepted. [27].

V. THE DEVELOPED HEURISTIC

In order to solve the optimization problem through the SA algorithm, the algorithm parameters have to be established [28], [29], namely:

- the initial and final values of the temperature which are the control parameters.
- the dimension of each temperature level, i.e., the number of iterations made for each value of the temperature. In the case being analysed the number of iterations in each level was considered proportional to the problem dimension and is given by \( 2N \), where \( N \) represents the number of batches to be scheduled.
- temperature tuning.

The utilization of the SA still requires the determination of: Initial solution; Neighbouring solutions; Cost function

A. Description of the Heuristic for the determination of the initial solution

This heuristic role for the purpose above mentioned is limited to respect the restrictions of the problem, namely in the particular problem studied by the authors, the compatibility between the width of the task to be performed and the required equipments involved. Starting from the task of bigger width, it selects all the compatible equipments and distributes the task to the less charged equipment.

The initial solution will have the following characteristics:
- the tasks having equal characteristics; assembly and colour for the same delivery date are joined in batches.
- Each machine has a waiting rank where a certain number of batches having the same known processing time are included.

- The batches have a preparation time depending on the process sequence in the processing machine.
- There is a “\( Z \)” cost expressed by units of time for the obtained sequencing, being this value the result of adding up the processing time, the preparation times and the time penalty when such is applicable.
- The transport unit is the batch.

As a result of the application of this heuristic, a possible solution is obtained and, then, from that point on, close solutions will be generated.

B. Heuristic for the determination of neighbouring solutions

Neighbouring solutions are obtained through random batch transferences and swaps. A neighbour solution can be generated by two ways

- Random transference with width restriction

The random transference when width restriction takes place consists in moving one random batch, positioned in the same queue, to other queue, only taking in account the width restriction.

- Random swaps with width restriction

In this case, random batches either belonging to the same queue, or in other ranks where only width restrictions exist, are chosen.

A comprehensive heuristic, as happens in our case, only based on random transfers or swaps, leads to a wider search space, which can take more time to converge to a minimum local.

In the present work, the above random criteria has been adopted, having been considered to be more important for the particular problem being analysed, in first place the quality of the solution and not the time involved for the execution.

C. Heuristic description for the cost “\( Z \)” evaluation

As explained before, the initial solution is a scheduling of grouped batches per machine, which implies that the system will be prepared to evaluate the time associated for the processing. In this case, the system, queue by queue, or, in other words, machine by machine, will evaluate the sum of the processing times associated to each batch.

After being done the exchange or transference, the system, queue after queue, will calculate the number of preparations of each existing type and proceed to the total calculation for all the preparations.

Considering all the times involved in production, the one where more calculation difficulties are met is the delay time calculated period by period. The system should consider the possibility of any advance or delay in a machine for a certain period of time is transferred to the following period. Such evaluation requires an additional computational effort, especially when big dimensions problems are faced.

Cost “\( Z \)” is then the result of adding up processing and setup times plus the penalties caused by eventually failed delivery dates.

VI. COMPUTATIONAL RESULTS

In order to validate the solutions, a heuristic for the determination of a lower bound has been used.
A good solution is the one resulting from the ratio production capacity versus processing and setups requirements.

If the processing time plus the setup time requirements exceeds the capacity, the lower limit should take in account a penalty time for the tardiness.

The penalty factor can take the form of a parameter.

Considering the amplitude of the problem that we are dealing with, is not an easy job to find an algorithm for computing the lower limit, covering with success all the situations.

An algorithm presenting good performances in environments where many groups of machines, many machines per group and many periods of time are involved was chosen; the authors think that is best adjusted algorithm to the practical case that they were dealing with.

A. Description of the carried out tests

Tests were made having in mind the structure of the identified problem. Ten scenarios having increasing complexity were created, the last one corresponding to the practical case being analyzed. For each of the situations, a significant number of tests have been done using different parameters. Generally speaking, the characteristics of the tests that have been done are shown in Table II.

In Table III is described the information contained in each column of Table II.

Scenario 10 concerns the practical studied case consisting on scheduling 5828 tasks distributed by 232 batches. These batches are distributed by 48 machines. The machines are grouped in 4 different categories according to its own characteristics. The planning horizon is on this case 10 weeks.

A certain amount of processing operation was made for the different scenarios in order to appreciate the best obtained performance. For this purpose the value δ has been changed in the range between 0, 0001 to 0, 1.

In each of the scenarios, four values for δ were tested, and for each value a significant number of processing operations.

Table IV shows the obtained results. In this table, Zm represents the best obtained performance, Z0 represents the initial solution. LB represents the lower band obtained through the applied heuristic, ((Zm-LB)/LB)*100 the percentage of improvement of the solution related to the lower band, and ((Z0-Zm)/Zm)*100 the percentage of improvement of the solution related to the initial solution.

Table III – Description of table II

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sce.</td>
<td>Identifies the constructed scenario</td>
</tr>
<tr>
<td>Mach. Gr.</td>
<td>Indicates the number of groups considered in the scenario</td>
</tr>
<tr>
<td>Mach. Nº</td>
<td>Indicates the total number of machines considered in the scenario</td>
</tr>
<tr>
<td>Nº Per.</td>
<td>Indicates the number of periods, the number of different delivery dates in the planning horizon. Within the same period all the tasks have the same delivery date.</td>
</tr>
<tr>
<td>Nº Ass.</td>
<td>Number of different assemblies considered in the scenario</td>
</tr>
<tr>
<td>Col.</td>
<td>Number of different colours considered in the scenario.</td>
</tr>
<tr>
<td>Nº Task</td>
<td>Indicates the number of tasks</td>
</tr>
<tr>
<td>Nº batch</td>
<td>Indicates the number of batches generated by the initial solution</td>
</tr>
</tbody>
</table>

Table IV – Obtained results

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Zm</th>
<th>Z0</th>
<th>LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12790</td>
<td>12790</td>
<td>12790</td>
</tr>
<tr>
<td>2</td>
<td>37910</td>
<td>49995</td>
<td>28040</td>
</tr>
<tr>
<td>3</td>
<td>79027</td>
<td>164908</td>
<td>65819</td>
</tr>
<tr>
<td>4</td>
<td>50739</td>
<td>50739</td>
<td>50739</td>
</tr>
<tr>
<td>5</td>
<td>62160</td>
<td>66910</td>
<td>57640</td>
</tr>
<tr>
<td>6</td>
<td>55345</td>
<td>74071</td>
<td>46495</td>
</tr>
<tr>
<td>7</td>
<td>61550</td>
<td>73585</td>
<td>58025</td>
</tr>
<tr>
<td>8</td>
<td>70445</td>
<td>80925</td>
<td>61950</td>
</tr>
<tr>
<td>9</td>
<td>248494</td>
<td>318428</td>
<td>231411</td>
</tr>
<tr>
<td>10</td>
<td>205318</td>
<td>227302</td>
<td>1946458</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>((Zm-LB)/LB)*100</th>
<th>((Z0-Zm)/Zm)*100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>2</td>
<td>35,20</td>
<td>31,88</td>
</tr>
<tr>
<td>3</td>
<td>20,07</td>
<td>108,67</td>
</tr>
<tr>
<td>4</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>5</td>
<td>7,84</td>
<td>7,64</td>
</tr>
<tr>
<td>6</td>
<td>19,03</td>
<td>33,84</td>
</tr>
<tr>
<td>7</td>
<td>6,07</td>
<td>19,55</td>
</tr>
<tr>
<td>8</td>
<td>13,71</td>
<td>14,88</td>
</tr>
<tr>
<td>9</td>
<td>7,38</td>
<td>28,14</td>
</tr>
<tr>
<td>10</td>
<td>5,48</td>
<td>10,71</td>
</tr>
</tbody>
</table>
fact should be underlined, the used methodology for dimensioning an sequencing of the batches is very close to the heuristic applied to generate the initial solution of the presented work.

VII. CONCLUSIONS

According to what has been said, considering the type of industry studied, the obtained results show a clear importance of the scheduling problem, when defining the planning of the production. As can be understood, it is not possible to extend the obtained results to every other type of industrial environment. Nevertheless, the now described study permits thinking in promising results in other different industrial environments.

The authors tried to demonstrate on this study, that, when well dominating the skilled tasks of planning and processing, it is possible to find something between the more conceptual approaches and the ones where more mathematics is involved. This will certainly induce to the development of new instruments, although with some expected limitations, can provide an important tool in what concerns one of the most vital managing information.

REFERENCES
