

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Lot Sizing and Scheduling in Parallel Uniform Machines – A Case Study

---

F. Charrua Santos, Francisco Brojo and Pedro M. Vilarinho

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/50975>

---

## 1. Introduction

This chapter presents the problem of scheduling tasks in uniform parallel machines with sequence-dependent setup times. The problem under analysis has a particular set of constraints, including equipment capacity, precedence tasks, lot sizing and task delivery plan. This kind of problem should be considered an operational planning problem and is found in different types of industries, namely, in the textile industry.

The main goal of this work is to minimise the total production time, including processing, setup and tardiness time.

The complexity of the studied model does not allow to locate the optimal solution. To solve this, the authors used a meta-heuristic known as the simulated annealing algorithm to obtain “nearly optimal” solutions for the problem.

The results obtained through the application of this particular heuristic show the importance of using a structured approach in production scheduling when compared with the results obtained from a reference plant where mainly “ad hoc” actions were normally taken.

The classical parallel machine scheduling problem considers  $p$  tasks with a fixed processing time that must be processed in  $k$  machines. This type of problem involves two kinds of decisions: i) which tasks are assigned to a specific machine and ii) tasks processing order (Mokotoff, 2001).

This problem becomes more complex as the number and characteristics of the available machines increases. The following levels of complexity can be considered: i) identical parallel machines; ii) uniform parallel machines and iii) unrelated parallel machines.

In the case of uniform parallel machines, each machine or group of machines has a different processing velocity. A processing time  $T_p$  for each task  $p$  in each machine  $k$ , described by  $T_{ppk}$ , can be considered.

The work described in this paper takes into consideration lot sizing because it is an important aspect of these kinds of problems. Scheduling and lot sizing are indivisible problems in a significant number of real-world operations, where the setup times are sequence dependent. The analysis of lot sizing usually results in a decrease in the setup numbers as a result of tasks grouping in batches (Potts and Wassenhove, 1992; Potts and Kovalyov, 2000). Often, as a consequence of their size and complexity, problems are not solvable by exact methods. This paper introduces a heuristic approach to solving the problem (Drexl and Kimms 1997; Allahverdi, Gupta and Aldowaisan, 1999).

In general, the problem can have different formulations that are dependent on the objective function and the algorithm used to solve the problem.

In next section of this chapter (Section 2) the bibliographic review is done, in (Section 3), the research problem characterisation is presented, followed by the problem statement and formulation (Section 4). In the next (Section 5), the simulated annealing algorithm is presented. The developed heuristic to solve the problem is shown in Section 6. In Section 7, is presented a set of computational results obtained with the developed heuristic. Finally, the conclusions and future areas of research are presented (Section 8).

## 2. Bibliographic review

The classical problem of scheduling parallel machines considered "n" tasks or assignments and "m" machines. Each work must be performed on a machine with a fixed processing time. Accordingly, the aim is to find the sequence of jobs to optimize determined performance indicators (Mokotoff (2001)). This type of problem involves two decisions: assigning work to a particular machine and its sequencing in the queue of this machine. The complexity of the problem increases exponentially with the number of machines considered.

The problems of parallel machines may also consider three cases (Gupta and Magnusson (2005) (Mokotoff (2001)):

- Identical parallel machines;
- Uniform parallel machines;
- Unrelated parallel machines.

In the case of parallel identical machines, the processing time of a task is independent of the machine where it is processed.

For uniform machines, each machine has a different processing speed. In practice we have to consider the processing time of job  $i$  on machine  $j$  ( $t_{ij}$ ). This time is equal to the processing requirements of the job  $i$  divided by the speed of the machine  $j$  ( $s_j$ ) ( $t_{ij} = t_i / s_j$ ). This feature leads to an increased complexity in solving the problem.

In unrelated machines sequencing, there is no particular relationship between the processing times in the different machines, i.e., there is no proportionality between the processing time of a task on a given machine and the time of the same task on another machine. It is necessary to define the processing time of each task in each machine.

In these cases the problem of sequencing seeks the best distribution of the tasks in the machines and the best result in each machine, with the aim of optimizing certain performance criteria.

Among all possible performance criteria, the most widely used is the production flow total time, since it provides the best balance between all factors considered. The problems in real environments are usually of this type. The solution of a problem may be different depending on the criteria considered and when the criteria do not conflict with each other, it is possible to find an optimal overall solution. Such problems were reviewed in depth in an article published by Nagar et al. (1995).

In general, the scheduling problems in parallel machines assume that:

- Each task or work is comprised of a single operation cannot be processed in more than one machine simultaneously;
- When a task is started, it must be completed before another task is started on the same machine;
- Interruption is not allowed, so each job once started must be completed;
- Processing time is independent of the sequence;
- Is permitted accumulation of work, i.e., the work can wait for a free machine;
- Is allowed downtime of machines;
- No machine can process more than one task at a time;
- During the sequenced period, stops due to equipment damage are not considered;
- The preparation time is zero for all jobs;
- The number of jobs is known and fixed;
- The number of machines is known and fixed;
- Processing times are known and fixed;
- All other specifications to define a particular problem must be known and fixed.

The characteristics presented herein refer to the classical approach of the problem, but there are other ways such as the proposed by Allahverdi et al. (1999), analysing the problems of scheduling parallel machines considering independent setup times and sequence dependent.

Also the optimization problems on parallel machines are identical, mostly, "NP-Hard." As is the case with the problems of single machine, there are two ways to deal with this problem:

- exact methods;
- heuristic approach.

Mokotoff (2001) published a review article on production scheduling in parallel machines. In this paper is made a thorough literature review on the work in different environments: identical parallel, uniform and unrelated machines.

The problem of sequencing with preparation time dependent of sequence in parallel identical machines, in order to minimize the time of preparation was made by Sumichrast and Baker (1987). Subsequently, França et al. (1996), again look into this matter.

Over the past three years there are many publications on identical parallel machines scheduling with the objective of minimizing production time. These publications present some variations on the objective that are:

- Minimization of production times with independent setup times (Tang and Luo (2006) and Haouari et al. (2006)). With preparation time dependent sequence refers to an article by Chen et al. (2006) who also consider batch processing and resizing. Gheunshie et al. (2004) analyze the sequencing batch using genetic algorithms. Allahverdi et al. (2008) presented a review paper on exact methods and heuristic approach to solving problems with positive time or cost of preparation in many industrial environments.
- Minimization of production times and meeting delivery dates. Shakhlevich and Strusevich (2005) analyze the sequencing of work while Jaehwan and Marc (2005) study the sequencing batch resorting to using several heuristics, no one considers the dependence of the sequence. In turn, using a tabu search algorithm, Shin and Leon (2004) analyze the production batch with setup times sequence dependent.
- Sequencing with replanning of work such as the work presented by Curry and Peters (2005).

There are also several works that address the sequencing problem in unrelated parallel machines environments. The objective focuses again around the minimization of the production time. Also in parallel study of unrelated machines are papers analyzing the sequence of separate operations such as the ones of Chen et al. (2006), while Rabadi et al. (2006) focus on the study of dependent sequences. Minimizing delays is also examined in environments of unrelated parallel machines. Chen et al. (2006) analyze the sequencing of jobs dependent on using for this purpose an algorithm of simulated recrystallization, while Kim et al. (2006) compared the use of simulated cooling algorithm with the genetic algorithm in the sequencing batch of time dependent preparation sequences. Meanwhile, Shim and Kim (2007) use techniques "Branch and Bound" to find the best sequence of jobs on machines not connected with preparation times independent of the sequence.

The problem of parallel machines may also appear associated with problems in continuous environments. Gupta et al. (1997) present a continuous production model consisting of two steps in which the first stage is comprised of multiple identical parallel machines. In this case the aim is to minimize the maximum processing time. Such problems are very common in practice, especially in the process industry, where in each step of the process there are multiple parallel machines, which may be identical or not.

Also the concept of the machine can be broadened if we consider parallel assembly lines. Considering the line as a black box, the sequence of batches in these lines can be considered as a problem of parallel sequencing machines. The problem of sequencing on automatic parallel production lines and considering the problem of scale batch was analyzed by Meyr (2002).

Table 1 presents a summary of works published in the literature that were considered relevant when compared with the work performed by the authors.

Ref.	Year	Parallel Machine Problem	Objective Function	Algorithm
Sivrikaya-Serifoglu, and Ulusoy	1999	Identical	Minimise Earliness (ME) Tardiness (MT) Penalties (MP)	Genetic algorithm(GA)
Allahverdi, Gupta and Aldowaisan	1999	Generalised	Review	
Radhakrishnan and Ventura	2000	Identical	ME/MT/MP	Simulated Annealing (SA)
Mokotoff	2001	Generalised	Review	
Anagnostopoulos and Rabadi	2002	Unrelated	Minimise Makespan(MM)	SA
Eom, shin, Kwun, Shim, and Kim	2002	Generalised	MT	Three-phase Heuristic, Tabu Search (TS)
Ref.	Year	Parallel Machine Problem	Objective Function	Algorithm
Meyr	2002	Generalised	Simultaneous Lot Sizing and Scheduling	
Liao and Lin	2003	Uniform	MM	Optimal Lexicographic Search
Jaehwan and Marc	2005	Generalised	Rescheduling to Increasing	Branch-and-Price
Tang and Luo	2006	Generalised	MM	Iterated Local Search (ILS)
Wen-Chiunge, Chin-Chia and Chen	2006	Identical	MM	SA
Chen	2006	Unrelated	(MT)	Tabu Lists
Haouari, Gharbi and Jemmali	2006	Identical	MM	Lower Bounding Strategies
Kim, Choi and Lee	2006	Identical	MT	TB, and SA
Lee, Wu and Chen	2006	Identical	MM	SA
Nessah, Yalaoui and Chu	2006	Identical	MM	Branch and Bound (BB)
Rabadi, Moraga and Al-Salem	2006	Single machine	E/T	Heuristic Tailored and SA
Shim and Kim	2007	Unrelated	MT	BB
Anghinolfi and. Paolucci	2007	Generalised	MT	Hybrid Metaheuristic Approach
Logendran, McDonell and Smucker	2007	Unrelated	MT	Six TS
Oh and Kim	2008	Identical	MT	BB

Ref.	Year	Parallel Machine Problem	Objective Function	Algorithm
Allahverdi, Cheng, Mikhail and Kovalyov	2008	Generalised	Review	
Raja, Selladurai, Saravanan and. Arumugam	2008	uniform	E/T	SA and Fuzzy Logic
Rocha, Ravetti and Mateus	2008	Unrelated	Due Dates and Weighted Jobs	Exact Methods (EM)
Su	2009	Identical	MM	
Eren, and Guner	2009	Identical	MM	Bi-criteria Scheduling with a Learning Effect
Mellouli, Sadfi, and Chu	2009	Identical	MM	Three EM
Chaudhry and Drake	2009	Identical	MT	GA
Wang and Cheng	2009	Identical	Minimise the Weighted Sum of the Last Arrival Time of the Jobs	Heuristics
Chen, and Chen	2009	Unrelated	MT	Hybrid Metaheuristics
Chen	2009	Unrelated	MT	SA
Kang and Shin	2010	Identical	MT and Reworks	Dispatching Algorithm
Ref.	Year	Parallel Machine Problem	Objective Function	Algorithm
Huo and Leung	2010	Identical	MM	Algorithm With a Worst-case Bound
Huang, Cai, and Zhang	2010	Identical	MM	Hybrid GA
Chuang, Liao, and Chao	2010	Unrelated	MM	BB
Unlu and Mason	2010	Identical	MM	Different Mixed Integer Programming
Gacias, Artigues and Lopez,	2010	Identical	MM	BB and Climbing Discrepancy Search
Behnamian; Zandieh and Ghomi,	2010	Identical	MM/MT/ME	GA
Fanjul-Peyro and Ruiz,	2010	Unrelated	MM	ILS

Ref.	Year	Parallel Machine Problem	Objective Function	Algorithm
Okolowski and Gawiejnowicz	2010	Identical	MM	EM
Moradi and Zandieh	2010	Identical	MM and System Unavailability for the Maintenance	GA
Lin, Lin and Zhou	2010	Identical	MM	Approximation Algorithm
Lin, Pfund and Fowler	2011	Unrelated	MM/MT	Review
Jouglet and Savourey	2011	Unrelated	MT	Dominance Rules and Filtering Methods
Edis and Ozkarahan	2011	Identical	MM	Three Optimisation Models
Hsu, Kuo Yang	2011	Unrelated	MM	Polynomial Time Solution

**Table 1.** Problems referred to in the literature.

The analysed problem of production lots sizing and scheduling is intended to minimize, for a given planning horizon, the setup and processing time of a set of orders with due dates set and the delays incurred in the due date of these orders. The problem occurs in an environment where the equipment for the production of such packages are configured as uniform parallel machines and setup times that are dependent on the sequence of orders generated by manufacturing orders. The equipment used induces width restrictions.

The problems of minimizing the production time for a sequence of jobs minimizing delays and work with common due dates have been solved as a problem of route salesman (TSP) by Baker (1974) and Picard and Queyranne (1978). These problems were further extended to include time dependent setup sequence between batches, and some of the preparation times are void (Monma and Potts 1989).

As it turned out, since the review work carried out by Mokotoff (2001) has been extensive published literature on the sequencing of parallel machines. However, the literature on uniform parallel machines is scarce ((Blazewicz et al. (2000), Liao and Lin (2003), Jiang et al. (2005) and Raja et al. (2008)).

Given the research done, the problem of sequencing and lot sizing in uniform parallel machines, with penalties for failure to meet delivery deadlines, setup times depend on the sequence and width restrictions and size of batch, identified during the study carried out in an industrial environment, has never been analyzed, it is therefore important to carry out the formulation and development of a method for its resolution.

In the objective function we choose a model that reflects the performance indicators in use in the company that accompanied the study, including the objective function to minimize the processing time, preparation (Setup) and delays in deliveries.

The high complexity of the proposed model, to the problem under analysis, prevents its resolution by use of the optimizing methods, even in cases where problems are of average size. Given this difficulty, we developed a heuristic for finding sub-optimal solutions to the problem, which is based on simulated annealing algorithm. Heuristics based on this algorithm have been used to solve different combinatorial optimization problems whose resolution by use of optimizing methods is not possible.

Simulated Annealing is a technique of non-deterministic local search, which uses an analogy between the minimum energy state of a physical system with minimal cost in a combinatorial optimization problem (Metropolis et al. (1953), Kirkpatrick et al. (1983), Radhakrishnan and Ventura (2000) and Wen-Chiunge et al. (2006)).

At liquid state, a given material has a random atomic ordering, it can be considered chaotic. As it cools, converges to a state of minimum energy structure with a rigidly defined and ordered. If cooling is done quickly, in most materials is produced a deformation of the microstructure which can lead to cracking. The different states which a material during the controlled cooling goes through correspond by analogy to the various possible solutions of an optimization problem, with the system energy corresponding to the function to be optimized.

The simulated annealing has been widely used in recent years in all kinds of problems. A search for the last 5 years reporting over 500 scientific articles concerning the use of simulated annealing algorithm optimization problems.

### 3. Problem characterisation

The analysis made to the production process of three of the largest producers of fabrics of Beira Interior, one of them the largest in Europe in the production of wool fabrics, shows that the planning methodology does not differ greatly from company to company. It is possible to observe significant differences at computerization level, but the planning philosophy is the same.

The companies analyzed have the following common features:

- The flow between sections is typical Batch Shop. Within each section you can find environments challenging to frame in a specific classification, being the most frequent a combination of a continuous environment with steps and parallel machines;
- The work is done mainly to order, having only, during some phases of the production process, work made for stock and that just to reduce order response times;
- Each section is a supplier or customer of the section that follows or precedes, respectively. In this perspective, planning and production control are highly complex;
- The production is characterized by strong seasonality and the fashion factor. These facts, coupled with the growing trend of product differentiation, but with higher added value, translates into unpredictability of sales increase;
- It is clear a decrease in the size of orders, with a visible impact on the size of the lots;
- The problem of sequencing and lot sizing is placed in all sections of the companies.

In Beira Interior (Portugal), the more credited tissue producing companies, wager on verticality. This option brings greater difficulties in terms of planning.

In the analyzed companies and in line with the literature, there are three identifiable levels of planning:

- first, of administration responsibility, typically includes strategic decisions;
- second, usually designated central, is responsibility of the company senior management, including tactical decisions, namely, the determination of supplies requirements and management of abilities. Sometimes, at this level of planning and at some sections of the company, are also made decisions regarding lot size and production scheduling, actions that are normally field of operational planning;
- third planning level, arises at the sequencing level and is based on some simple heuristics rules and occurs in certain sections, being the responsibility of the section responsible. At this level, the planning may also include some decisions issued by central planning, related to the distribution of production orders by the equipment.

The problem of production sequencing cannot be analyzed in an integrated way to all sections of the companies. Such problem would be highly complex and impossible to solve. The question that arises is: which of the sequencing problems that arise in the different sections may be more relevant to a greater efficiency of the company as a whole? To try to answer this question, the planning process will be analyzed in some detail.

In practice, it is observed that the orders breakdown and advances are not made from the orders backlog, but from the programming of these at weaving. If, in terms of quantities to produce, the difference is not significant, in terms of delivery dates of the sections that are upstream is important because the reference will be the weaving needs and not the delivery date to the client. This reasoning is always implicit in the decisions that are taken, i.e., underlying the planning process of the textile industry there is always the question of maximizing the weaving production.

The problem of weaving lot sizing lays on the assumption: the larger the lot the greater the efficiency of the company. The reasons for this assumption are the following:

- setup times associated with the change of batch. For larger batches are fewer setups and the weaving efficiency will be higher;
- since the components needs (strings) are generated from the weaving programming, the larger the batch, the greater the size of the lots in sections upstream (dyeing and spinning).

Production planning in the analyzed companies is a highly centralized planning system. This fact is understood by the need to integrate information in a complex environment dependent of demand.

In spinning and dyeing, the sequencing is responsibility of the sections responsible, with only a delivery date associated with each manufacturing order that the person responsible has to respect.

In weaving, the interference of the central planning in scheduling operations is significant and justified by the disaggregation model presented above.

On the other hand, there are other reasons why weaving deserves special attention:

- the productive capacity of a company is given by the ability to weave, even when including outsourcing;
- marketed product (fabric), even not finished, is physically produced in this section, which induces two clearly distinct types of control. Upstream weaving we have a pull process, where the weaving facing its needs "pulls" the production, downstream of weaving, because the relation order/product is very visible, we have a push system, where the delivery pressure is exerted by both the commercial and production sectors.

Given the foregoing, it is concluded that the problem of scaling and sequencing of production batches arises in all sections of the companies. However, by the observations made, the problem that seems to have a greater impact on productive efficiency is the weaving batch sizing and sequencing.

It was concluded that a set of parts are a lot, and that within the same lot, the batch preparation time between parts is zero. However, there is a limit to the number of parts that can form a batch. This limit is given by the technical limitations of product and equipment.

Once grouped into batches, the pieces can lead to two types of setup, depending on their technical characteristics. So, considering the sequencing of parts, one can consider the following setup times:

- zero, when the parts are equal and belong to the same lot,
- 8 hours, when the parts are equal but are from different batches,
- 24 hours, when the parts are different.

For the analyzed problem, it was found that the demand is known in advance, being this way deterministic and stable for a planning horizon of 10 weeks. In each week, there is only a delivery date, corresponding to the last day of the week. Thus, we are facing a problem where the planning horizon is finite, with multiple periods of time. The periods are constant, which means that the difference between two consecutive delivery dates are equal.

In order to characterize the demand, was made a survey of confirmed orders for a period of 10 weeks, in one of the companies, concluding that there were 87 different products totaling 5828 pieces to schedule in 48 machines.

The processing time per piece is known.

From data analysis of the problem and characteristics of the industry producing tissues, it was concluded that a factor of competitiveness for the sector is the fulfillment of deadlines, so this feature should be included in the formulation of the problem. In order to be compared with current practice in the company, was chosen an objective function that reflected the heuristics applied in the company, which took into account the production time, the setup time and a tardiness penalty.

Beyond the characteristics presented above, it was verified that there are still a number of assumptions that, in terms of problem formulation, is important to consider, including:

- a machine can only process one piece at a time;
- one piece can be processed in only one machine;
- the production of one piece on a machine cannot be interrupted;
- a piece can move from the period;
- initial characteristics of the machines are known.

As previously stated, the problem under research was inspired by the textile industry, weaving in particular, where  $n$  jobs are sequenced on  $k$  machines. The machines are grouped into  $G$  groups according to their processing velocity, turning this into a uniform parallel machine problem. Not all tasks can be processed on all machines because the plant has looms with different widths. This technical aspect confers to the problem an increased level of complexity, and it is a distinguishing factor when compared with the other situations reported in the literature.

The tasks have sequence-dependent setup times. Same type of tasks sequenced in the same machine are joined in batches, so the setup time between same type of tasks (i.e., for tasks belonging to the same batch) is zero. Nevertheless, when the amount of same type of tasks reaches an established lot size, a setup should then take place. The lot size varies according to the type of tasks. This is a major achievement compared with what was found in the literature. For same type of tasks with a continuous sequence on the same machine, two setup times take place: a) the setup time will be zero if the limit lot size has not been reached; and b) it will have a certain value (the setup time) if another situation occurs.

The demand is deterministic for the planning horizon with a due date associated with each task. On the planning horizon, we can have  $n$  due dates.

#### 4. Problem statement and formulation

The generalised version of the problem can be described as the problem of scheduling tasks in uniform parallel machines, with due dates and sequence-dependent setup times.

The problem under analysis has the following characteristics:

- i. The throughput is stable during the period of production;
- ii. The demand is known in the planning horizon;
- iii. The production can be adequately represented by a limited set of tasks ( $p=1, \dots, P$ );
- iv. The production is performed in a set of machines joined together according to their characteristics in  $G$  groups ( $g=1, \dots, G$ ), each one having  $k$  machines ( $k=1, \dots, K$ );
- v. For each task  $p$ , a due date  $d_p$  exists;
- vi. The task can be delivered after the due date but incurs tardiness. This tardiness is penalised by a factor  $\rho$
- vii. Each task is processed in a single machine of a defined group;
- viii. Each task  $p$  can be processed in machines belonging to any group, whereas the required characteristics need verifying, considering  $T_{ppg}$ , the processing time of task  $p$  on one of the machines belonging to group  $g$  ( $p=1, \dots, P$  and  $g=1, \dots, G$ );

- ix. The setup time for each machine that processes task  $j$ , after having processed task  $i$ , is known and represented by  $s_{ij}$  and is independent of the machine in which the task has been processed but always considers the characteristics of the sequence of the tasks;
- x. The setup time of the machine that produces task  $i$  in the beginning of the job sequence on a machine is known and represented by  $s_{0i}$ ;
- xi.  $L_{max}$  represents the maximum number of tasks than can be joined and processed together continuously without requesting setup time;
- xii. There is no need to consider a setup time for tasks with the same technical characteristics. The amount of different assembly combinations is  $A$ , whereas  $N_a$  ( $a=1, \dots, A$ ) is the set of tasks that have the same characteristics, assembly and colour, with  $a$  as the reference.
- xiii.  $M$  is a large positive number;

$$xiv. \beta_{ig} = \left\{ \begin{array}{l} 1 \text{ if task } i \text{ can be} \\ \text{processed in group } g \\ 0 \text{ otherwise} \end{array} \right\}$$

Before providing a mathematical formulation, the following variables are defined:

- $C_p$  date of conclusion of task  $p$
- $t_p$  tardiness of task  $p$
- $s'_{ij}$  setup time for task  $j$  when this occurs after task  $i$

$$x_{ijk} = \left\{ \begin{array}{l} 1 \text{ if task } i \text{ is followed by } j \text{ in} \\ \text{machine } k \text{ of group } g \\ 0 \text{ otherwise} \end{array} \right\}$$

$$y_{ikg} = \left\{ \begin{array}{l} 1 \text{ if } i \text{ is assigned to a machine in group } g \\ 0 \text{ otherwise} \end{array} \right\}$$

$$z_{ij} = \left\{ \begin{array}{l} 1 \text{ the tasks } i \text{ and } j \text{ are processed in} \\ \text{the same machine } k \text{ of group } g \\ 0 \text{ otherwise} \end{array} \right\}$$

The problem can then be modelled as one of mixed integer programming by the following expressions, which are explained next.

$$\min \sum_i \sum_k \sum_g y_{ikg} \cdot \rho_{ig} + \sum_i \sum_j \sum_k \sum_g x_{ijk} + \sum_{i,j \in N_a; a=1}^A \sum_k \sum_g S_{ijg} \cdot R_{akg} + \rho \cdot \sum_i t_i \tag{1}$$

$$\sum_j x_{0jkg} = 1, k=1, \dots, M_g; g = 1, \dots, G \tag{2}$$

$$c_j + M.(1 - x_{0jkg}) \geq s_{0jg} + p_{jg}, j = 1, \dots, P; k = 1, \dots, M_g; g = 1, \dots, G \quad (3)$$

$$\sum_{i; i \neq j} x_{ijk} = y_{jk}, j = 1, \dots, P; k = 1, \dots, M_g; g = 1, \dots, G \quad (4)$$

$$\sum_{j; j \neq i} x_{ijk} = y_{ik}, i = 1, \dots, P; k = 1, \dots, M_g; g = 1, \dots, G \quad (5)$$

$$c_j + M.(1 - x_{ijk}) \geq c_i + p_{jg}, i, j = 1, \dots, P; k = 1, \dots, M_g; g = 1, \dots, G \quad (6)$$

$$c_i - d_i \leq t_i, i = 1, \dots, P \quad (7)$$

$$\sum_k \sum_g y_{ikg} * \beta_{ikg} = 1, i = 1, \dots, P \quad (8)$$

$$y_{ikg} + y_{jkg} - 1 \leq z_{ij}, i, j = 1, \dots, P; k = 1, \dots, M; g = 1, \dots, G \quad (9)$$

$$s'_{ij} = s_{ij} \cdot z_{ij}, \forall i, j \in N_a; a = 1, \dots, A \quad (10)$$

$$\left[ \frac{\sum_{\forall (i,j) \in N_a} x_{ijk}}{L_{a \max}} \right] \leq R_{akg}, \forall i, j \in N_a; a = 1, \dots, A \quad (11)$$

In the objective function,

(1) is the total production time and is the sum of the following four items: i) tardiness time, ii) setup time among different tasks, iii) setup time among same type of tasks and iv) processing time.

- The tardiness time is given by the sum of all the delays that 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 of the task processing times included in the planning analysis.

The model constraints can be interpreted as follows:

- 2) Constraints ensuring that only one initial setup can take place in each machine  $k$  belonging to group  $g$ ;
- 3) Constraints ensuring that the conclusion time of task  $j$ , when processed at the beginning of the job sequence on a machine  $k$ , belongs to group  $g$  and is equal to the initial setup time of task  $j$  plus the processing time of task  $j$  in machine  $g$ .
- 4) Constraints ensuring that if the setup time for task  $j$  takes place in machine  $k$  of group  $g$ , then  $j$  is either preceded by another task  $i$  or is the first task in the queue and will be processed in machine  $k$  of group  $g$ .

- 5) Constraints ensuring that if the setup time for task  $j$  takes place in machine  $k$  of group  $g$ , then  $j$  is either preceded by another task  $i$  or is the last task in the queue and will be processed in machine  $k$  of group  $g$ .
- 6) Constraints ensuring that the processing of each task once started cannot be interrupted;
- 7) Constraints ensuring that the tardiness time of task  $i$  is given by the difference between the conclusion date and the due date. From this problem, it can be concluded that  $t_i$  is a positive number and tardiness time only occurs when  $c_i$  is larger than  $d_i$ ;
- 8) Constraints ensuring that each task is manufactured only once and in a compatible machine;
- 9) Constraints ensuring that if one task  $i$  is processed in machine  $k$  of group  $g$ , and another task  $j$  is processed in machine  $k$  of the same group  $g$ , then both tasks are processed in the same machine.
- 10) Constraints ensuring that the setup time taken by changing from task  $i$  to task  $j$  in group  $g$  is zero if  $i$  and  $j$  have the same technical characteristics, assembly and colour. Otherwise, it assumes a known value  $S_{ij}$ ;
- 11) Constraints ensuring that when the number of same type of tasks, continuously sequenced, exceeds the maximum lot size, then the resulting setup time is taken into account.

Restrictions 4) and 5) ensure that tasks  $p$  will be processed by the  $k$  machines of  $g$  groups, which means that all the tasks will be processed. Simultaneously, these conditions ensure that each machine only processes a single task at a time, considering that the number of setups plus the initial setup gives the total number of processed tasks.

## 5. Simulated annealing algorithm

According Radhakrishna and Ventura (2000) the simulated annealing (SA) algorithm is a stochastic approach that endeavours to overcome local optimality by accepting bad solutions with a definite probability. It has been used to solve many combinatorial problems. SA is a technique of searching for good solutions in a configuration space which draws an analogy between minimum cost configuration in a combinatorial optimization problem (Metropolis et al (1953); Kirkpatrick et al (1983) Liao and Lin (2003); Lee, Wu and Chen(2006); Logendran et al (2007), and Kimet et al, (2006)).

The motivation for the SA algorithm comes from an analogy between the physical annealing of solids and combinatorial optimization problems. Physical annealing refers to the process of finding low-energy states of a solid by melting the substance initially and then lowering the temperature slowly, to temperatures close to the freezing point. An example would be to produce a crystal from the molten substance. In a liquid, the particles are arranged randomly, but the ground state of the solid, which corresponds to the minimum energy configuration, will have a particular structure, such as seen in a crystal. If the cooling is not

performed slowly, the resulting solid will not attain the ground state but will be frozen into a meta-stable, locally optimal structure, e.g. glass or crystal with several defects in the structure. In the analogy, the different states of the substance correspond to the different feasible solutions of the combinatorial optimization problem, and the energy of the system corresponds to the function to be minimized. (Radhakrishna and Ventura, 2000).

The basic idea of the utilized algorithm is as follows. An initial feasible solution to the problem is generated using Heuristic, for which, a starting value of the temperature parameter, ( $T_0$ ) is estimated. A neighbouring solution is then randomly generated, which is accepted or rejected using the SA methodology, e.g. if the energy for the new solution is smaller, the solution is accepted, if the energy for the new solution is higher, then Metropolis criterion is employed for acceptance. This process is repeated until the algorithm converges to the final temperature value ( $T_f$ ). SA ensures that the solution found increase the possibility to find the global minimum.

For the present case, the energy is substituted by the cost. A specific cost,  $Z_x$ , matches the initial solution. From this initial solution, a neighbouring solution is randomly reached, with a cost represented by  $Z_y$ . The difference between  $Z_y$  and  $Z_x$  is represented by  $\Delta Z_{yx}$ . If the cost decreases (i.e., if  $\Delta Z_{yx} = Z_y - Z_x < 0$ ), then the neighbouring solution  $Z_y$  will be accepted. If the opposite occurs, solution  $Z_x$  will be retained if respecting the Metropolis criterion. This procedure will be executed as long as a local minimum is not reached. The local search algorithms are very easy to apply, but they may converge to a unique local minimum. As a result, a considerable deviation from the optimal solution can be introduced.

The SA algorithm improves the overall performance of this type of method because it makes it possible to overtake one local minimum with a given probability, i.e., there is a probability of accepting a worse solution, expressed by  $\Delta Z_{yx} = Z_y - Z_x > 0$ .

The pseudocode developed by Madhavan (1993), is shown below:

Begin

Select an initial solution (where  $\pi$  is the solution space)

Select an initial temperature,  $T=T_0$ ;

Set temperature counter,  $u=0$

Repeat

    Set interaction counter,  $n=0$

    Repeat

        Generate solution,  $y \in \Pi$ , a neighbour of  $x$ ;

            Calculate  $\Delta Z_{yx} = Z_y - Z_x$

            If  $\Delta Z_{yx} < 0$  then  $x=y$

```

Else
if > random number from  $U(0,1)$ , then  $x=y$ ;
     $n=n+1$ 
Until ( $n=R$ );
     $u=u+1$ ;
     $T=T_u$ ;
Until (stopping criterion ( $T_u \leq T_f$ ) is satisfied);
End

```

## 6. The developed heuristic

To solve the optimisation problem using the SA algorithm, we must establish the parameters, namely:

- the initial and final values of the temperatures given in Eq. (12) and (13), which are the control parameters.

$$T_0 = \Delta Z_{\min} + (\Delta Z_{\max} - \Delta Z_{\min}) / 10 \quad (12)$$

$$T_f = \Delta Z_{\min} \quad (13)$$

- the dimension of each temperature stage, i.e., the number of iterations executed for each value of the temperature. In the case being analysed, the number of iterations in each stage was considered proportional to the problem dimension and is given by  $2N$ , where  $N$  represents the number of batches to be scheduled.
- the cooling rate:

$$T_{u+1} = \frac{T_u}{1 + \beta T_u}, \quad (14)$$

where

$$\beta = \frac{(T_0 + T_f)}{p T_0 T_f} \quad (15)$$

and

$$p = \frac{50\delta N(N-1)}{2}, \quad (16)$$

where  $\delta$  controls the cooling rate.

The utilisation of the SA algorithm still requires the determination of the following:

- the initial solution;
- neighbouring solutions;
- the cost function.

### 6.1. Initial solution heuristic

The restrictions on the problem must be observed by the heuristic used to find its initial solution, namely, the compatibility between the width of the task to be performed and the required equipment. The task with a larger width is assigned to the less busy compatible equipment. The initial solution will have the following characteristics:

- i. All the tasks with the same characteristics and due date are joined in the same batch.
- ii. Each machine has a queue in which the number of batches and the processing time are known.
- iii. The setup time required by the batch depends on its sequence inside the queue.
- iv. Cost  $Z$  is expressed in time units. This value is obtained by adding the processing time, the setup times and the time penalty resulting from tardiness.
- v. The batch is considered the carry unit.

From the application of this heuristic, a possible solution is obtained and is used to generate the neighbouring solutions.

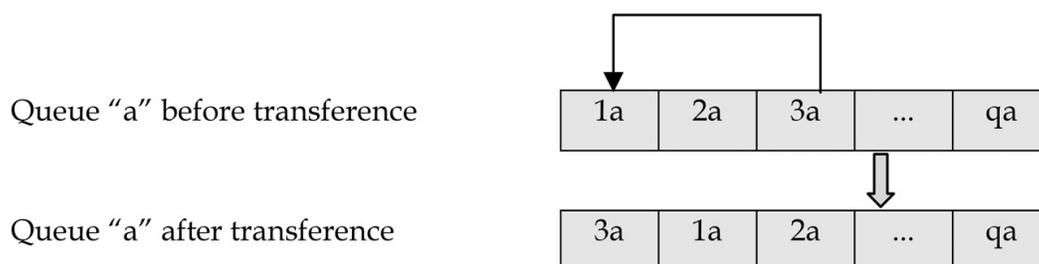
### 6.2. Heuristic for the determination of neighbouring solutions

Neighbour solutions are obtained through random batch transferences and swaps.

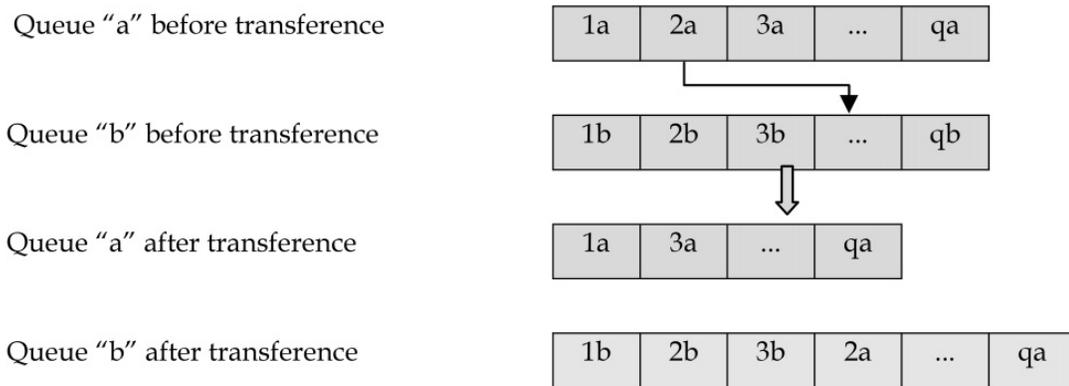
A neighbour solution can be generated by two different methods:

- Random transference in which the width is a restriction

This method consists of changing the batch position inside the queue (see Figure 1) or transferring it to another queue (see Figure 2). This task only takes into account the width restriction.



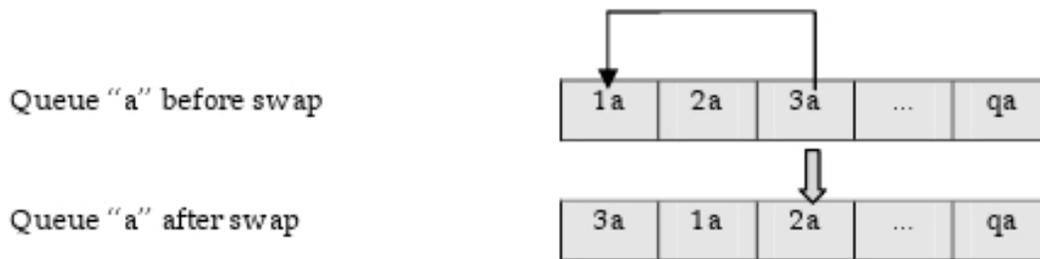
**Figure 1.** Transference of batches in the same queue



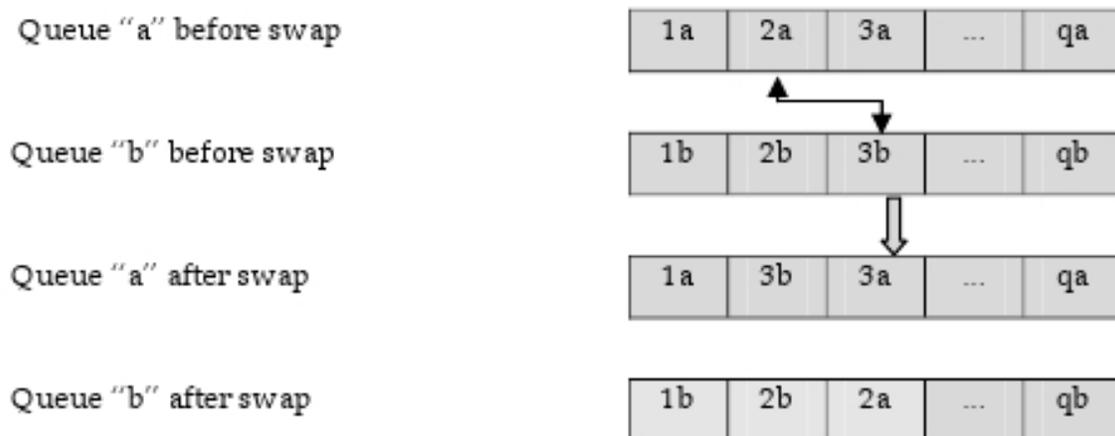
**Figure 2.** Batch transference between queues of compatibles machines.

- Random swaps in which the width is a restriction

In this situation, in the same queue, two batches are allowed to swap positions, as represented in figure 3. If the two batches are from different queues, they can also be exchanged, as shown in figure 4.



**Figure 3.** Swapping batches in the same queue



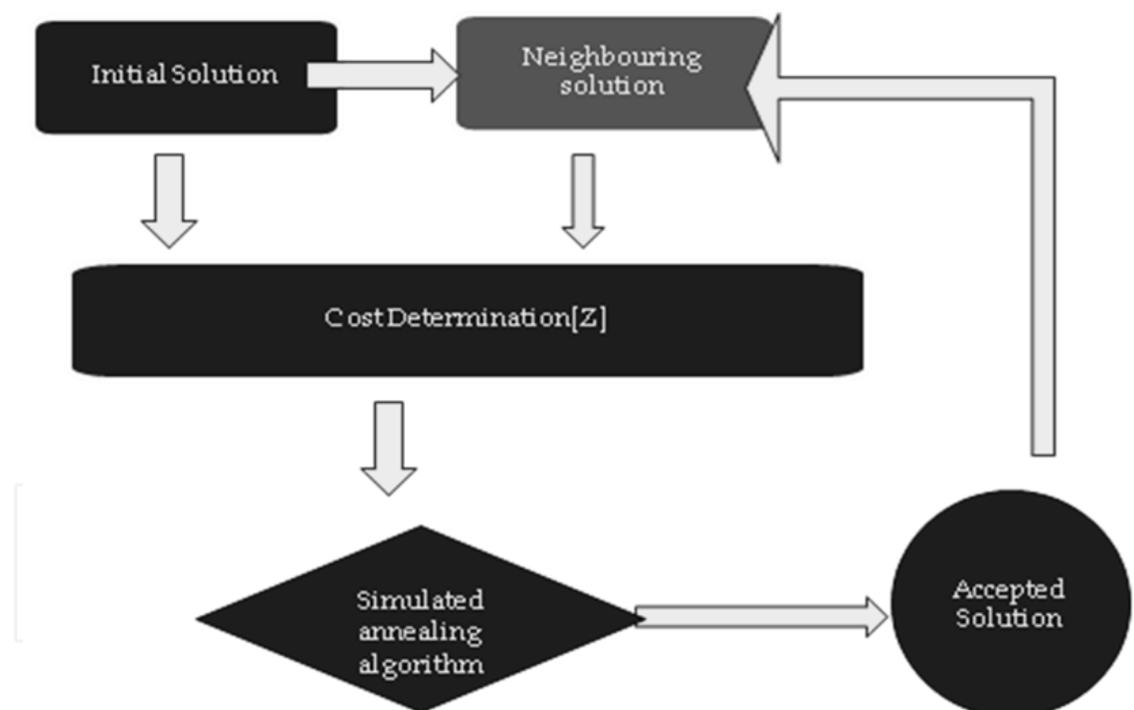
**Figure 4.** Swapping batches between queues of compatibles machines.

The heuristic for the determination of neighbouring solutions, as presented in this paper, which is based only on random transfers or swapping, leads to a wider search space.

Although the proposed method takes more time to converge to a minimum local, it has the advantage of reaching improved solutions. This option has proved to be a good choice for the particular problem under analysis, where the quality of the solution was considered much more important than the computational time.

As a result of the swap or transference described in this paragraph results a new possible solution to the problem. This new solution is acceptable if the cost  $z$  is less than or greater in some cases. The cost  $Z$  referred to in paragraph 6.3 reflects the objective function. This is, the solution resulting from the swap is evaluated according to the setup and processing time and delays it generates. Thus a relation between the objective function and the generation process of surrounding solutions is established.

As shown in the figure 5 is generated the initial solution, applying the objective function to this solution results in a certain cost  $Z$ . From the initial solution is generated a neighbouring solution as described in section 6.2. The objective function is applied to the new solution and tested in the simulated annealing algorithm and thus accepted or rejected. The solution obtained will give rise to another solution neighbour and so on until the process is interrupted according to the criteria described.



**Figure 5.** Schematic representation of performed algorithm

### 6.3. Description of the heuristic for evaluating the “Z” cost

As previously explained, the initial solution is a sequence of batches ordered by machine. Therefore, it is possible to find the time required to process the batches. In this case, the algorithm, queue by queue, or, in other words, machine by machine, will evaluate the sum of the processing times associated with each batch.

After task exchange, or transference, the heuristic will compute queue by queue the number and type of setups and then proceed to the total calculation for all of the setups.

Of all of the production times, the one that presents more calculation difficulties is the tardiness time, period by period. The system should take into account the possibility of an advance or delay for a machine that lasts for a certain period of time and should transfer the work to the following period. Such evaluation requires an additional computational effort, especially when large dimension problems are faced.

Cost “Z” is then the result of adding up processing and setup times plus the penalties that arise from failing to meet the due dates.

## 7. Computational results

To validate the solutions, a heuristic for the determination of a lower bound has been used.

### 7.1. Determination of “lower bound”

A good solution is the one resulting from the ratio of the production capacity versus the processing and setup requirements.

If the processing time plus the setup time requirements exceed the capacity, the lower limit should take into account the penalty time for the tardiness.

The penalty factor can take the form of a parameter.

Considering the magnitude of the problem that we are dealing with, it is not an easy job to find an algorithm for computing the lower limit that can successfully address every situation.

An algorithm that demonstrated good performance in environments with many groups of machines, many machines per group and many periods of time was chosen; the authors think that it is the best adjusted algorithm for the practical case that they were studying.

### 7.2. Description of the tests that were conducted

Tests were designed, keeping in mind the structure of the identified problem. Ten scenarios with increasing complexity were created, with the last one corresponding to the practical case being analysed. For each of the scenarios, a significant number of tests have been done using different parameters.

The characteristics of the performed tests are shown in Table 2.

In Table 3, the information contained in each column of Table 2 is described.

From the analysis of the results, the authors concluded that the algorithm performance was strongly related to the structure and the dimensions of the problem. This was expected once the evaluation criteria (convergence to the lower bound) were developed because of the previous experience of the skilled planner with large dimension problems.

In fact, the lower bound reflects the value that would be considered best by the planner for a problem of such size. It should be emphasized that in the analysed industrial environments, the methodology used for lot sizing and scheduling is very close to the heuristic applied to generate the initial solution in the presented work.

Scenario	Machine Group	Machine Number	Period	Assembly	Colour	Task Number	Batch Number
1	1	1	3	1	3	50	3
2	1	1	3	2	3	105	9
3	1	1	5	2	5	292	20
4	2	2	3	2	6	189	6
5	2	2	3	2	6	235	12
6	2	2	4	4	10	242	32
7	2	4	2	4	7	295	13
8	4	4	2	4	7	295	13
9	4	8	4	6	14	1148	50
10	4	48	10	39	87	5828	232

**Table 2.** Performed tests

Column	Description.
Scenario	Identifies the constructed scenario
Machine Group	Indicates the number of groups considered in the scenario.
Machine Number	Indicates the total number of machines considered in the scenario.
Period	Indicates the number of periods—the number of different due dates in the planning horizon. Within the same period, all the tasks have the same due date.
Assembly	Number of different assemblies considered in the scenario.
Colour	Number of different colours considered in the scenario.
Task Number	Indicates the number of tasks.
Batch Number	Indicates the number of batches generated by the initial solution.

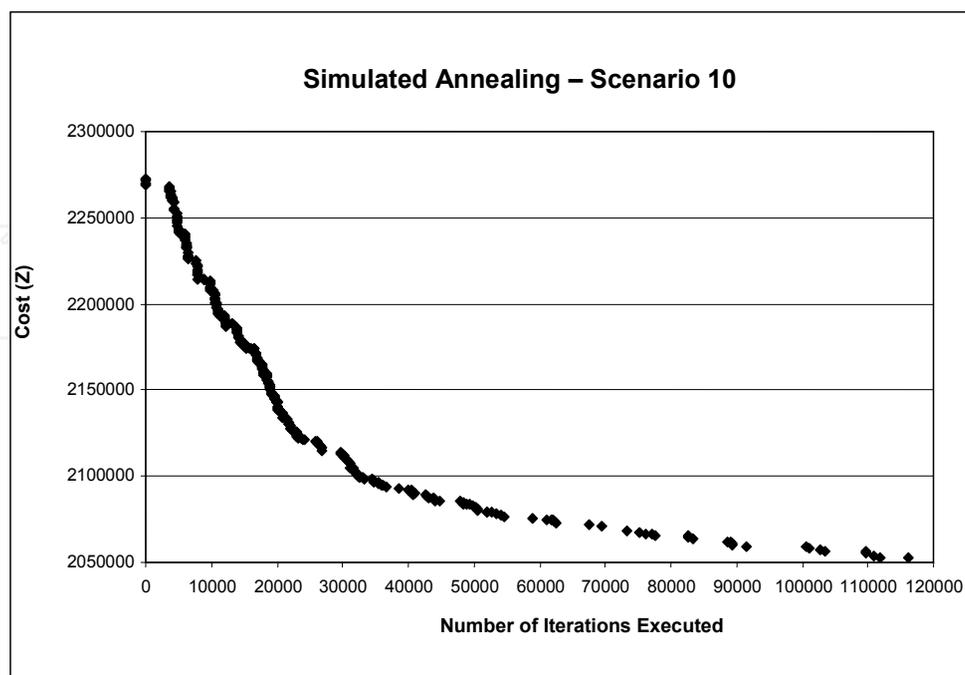
**Table 3.** Description of Table 2

For the smaller problems, scenario 1 and 4 of Table 4, the best solution was obtained by the initial solution. In these cases, the solution equals the value found for the lower limit because this is the optimal solution. In other scenarios, there is a significant improvement in the value found through the heuristic when compared with the value of the initial solution.

Scenario	Zm	Z0	LB	$((Zm-LB)/LB)*100$	$((Z0-Zm)/Zm)*100$
1	12,790	12,790	12,790	0.00	0,00
2	37,910	49,995	28,040	35.20	31.88
3	79,027	164,908	65,819	20.07	108.67
4	50,739	50,739	50,739	0.00	0,00
5	62,160	66,910	57,640	7.84	7.64
6	55,345	74,071	46,495	19.03	33.84
7	61,550	73,585	58,025	6.07	19.55
8	70,445	80,925	61,950	13.71	14.88
9	248,494	318,428	231,411	7.38	28.14
10	2,053,181	2,273,021	1,946,458	5.48	10.71

**Table 4.** Obtained results

The results obtained for the real scenario (shown in table 4 and figure 6) show that there was a reduction of 10.71% when the best solution obtained was compared with the value of the initial solution.



**Figure 6.** Best solution for scenarios 10

It should be noted that the initial solution reflects the heuristics used by the planner in a real environment. Additionally, in relation to the lower bound value, there is a clear convergence between the value of the solution and the limit. In this case, the better solution lies just 5.48% from the value obtained for the lower bound and may be considered a satisfactory result.

## 8. Conclusion

Considering the type of industry studied, the results show the clear importance of a structured approach to improving the efficiency in problems of lot sizing and scheduling operations. In the real case study, the developed tool allowed for an improvement of approximately 10%.

It is possible, based on the developed work, that when this is extrapolated to other industrial environments, it may also bring significant benefits.

The generalisations made for the initial problem allow an innovative formulation compared with the known approaches to problems of lot sizing and scheduling of uniform parallel machines. Specifically, the following generalisations were useful: i) consider different setup times for the same sequence of tasks depending on the lot size and the task position in the queue of the machine; ii) the machines, in addition to being grouped together according to their processing speed, should be further sub grouped according to technical features, including the machine width.

From a more conceptual perspective, the work presented also provides significant advantages in terms of tactical planning. The study identified a planning philosophy in the textile industry based on their Material Requirement Planning (MRP) that generates production needs from the backlog.

It is expected, although difficult to quantify, that if the material requirements are generated from the nearly optimal sequence obtained with the developed program, additional gains beyond the direct gains demonstrated in the work will exist in the upstream production sections because it is a system-dependent demand.

### Author details

F. Charrua Santos

*Electromechanical Department of University of Beira Interior, Calçada Fonte do Lameiro, Covilhã, Portugal*

Francisco Brojo

*Department of Aeronautical Sciences of University of Beira Interior, Calçada Fonte do Lameiro, Covilhã, Portugal*

Pedro M. Vilarinho

*Department of Economics, Management and Industrial Engineering University of Aveiro, Campo Universitário de Santiago, Aveiro, Portugal*

## 9. References

- Allahverdi, Gupta and Aldowaisan (1999), "A review of scheduling research involving setup considerations", *International Journal of Management Science*, Vol. 27, pp 219-239
- Allahverdi, Ng, Cheng, Mikhail and Kovalyov (2008), A survey of scheduling problems with setup times or costs, *European Journal of Operational Research*, 187 pp. 985–1032.
- Anagnostopoulos and Rabadi (2002), A simulated annealing algorithm for the unrelated parallel machine scheduling problem, *Robotics, Automation and Control and Manufacturing: Trends, Principles and Applications*, pp15-120.
- Anghinolfi D and M. Paolucci (2007), Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach, *Computers and Operations Research*, 34 (11), pp. 3471-3490.
- Baker, K. R. (1974), *Introduction to Sequencing and Scheduling* New York: Wiley.
- Behnamian, J; Zandieh, M; Ghomi, SMTF 2010 A multi-phase covering Pareto-optimal front method to multi-objective parallel machine scheduling *International Journal Of Production Research* 48 pp: 4949-4976
- Blazewicz; Drozdowski; Formanowicz; Kubiak and Schmidt (2000), Scheduling preemptable tasks on parallel processors with limited availability, *Parallel Computing*, 26, pp. 1195-1211.
- Chaudhry, IA; Drake, PR 2009 Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms *International Journal Of Advanced Manufacturing Technology* pp 581-594
- Chen, CL; Chen, CL 2009 Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times *International Journal Of Advanced Manufacturing Technology* 43 pp161-169
- Chen, J.F. (2006), Minimization of maximum tardiness on unrelated parallel machines with process restrictions and setups, *The International Journal of Advanced Manufacturing Technology*, 29, pp. 557-563.
- Chen, JF 2009 Scheduling on unrelated parallel machines with sequence- and machine-dependent setup times and due-date constraints. *International Journal Of Advanced Manufacturing* 44 Issue: 11-12 pp1204-1212
- Chuang, MC; Liao, CJ; Chao, CW 2010 Parallel machine scheduling with preference of machines *International Journal Of Production Research* 48 pp 4139-4152
- Curry and Peters (2005), Rescheduling parallel machines with stepwise increasing tardiness and machine assignment stability objectives, *International Journal of Production Research*, 43, pp. 3231-3246.
- Edis, EB; Ozkarahan, I 2011 A combined integer constraint programming approach to a resource-constrained parallel machine scheduling problem with machine eligibility restrictions *Engineering Optimization* 43 pp: 135-157
- Eom, D.H., H.J. Shin, I.H. Kwun, J.K Shim, and S.S. Kim (2002), Scheduling jobs on parallel machines with sequence-dependent family set-up times, *International Journal of Advanced Manufacturing Technology*, 19, pp. 926–932.

- Eren, T; Guner, E 2009A bicriteria parallel machine scheduling with a learning effect  
International Journal Of Advanced Manufacturing Technology 40 pp 1202-1205
- Fanjul-Peyro, L; Ruiz, R 2010Iterated greedy local search methods for unrelated parallel machine scheduling European Journal Of Operational Research 207 pp: 55-69
- França, Gendreau, Laport, and Muller (1996), A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times, International Journal of Production Economics, 43, pp. 79-89.
- Gacias, B; Artigues, C Lopez, P 2010Parallel machine scheduling with precedence constraints and setup times Computers & Operations Research 37pp: 2141-2151
- Gupta, Hariri and Potts (1997), Scheduling a two-stage hybrid flow shop with parallel machines at the first stage, Annals of Operations Research, 69, pp. 171 – 191.
- Haouari, Gharbi and Jemmali (2006), Tight bounds for the identical parallel machine scheduling problem, International Transactions in Operational Research, 13, pp. 529-548.
- Hsu, C.J.; Kuo, WH; Yang, D.L., 2011. Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects. Applied Mathematical Modelling 35, 1492-1496
- Huang, SM; Cai, LN; Zhang, XY 2010 Parallel dedicated machine scheduling problem with sequence-dependent setups and a single server Computers & Industrial Engineering 58 pp165-174
- Huo, Y; Leung, JYT 2010 Parallel machine scheduling with nested processing set restrictions European Journal Of Operational Research 204 pp229-236
- Jaehwan and Marc, Scheduling Parallel Machines for the Customer Order Problem, Journal of Scheduling, Vol. 8, 2005, pp. 49-74
- Jiang, Yiwei; Tan, Zhiyi; He and Yong (2005), Preemptive Machine Covering on Parallel Machines, Journal of Combinatorial Optimization, 10, pp. 345-363.
- Jouglet, A; Savourey, D 2011 Dominance rules for the parallel machine total weighted tardiness scheduling problem with release dates Computers & Operations Research 38 pp1259-1266
- Kang, YH; Shin, HJ 2010 An adaptive scheduling algorithm for a parallel machine problem with rework processes International Journal of Production Research 48 pp95-115
- Kim, Na, Jang, and Chen (2006). Simulated annealing and genetic algorithm for unrelated parallel machine scheduling considering set-up times. International Journal of Computer Applications in Technology 26, 28-36.
- Kim, S.I., H.S. Choi and D.H. Lee, (2006) Scheduling algorithms for parallel machines with sequence-dependent set-up and distinct ready times: minimizing total tardiness, Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture, 221, pp. 1087-1096
- Lee, Wu and Chen (2006), A simulated annealing approach to makespan minimization on identical parallel machines, International Journal of Advanced Manufacturing Technology, 31, pp. 328-334.

- Liao and Lin, (2003), Makespan minimization for two uniform parallel machines, *International Journal of Production Economics*, 84, pp. 205-213.
- Lin, L; Lin, YX; Zhou, XW, 2010 parallel Machine Scheduling With A Simultaneity Constraint And Unit-Length Jobs To Minimize The Makespan *Asia-Pacific Journal Of Operational Research* 27 pp: 669-676
- Lin, YK; Pfund, ME; Fowler, JW 2011 Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems *Computers & Operations Research* 38 pp901-916
- Logendran, R., B. McDonell and B Smucker (2007), Scheduling unrelated parallel machines with sequence-dependent setups, *Computers & Operations Research*, 34, pp. 3420-3438.
- Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., 1983, Optimization by simulated annealing. *Science*, 220, 671± 680.
- Madhavan, K., (1993). Weighted earliness-tardiness minimization for jobs with sequence dependent set-up times using simulated annealing, Master's thesis, Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, USA
- Mellouli, R; Sadfi, C; Chu, CB, et al. 2009 Identical parallel-machine scheduling under availability constraints to minimize the sum of completion times *European Journal Of Operational Research* 197 pp 1150-1165
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E., 1953, Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21, 1087± 1092
- Meyr (2002), Simultaneous lotsizing and scheduling on parallel machines, *European Journal of Operational Research*, 139, pp. 277–292.
- Mokotoff (2001), Parallel Machine scheduling problems: A survey, *Ásia-Pacific Journal of Operational Research*, 18, pp. 193-242.
- Monma and Potts (1989), On the complexity of scheduling with batch setup times, *Operations Research*, 37, pp. 798-804
- Moradi, E; Zandieh, M 2010 Minimizing the makespan and the system unavailability in parallel machine scheduling problem: a similarity-based genetic algorithm *International Journal Of Advanced Manufacturing Technology* 51 pp 829-840
- Nessah, Yalaoui and Chu (2006), A branch and bound algorithm to minimize total weighted completion time on identical parallel machines with job release dates, *Proceedings of International Conference on Service Systems and Service Management*, 2, pp. 1192-1198.
- Oh and Kim (2008), A branch and bound algorithm for an identical parallel machine scheduling problem with a job splitting property, *Computers & Operations Research*, 35, pp. 863-875.
- Okolowski, D; Gawiejnowicz, S 2010 Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect *Computers & Industrial Engineering* 59 pp:272-279

- Picard, and Queyranne (1978), The time dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling, *Operations Research*, 26, pp. 86-110.
- Potts and Kovalyov (2000), Scheduling with batching: A review, *European Journal of Operational Research*, 120, pp. 228-249. Drexl and Kimms (1997), Lot sizing and scheduling - Survey and extensions Invited Review, *European Journal of Operational Research*, 99, pp. 221-235.
- Potts and Wassenhove (1992), Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity, *Journal of the Operational Research Society*, 43, pp. 395-406.
- Rabadi, Moraga and Al-Salem (2006), Heuristics for the Unrelated Parallel Machine Scheduling Problem with Setup Times, *Journal of Intelligent Manufacturing*, 17, pp. 85-97.
- Radhakrishnan and Ventura (2000), Simulated annealing for parallel machine scheduling with earliness tardiness penalties and sequence-dependent set-up times, *International Journal of Production Research*, 38, pp. 2233- 2252.
- Raja, K., V. Selladurai, R. Saravanan and C. Arumugam, (2008), Earliness-tardiness scheduling on uniform parallel machines using simulated annealing and fuzzy logic, *Proceeding of the Institution of Mechanical Engineers part B-Journal of Engineering Manufacture*, 222, pp. 333-346
- Rocha P. L., M. G. Ravetti and G. R. Mateus (2008), Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times, *Computers and Industrial Engineering*, 35, pp. 1250-1264
- Shakhlevich and Strusevich (2005), Pre-Emptive Scheduling Problems with Controllable Processing Times, *Journal of Scheduling*, 8, pp. 233-253.
- Shie-Gheun, Pyung, Jae and Woon (2004), Scheduling parallel batch processing machines with arbitrary job sizes and incompatible job families, *International Journal of production Research*, 42, pp. 4091-4107.
- Shim and Kim (2007), Minimizing total tardiness in an unrelated parallel-machine scheduling problem, *Journal of the Operational Research Society*, 58, pp. 346-354
- Shin and Leon (2004), Scheduling with product family set-up times: an application in TFT LCD manufacturing, *International Journal of Production Research*, 42, pp. 4235-4248.
- Sivrikaya-Serifoglu, F. and G. Ulusoy, "Parallel machine scheduling with earliness and tardiness penalties", *Computers and Operations Research*, Vol. 26,1999, pp. 773-787.
- Su, LH 2009 Scheduling on identical parallel machines to minimize total completion time with deadline and machine eligibility constraints *International Journal of Advanced Manufacturing Technology*, 40 pp 572-581
- Sumichrast and Baker (1987), Scheduling parallel processors: an integer linear programming based heuristic for minimizing setup time, *International Journal of Production Research*, 25, pp. 761- 771.

- Tang and Luo (2006), A new ILS algorithm for parallel machine scheduling problems, *Journal of Intelligent Manufacturing*, 17, pp. 609-619.
- Unlu, Y; Mason, SJ 2010 Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems *Computers & Industrial Engineering* 58 pp785-800
- Wang, XL; Cheng, TCE 2009 Heuristics for parallel-machine scheduling with job class setups and delivery to multiple customers *International Journal Of Production Economics* 119 pp199-206
- Wen-Chiunge, Chin-Chia and Chen (2006), A simulated annealing approach to makespan minimization on identical parallel machines, *The International Journal of Advanced Manufacturing Technology*, 31, pp. 328-334