Exact and heuristic approaches for lot splitting and scheduling on identical parallel machines

Carina Pimentel*

Centro de Investigação Algoritmi,
Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal
E-mail: carina@dps.uminho.pt
*Corresponding author

Filipe Pereira e Alvelos

Centro de Investigação Algoritmi,
Departamento de Produção e Sistemas,
Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal
E-mail: falvelos@dps.uminho.pt

António Duarte

Departamento de Gestão Industrial,
Instituto Politécnico de Bragança,
Campus de Santa Apolónia,
Ap. 1134 Bragança, Portugal
and
Centro de Investigação Algoritmi,
Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal
E-mail: aduarte@ipb.pt

José M. Valério de Carvalho

Centro de Investigação Algoritmi,
Departamento de Produção e Sistemas,
Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal
E-mail: vc@dps.uminho.pt

Abstract: In this paper, we address a lot splitting and scheduling problem existent in a textile factory. The factory we study produces a set of products that are made of, or assembled from, a list of components. During production, each component can be split into one or several lots of different sizes and each lot will be produced independently on one of a group of identical parallel machines.

We formulate the problem into a mixed integer programming model and develop a heuristic method to solve the model. The heuristic method is based on a network flow model with the objective to minimise the weighted sum of the total tardiness of products and the deviations occurred during production of each product. The deviation of a product is measured by the deviation of product completion time (the last component lot completion time) and completion time of the rest of components lots for the same product.

We present computational results and performance measures of the network flow heuristic for a set of randomly generated instances based on real world data.

Keywords: lot splitting and scheduling; mixed integer programming; MIP; network flows.

Reference to this paper should be made as follows: Pimentel, C., Alvelos, F.P., Duarte, A. and Valério de Carvalho, J.M. (2011) 'Exact and heuristic approaches for lot splitting and scheduling on identical parallel machines', *Int. J. Manufacturing Technology and Management*, Vol. 22, No. 1, pp.39–57.

Biographical notes: Carina Pimentel is a PhD student in Production and Systems Engineering at the University of Minho, Portugal. She graduated with a degree in Industrial Engineering (five-year course) since 2000 and she received her MSc in Industrial Engineering, in 2005. Both degrees were obtained at the University of Minho, Portugal. She is a member of the Systems Engineering, Optimization and Operations Research Group of the Algoritmi Research Center since 2001 and she has already been involved in several Portuguese funded projects. During her professional experience, she has worked in manufacturing industry, on teaching and on research. Her research interests include the development of exact and heuristic methods for production planning and scheduling problems.

Filipe Pereira e Alvelos is an Assistant Professor at the Department of Production and Systems Engineering, of the University of Minho, Portugal. He is also a Researcher of the Systems Engineering, Optimization and Operations Research Group of the Algoritmi Research Center. He received his PhD in 2005 in Operations Research from the University of Minho, his MSc and BS both in Electrical and Computers Engineering from the Faculty of Engineering of the University of Porto, Portugal, in 1996 and 1999, respectively. His current research interests are exact and heuristic methods for integer and combinatorial optimisation with applications to network design and routing, production planning and scheduling, and cutting and packing problems. He has authored several papers in refereed international journals and conferences and has been involved in several Portuguese and European Union funded projects.

António Duarte is a Professor at the School of Technology and Management of the Polytechnic Institute of Bragança, Portugal and a Researcher of the Systems Engineering, Optimization and Operations Research Group of the Algoritmi Research Center. He received his PhD in Operations Research, in 2006, from the University of Minho. His current research interests are the development of algorithms for production planning and scheduling using exact methods (mainly, branch-cut-price) and heuristics.

José M. Valério de Carvalho received his PhD in Engineering Production and Operations Research from the University of Minho, in 1991. He is a Full Professor of Operations Research in the Department of Production and Systems Engineering, School of Engineering, University of Minho, and also a Researcher of the Systems Engineering, Optimization and Operations Research (SEOOR) Group of the Algoritmi Research Center. His main research interests include large scale integer programming, and applications in cutting and packing problems, scheduling and network design.

1 Introduction

This paper studies a planning and scheduling problem existent in a knitting section of a textile industry that produces fine knitted goods. The objective is to develop a tool that simultaneously solves:

1 a lot splitting problem where components are split into lots of different sizes

2 an assignment and scheduling problem, in which the lots determined in (1) are assigned and scheduled on a set of identical parallel machines.

The factory we study produces a set of products made up of a group of components. The demand quantity and due date for the products are known in advance. Each component order can be split into several lots of different sizes, and each lot will be assigned and produced independently on one of a group of identical knitting machines. The objectives are:

1 to minimise total tardiness

2 to minimise the sum of deviations completion times of all the lots of components that belong to the same product from the product's completion time, which is assumed to be the completion time of the last component lot.

The first objective is related to on-time delivery of the products, while the second one is related to the reduction of the work-in-process inventory. The first objective is prioritised over the second one, since it has a major impact in the quality of service. Nevertheless, the second objective is very important to guarantee a smooth production flow. This second objective has not received an appropriate treatment in the literature. We are not aware of any work considering this topic.

We formulate the problem into a mixed integer programming (MIP) model and develop a heuristic method to solve the MIP model. The MIP is a continuous time model, with variable length time periods, in which only single component production runs may occur, i.e., only one lot may be produced. In Floudas and Lin (2004), continuous time representations are examined in detail.

The heuristic approach consists of solving two network flow problems in which the origins are associated with components to be produced and the destinations are associated with time intervals on each machine. The flow in an arc corresponds to the production time of the component (associated with the origin) spent in the time interval and on the machine (associated with the destination). Therefore, a solution of this model defines how much to produce of each component in each (pre-defined) time interval on each machine. In order to obtain a schedule from that solution, we apply a simple single pass procedure.

The problem of how to split and schedule a set of jobs on a set of parallel machines, motivated by time based objectives, has not yet been sufficiently studied in the literature. Yalaoui and Chu (2003) and Tahar et al. (2006) propose a heuristic algorithm to solve the identical parallel machine scheduling problem with sequence dependent setup times and job splitting to minimise makespan. They solve the problem in two phases. In Phase 1 the problem is reduced into a single machine scheduling problem with sequence dependent setup times. They transform this problem into a travelling salesman problem and solve it using Little's method. In Phase 2, Yalaoui and Chu (2003) try to improve the solution obtained in Phase 1 in a step by step procedure, taking into account setup times and job splitting, while Tahar et al. (2006) use a linear program to determine the size of the lots.

The main differences between our problem and the one studied by Yalaoui and Chu (2003) and Tahar et al. (2006) are that:

1. they consider the minimisation of the makespan, while we consider the minimisation of total tardiness plus the minimisation of the deviation between the completion time of a product and the completion times of all the lots of components that belong to that product
2. they consider sequence dependent setups, while we do not
3. they consider that all the machines can process all the jobs, while we restrict job assignments to specific machines.

Sheen and Liao (2007) present a network flow technique to solve a preemptive scheduling problem of identical parallel machines with availability constraints. Their goal is to minimise the maximum lateness. In their problem, each job can only be processed on specific machines. They solve this problem using a series of maximum flow problems. They propose a polynomial time two-phase binary search algorithm to verify the feasibility of the problem and to solve the scheduling problem optimally if a feasible schedule exists. This problem is related to ours, although we allow lot splitting while they allow preemption (a job cannot be processed at the same instant on different machines) and the objectives are different.

We adopt a continuous time representation in order to accurately determine the completion time deviations. We do not find in the literature any MIP model for the combined lot splitting and scheduling problem with the same representation. Rocha et al. (2008) develop two continuous time MIP models for a scheduling problem, although they do not consider lot splitting. Besides that, their problem has other characteristics, such as unrelated parallel machines and setup times that depend on both the job sequence and machine, which is different from ours.

Three general review papers related to scheduling are Allahverdi et al. (1999, 2008) and Zhu and Wilhelm (2006). All of them consider research involving setups and present the literature categorised by shop environment/machine configuration. In Zhu and Wilhelm (2006) both optimisation and heuristic solution methods are reviewed. Zhu and Wilhelm (2006) argue that there are fertile opportunities available for research addressing due date related objectives. We also agree and, as far as we know, there are no papers addressing the minimisation of deviations between the completion times of all the lots belonging to the same product, which we study here. Allahverdi et al. (1999) and Allahverdi et al. (2008) also argue that due date related objectives need to be emphasised, especially on parallel machines and multi-stage scheduling configurations.

Xing and Zhang (2000) prove that the identical parallel machine scheduling problem with splitting jobs, without setup times and with the objective of minimising total tardiness is NP-hard. As our problem is an extension of the previous one, it is also NP-hard.

The main contributions of this paper are as follows:

1. The development of a MIP model for a lot splitting and scheduling problem that, besides splitting demands into lots of smaller size and sequencing those lots on a set of parallel machines, determines the beginning and finishing instant times of each lot

on each machine. This allows to model the objective related to the minimisation of deviations between the completion times of related lots.

2 The development of a network flow heuristic that produces solutions for this NP-hard problem very quickly, making it an effective tool for practical use.

3 The development of algorithms that provide automated scheduling plans for a knitting problem of a textile industry.

The remainder of this paper is structured as follows: in Section 2 we define the lot splitting and scheduling problem and in Section 3 we present a MIP model for it. Section 4 is dedicated to the network flow heuristic and Section 5 to the computational experiments. Finally, in Section 6, the main conclusions of this work are summarised.

2 Problem definition

This research is motivated by the need of a textile company to improve their current knitting scheduling plans and to increase its efficiency. The knitting section produces the garment parts of a product, such as sleeves, back bodies, front bodies and scarves.

The production system of the knitting section consists of three groups of identical parallel machines. The main characteristic that distinguishes these three sets of machines is a gauge (21, 24 or 27) which is also a characteristic of each product. The planning and scheduling of the section is done by gauge. Therefore, three independent production plans must be prepared.

Plans are made weekly. Each production order has information about the products (defined by a piece of cloth and size) requested by customers, including their due dates, quantities and the corresponding set of components. All the components belong to the same level of the product structure. The customer demands by component will be split into smaller lots of variable size and each lot will be assigned to one machine. The completion time of a product is the completion time of the last component produced. The setup times involved in a change between garment parts are neglected.

Due to machines and components technical characteristics, a compatibility matrix between machines and components must be considered. Even within the same gauge type, a given machine may be appropriate to produce a given component, while another one may not. Besides, each machine has a given release date. The machines are identical, since they take the same amount of time to produce a unit of a given component.

The first objective is to minimise total tardiness. The second objective is related to the minimisation of the time deviation between the completion time of the product (the moment when all its components are finished) and the completion time of all the components lots.

The production plan must state how to split the components demand into lots of variable size, on which machines those lots are produced, and when they are started and finished. As the lot splitting decisions are taken at the same level and in coordination with the scheduling decisions, the quality of the solutions is increased. It is worth noting that, in an optimal solution, two or more lots of the same component may be assigned to the same machine (with lots of other components in-between them). Another important feature of the problem is that, in an optimal solution, empty intervals may exist between two successive lots.

3 MIP model

In this section, we present a MIP model developed for the lot splitting and scheduling problem presented in Section 2. This MIP is based on a discretisation of the time horizon into several macro-periods (days), each one with a pre-defined number of micro-periods (production runs). In each micro-period a single component is produced or there is no production at all. The lengths of the macro-periods and the micro-periods can vary. We consider arbitrary capacities per macro-period in order to introduce more flexibility into the model, for example, to allow a part of the day for machine maintenance. If the machines are always available, then the set of macro-periods can be removed and only a set of micro-periods is needed.

The main decision variables are related to:

1. the quantities to produce of each component, in each run, in each day and on each machine
2. the completion time of each component, in each run, in each day and on each machine
3. the production completion time of each product
4. the tardiness of each product
5. a binary variable that takes the value one if there is a production of a given component, in a given run, in a given day, and on a given machine, or zero otherwise.

There are several sets of constraints to consider. The first set of constraints is related to the product demand. The second set determines the completion times of each lot of components. The third set determines the completion time of each product, and the fourth set determines the tardiness of each product. We also consider a set of capacity constraints guaranteeing that the completion times of each component, in each run of a day, on each machine respect the daily capacity. Another set of constraints determines the values of the variables that are associated with the calculations of the completion time deviations and, the last set, guarantees that at most one component is produced in each production run.

Next we formally present the proposed MIP model which we designate by lot splitting and scheduling model (LSSM). Consider the following sets, parameters and decision variables:

Sets:

N	set of products
J	set of components
M	set of machines
K	set of runs
I	set of days
$S(n)$	set of components that belong to product n.

Parameters:

D_n demand of product *n*

r_m ready time of machine *m*

a_j production unit time of component *j*

f_{jn} number of units of component *j* required to produce one unit of product $n(j \in S(n))$

β_n tardiness weight of product *n* per unit time

d_n due date of product *n*

h_{mi} clock time when day *i* finishes on machine *m*

θ_j weight of a production run of component *j*

α_n per unit time weight resulting from the deviation between the production completion time of product *n* and each of its components

b_{jm} is equal to one if component *j* can be processed on machine *m* and is equal to zero otherwise

$k_{\max}$ is the pre-defined number of runs of a day

M a big number.

Decision variables:

X_{jmki} quantity to produce of component *j* during the *k*th production run of day *i* on machine *m*

T_{jmki} completion time of component *j* in the *k*th production run of day *i* on machine *m*

C_n production completion time of product *n*

L_n tardiness of product *n*

Z_{jmki} is equal to one if component *j* is produced in the *k*th production run of day *i* on machine *m* and is equal to zero otherwise

U_{jmki} is equal to T_{jmki} if component *j* is produced in the *k*th production run of day *i* on machine $m(Z_{jmki} = 1)$ and is equal to C_n if component *j* is not produced in the *k*th production run of day *i* on machine $m(Z_{jmki} = 0)$.

LSSM model is:

$$Min \sum_{n\in N} \sum_{j\in J|j\in S(n)} \sum_{m\in M|b_{jm}=1} \sum_{k\in K} \sum_{i\in I} \alpha_n \left(C_n - U_{jmki} \right) +$$

$$\sum_{j\in J} \sum_{m\in M|b_{jm}=1} \sum_{k\in K} \sum_{i\in I} \theta_j Z_{jmki} + \sum_{n\in N} \beta_n L_n$$

subject to:

$$\sum_{m\in M|b_{jm}=1} \sum_{k\in K} \sum_{i\in I} X_{jmki} = f_{jn} D_n, \forall n \in N, \forall j \in J \,|\, j \in S(n) \qquad (1)$$

$$T_{jmki} \geq r_m + a_j X_{jmki} - M(1 - Z_{jmki}), \forall j \in J \middle| b_{jm} = 1, \forall m \in M \tag{2}$$

$$\begin{aligned} T_{jmki} \geq T_{1m,k-1,i} + a_j X_{jmki} - M(2 - Z_{jmki} - Z_{1m,k-1,i}),& \\ \forall j, l \in J \,\middle|\, b_{jm} = 1 \text{ and } b_{lm} = 1, \forall m \in M, k = 2, \ldots, K_{max}, \forall i \in I& \end{aligned} \tag{3}$$

$$\begin{aligned} T_{jm1i} \geq T_{1m,k_{max},i-1} + a_j X_{jm1i} - M(2 - Z_{jm1i} - Z_{1m,k_{max},i-1}),& \\ \forall j, l \in J \,\middle|\, b_{jm} = 1 \text{ and } b_{lm} = 1, \forall m \in M, i \geq 2& \end{aligned} \tag{4}$$

$$\begin{aligned} C_n \geq T_{jmki} - M(1 - Z_{jmki}),& \\ \forall n \in N, \forall j \in J \,\middle|\, b_{jm} = 1 \text{ and } j \in S(n), \forall m \in M, \forall k \in K, \forall i \in I& \end{aligned} \tag{5}$$

$$T_{jmki} \leq h_{mi}, \forall j \in J \,\middle|\, b_{jm} = 1, \forall m \in M, \forall k \in K, \forall i \in I \tag{6}$$

$$T_{jm1i} - a_j X_{jm1i} \geq h_{m,i-1}\left(1 - (1 - Z_{jm1i})\right), \forall j \in J \,\middle|\, b_{jm} = 1, \forall m \in M, i \geq 2 \tag{7}$$

$$L_n \geq C_n - d_n, \forall n \in N \tag{8}$$

$$U_{jmki} - T_{jmki} \leq M(1 - Z_{jmki}), \forall j \in J \,\middle|\, b_{jm} = 1, \forall m \in M, \forall k \in K, \forall i \in I \tag{9}$$

$$\begin{aligned} U_{jmki} - C_n \leq M Z_{jmki}, \forall n \in N, \forall j \in J \,\middle|\, b_{jm} = 1 \text{ and } j \in S(n),& \\ \forall m \in M, \forall k \in K, \forall i \in I& \end{aligned} \tag{10}$$

$$\sum_{j \in J | b_{jm} = 1} Z_{jmki} \leq 1, \forall m \in M, \forall k \in K, \forall i \in I \tag{11}$$

$$X_{jmki} \leq M Z_{jmki}, \forall j \in J \,\middle|\, b_{jm} = 1, \forall m \in M, \forall k \in K, \forall i \in I \tag{12}$$

$$\sum_{j \in J | b_{jm} = 1} Z_{jmki} \geq \sum_{j \in J | b_{jm} = 1} Z_{jm,k+1,i}, \forall m \in M, \forall i \in I, k \leq K_{max} - 1 \tag{13}$$

$$X_{jmki}, T_{jmki}, C_n, L_n \text{ and } U_{jmki} \geq 0 \;\; \forall n \in N, \forall j \in J, \forall m \in M, \forall k \in K, \forall i \in I \tag{14}$$

$$Z_{jmki} \in \{0,1\}, \forall j \in J, \forall m \in M, \forall k \in K, \forall i \in I \tag{15}$$

In the objective function, the deviations between the completion time of a product and the completion time of all the components that belong to this product are minimised, as well as the tardiness. The second term of the objective function is a penalty for running a production, which can be considered as a dummy cost. This cost is only considered because of the set of constraints (5). Constraints (1) are the demand constraints. These constraints associate the demand of a product with the demand of its components. The quantities to produce of each component (X_{jmki}) can be sized in lots of smaller quantities and can be produced on several machines at the same time or at different times. Constraints (2), (3) and (4) give the completion time of each component, on each machine, run and day. In constraints (5), the completion time of a given product is determined. Constraints (6) and (7) are the capacity constraints. They associate the completion time of each component, on each machine, run and day with the finish time of a day. Constraints (8) relate the due date of a product with its completion time and tardiness. The sets (9) and (10) are related to the minimisation of the deviations between the completion time of a product and the completion time of all the lots of components that belong to this product. In these constraints, the values of the U_{jmki} variables, that are present in the objective function, are determined. In constraints (11) it is stated that at most one production per machine, run and day can occur. Constraints (12) establish the relation between the quantity to produce of a given component on a given machine, run and day (X_{jmki}) and the Z_{jmki} variables, forcing the X_{jmki} to be equal to zero if the Z_{jmki} variables are equal to zero. Constraints (13) are included in the model in order to correctly determine the completion times of each component, on each machine, run and day. Constraints (14) are the non-negativity constraints and constraints (15) force the Z_{jmki} decision variables to take binary values.

Solving real-world instances directly with model *LSSM* is not computationally efficient. Besides the large number of binary variables and constraints, the presence of a large number of big-M constraints suggests that the linear relaxation of the model provides poor quality lower bounds. Furthermore, the model has symmetry (usual on parallel machine problems) which makes branching (in a branch-and-bound based approach) less effective. LSSM was a first effort to model time deviations in scheduling problems, using integer programming. For addressing real-world instances, we propose the heuristic approach described in the next section.

4 Network flow heuristic

The MIP model formulated in Section 3 can be used to solve the lot splitting and scheduling problem defined in Section 2. However, the problem is NP-hard and the MIP model takes a large amount of time to solve real world instances. Since solution times are an important concern for practical problems, we adopted a heuristic approach. This approach consists of solving the problem in two phases using a network flow model and a scheduling heuristic.

4.1 Phase I

In this subsection we present the network model and a scheduling heuristic that converts the network model solution (a set of flows) into a valid schedule.

4.1.1 Network flow model

In this first phase, we only consider one component of each product: the one with the longest processing time. The network is defined in a bipartite graph. The components are associated with origin nodes, with supplies defined by the processing time of the corresponding component.

The destinations nodes are associated with time intervals and machines. The information used to define the time intervals consists of:

1 the machine release times

2 the due dates of the components considered in the origins

3 the planning horizon.

The definition of the intervals is illustrated in the example described below.

Example: Consider a problem with five products that must be scheduled, at most, on five machines, in the next 48 hours. The data associated with this example is presented in Table 1 and in Table 2.

Table 1 Example data

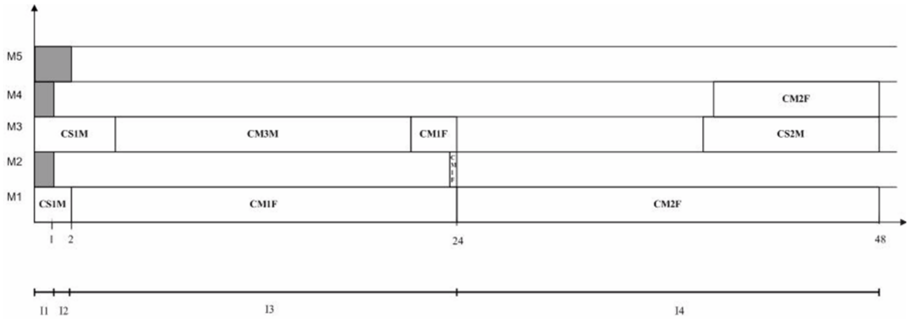
Product	*Due date (hours)*	*Component*	*Compatible machines*	*Total processing time (hours)*
CM1	24	CM1F	1, 2, 3, 4, 5	25
		CM1C	1, 2, 3, 4, 5	25
CM2	48	CM2F	1, 4, 5	33.33
		CM2C	1, 4, 5	33.33
CM3	24	CM3F	1, 2, 3, 4	8.33
		CM3C	1, 2, 3, 4	8.33
		CM3M	1, 2, 3, 4, 5	16.67
CS1	24	CS1F	1, 3, 4	3.33
		CS1C	1, 3, 4	3.33
		CS1M	1, 2, 3, 4, 5	6.66
CS2	48	CS2F	1, 2, 3, 4, 5	5
		CS2C	1, 2, 3, 4, 5	5
		CS2M	1, 2, 3, 4, 5	10

Table 2 Release times of machines

Machine	*Release time (hours)*
1	0
2	1
3	0
4	1
5	2

The components defining the origins are: CM1F, CM2F, CM3M, CS1M and CS2M (the ones with longest processing times). In Figure 1, where the solution is represented, the time intervals are defined, based on the release times of the machines presented in Table 2, the due dates of the components considered in the origins and the planning horizon. The total number of destinations of example is 16 (4 associated with machine 1, 3 with machine 2, 4 with machine 3, 3 with machines 4 and 2 with machine 5). Machine 1 destinations will be: I1, I2, I3 and I4. The demand of each destination is equal to the duration of the time interval. For example, the demand associated with time interval I3-M1 is 22.

Figure 1 Time intervals and Gantt chart for the solution of Phase I of example



Each time interval must have a machine associated because of the compatibility matrix between the components and the machines. If all the components could be processed on all the available machines, this association would not be needed and the demand of each time interval would be the product of the time interval duration by the number of machines available in that time interval.

Between each origin and each destination, there is an arc if the component associated with the origin can be produced on the machine associated with the destination. The flow in each arc represents the processing amount of the corresponding component in the corresponding interval.

The cost of an arc depends on three factors: tardiness, earliness and machines prioritisation. We associate weights with the tardiness and earliness penalties, being the tardiness weight much larger. The tardiness penalty only appears in arcs associated with time intervals beyond the associated component due date, and its value is the product of the tardiness weight by the difference between the end time of the interval and the due date of the given component. The earliness penalty is associated with arcs representing time intervals that end earlier than the due date of the corresponding component. Its value is the product of the earliness weight by the difference between the due date of the component and the end time of the considered time interval. The machines are prioritised with the aim of minimising the deviations between the completion time of a product and the completion time of all the lots of related components. That is accomplished by associating an order number to each arc that is multiplied by a prioritisation penalty. All the arcs of a given machine will have the same order number. We use the data of the example to illustrate how the order numbers are determined. We order the machines according to their release times and we use that order during the assignment of the machines to the time intervals. Considering the release dates of Table 2 one possible order of machines is: M1 – M3 – M2 – M4 – M5. This order will be used in the decomposition of the intervals by machines. For example time interval 3 (see Figure 1) will be decomposed into 5 intervals. The first one will be I3-M1, the second one I3-M3, the third I3-M2, the fourth I3-M4, and finally the fifth I3-M5. All the arcs associated with machine 1 (for example I3-M1) will have order number 1; all the arcs associated with machine 3 will have order number 2; all the arcs associated with machine 2 will have order number 3; and so on.

We solve the problem using all the alternative orders of machines and we select the best solution, i.e., the one with a lower value of total tardiness plus total deviation between a product completion and the completion times of all the lots of components associated with that product.

We next present the network flow model. Consider the sets, parameters and decision variables defined below.

Sets:

N_1 set of selected components (one for each product)

N_2 set of interval/machine pairs

A set of arcs

M_j set of components that can be produced on machine j.

Parameters:

p_i total production time of component i

b_j duration of time interval j

c_{ij} cost of arc (i, j) based on the tardiness, earliness and machine priority

d_i due date of component i

t_j finish time of interval j

α weight of machine priority

β weight of tardiness per unit time

μ weight of earliness per unit time

o_{ij} order number of arc (i, j).

4.1.2 Decision variables

x_{ij} – flow of arc (i, j), i.e., time spent producing component associated with node i in the interval and on the machine associated with node j.

The network flow model is:

$$Min\ Z = \sum_{(i,j)\in A: i\in M_j} c_{ij}x_{ij}$$

subject to:

$$\sum_{j:(i,j)\in A \text{ and } i\in M_j} x_{ij} = p_i,\quad \forall i \in N_1 \tag{16}$$

$$\sum_{i:(i,j)\in A\ and\ i\in M_j} x_{ij} \le b_j, \forall j \in N_2 \tag{17}$$

$$x_{ij} \ge 0, \forall (i, j) \in A \tag{18}$$

The value of c_{ij} is determined using equation (19).

$$\sum_{(i,\,j)\in A:d_i<t_j} \beta\left(t_j-d_i\right)+\sum_{(i,\,j)\in A:d_i>t_j} \mu\left(d_i-t_j\right)+\sum_{(i,\,j)\in A:d_i\geq t_j \text{ and } i\in M_j} \alpha o_{ij} \tag{19}$$

In the objective function we minimise the sum of tardiness, earliness and machine priority penalties. Constraints (16) are the supply constraints that force the sum of the flows out of each component to be equal to its total production time. Constraints (17) are the demand constraints. In these constraints we force the flow into each interval/machine pair not to exceed the interval duration. Constraints (18) are non-negativity constraints. The reader is referred to Ahuja et al. (1993) for additional details about network flow models.

In Table 3, we present one of the optimal solutions of the network flow model of Phase I for the above presented example, for machine order M1 – M3 – M2 – M4 – M5. In this solution, component CM3M is produced using 16.67 hours of time interval 3 on machine 3. The same type of analysis can be done for the remaining components.

Table 3 Optimal solution of the network flow model of Phase I of example

Component/size	*Interval*	*Machine*	*Flow*
CM1F	I3	M1	22
CM1F	I3	M2	0.33
CM1F	I3	M3	2.67
CM2F	I4	M1	24
CM2F	I4	M4	9.33
CM3M	I3	M3	16.67
CS1M	I1	M1	1
CS1M	I2	M1	1
CS1M	I1	M3	1
CS1M	I2	M3	1
CS1M	I3	M3	2.67
CS2M	I4	M3	10

4.1.3 Scheduling procedure

After solving the network flow model, a single-pass procedure is applied to compute a schedule from the obtained flow solution. Note that the network flow solution defines the time spent by each component in each time interval and on each machine, but does not provide the exact moment when the production should start. We apply the following three steps in the scheduling procedure:

Step 1 Schedule all the fully occupied intervals with only one component.

Step 2 For all the intervals with only one component, but not fully occupied.

Step 2.1 Schedule the component next to an already scheduled part of it, if the component is also produced in the first position of the next interval or in the last position of the previous interval and go to Step 3.

Step 2.2 Schedule the component from the end time of the interval to its start time, if the finish time of the interval is smaller than or equal to the due date of the component. If the finish time of the interval is greater than the due date of the component, schedule it from the beginning of the interval.

Step 3 For all the intervals with more than one component.

Step 3.1 Select the component(s) that is(are) also scheduled in the first position of the next interval or in the last position of the previous interval, and schedule it (them) next to the already scheduled part(s) of it(them). If a given component is simultaneously scheduled in the first position of the next interval and in the last position of the previous interval, schedule it at the beginning of the current interval.

Step 3.2 Schedule all the remaining components, according to the minimum slack first rule (described ahead), starting in the beginning of the interval or in the first free instant inside the interval. If more than one component has the same slack, select arbitrarily one of them.

The minimum slack of a given component, *i*, in a given interval/machine, *j*, is given by: $d_i - x_{ij} - t$, where *t* is the beginning of the free subinterval inside interval *j*. Note that t can correspond to the beginning instant time of the interval *j* or to a certain instant time inside interval *j*. This second case can exist if in interval *j* there is one or more components already scheduled at the beginning of that interval. In Figure 1 we present the Gantt chart resulted from the application of the scheduling heuristic to the solution presented in Table 3.

4.2 Phase II

In Phase II, we define a new network flow model based on the solution obtained in the first phase. In this model the origins are associated with the components that were not considered during Phase I and the supply of each origin is given by the production time of the corresponding component. As in Phase I, the destinations are also associated with time intervals and machines.

The release times of the machines and the planning horizon are again used in the intervals definition. The due dates of the components are now given by the Phase I completion time of the component belonging to the same product. Consider, for example, product CM3 of the presented example. The due date of this product is 24 as can be seen in Table 1, but as CM3M finishes at 21.33, as can be seen in Figure 2, the due date of CM3F and of CM3C (that are the other components of CM3) will be 21.33. Besides considering this data to define the time intervals, in Phase II, the machine loads of Phase I also need to be considered. The time intervals of Phase II for the example, which we present in Figure 2, are: I1-M2, I1-M4, I2-M2, I2-M4, I2-M5, I3-M2, I3-M4, I3-M5, I4-M2, I4-M4, I4-M5, I5-M4, I5-M5, I6-M4, I6-M5, I6-M2, I6-M3, I7-M4, I7-M5, I7-M2, I8-M5 and I8-M2. The demand associated with each time interval/machine is given by the duration of the time interval.

Figure 2 Time intervals and Gantt chart for the solution of Phase II of example

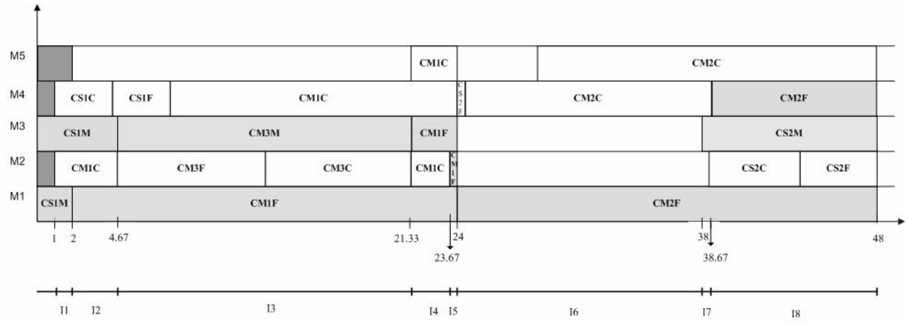


Table 4 Optimal solution of the network flow model of Phase II of example

Component/size	*Interval*	*Machine*	*Flow*
CM1C	I1	M2	1
CM1C	I2	M2	2.67
CM1C	I4	M2	2.33
CM1C	I3	M4	13.67
CM1C	I4	M4	2.33
CM1C	I5	M4	0.33
CM1C	I4	M5	2.34
CM1C	I5	M5	0.33
CM2C	I6	M4	13.33
CM2C	I7	M4	0.67
CM2C	I6	M5	9.33
CM2C	I7	M5	0.67
CM2C	I8	M5	9.33
CM3F	I3	M2	8.33
CM3C	I3	M2	8.33
CS1F	I2	M4	0.33
CS1F	I3	M4	3
CS1C	I1	M4	1
CS1C	I2	M4	2.33
CS2F	I8	M2	4.33
CS2F	I6	M4	0.67
CS2C	I8	M2	5

The arcs of the network of Phase II are defined in the same manner as in Phase I and the arc penalties are also calculated in the same way. In Phase II the order number of each arc is formed dynamically, using the machine release dates and the machine loads of the solution of Phase I. The order numbers of the arcs that link component CM1C to the time intervals above presented, is respectively:1, 2, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 1, 2, 3, 4, 1, 2, 3, 1, 2. We do not present in this subsection the network model analytically as it is similar to the one presented in Phase I. The scheduling heuristic used in Phase II is equal to the one used in Phase I. In Table 4 and in Figure 2 we present the flow solution and the Gantt chart of Phase II, respectively.

5 Computational experiments

In this section we present some computational results for the network flow heuristic. We address instances similar to those occurring in the factory: the number of machines of each gauge is the same as in the company, and the demands of different product types and the processing times were randomly generated but within intervals defined by data obtained at the company. There are other parameters of the real problem that are generated randomly taking into account the data provided by the company.

We tested 18 instances grouped by gauge (six instances of gauge 21, six of gauge 24 and six of gauge 27). In Table 5 the instances characteristics are depicted, being the first six associated with gauge 21, the next six with gauge 27 and the last six with gauge 24.

Table 5 Instances characteristics

Instance	*Number of products*	*Number of components*	*Number of machines*
Inst20T1.4.G21	8	18	5
Inst30T1.5.G21	20	60	5
Inst40T1.3.G21	20	58	5
Inst50T1.1.G21	29	74	5
Inst60T1.2.G21	26	70	5
Inst70T1.3.G21	30	90	5
Inst20T1.4.G27	31	89	11
Inst30T1.5.G27	44	128	11
Inst40T1.3.G27	53	142	11
Inst50T1.1.G27	43	120	11
Inst60T1.2.G27	71	197	11
Inst70T1.3.G27	70	182	11
Inst20T1.4.G24	34	94	13
Inst30T1.5.G24	38	90	13
Inst40T1.3.G24	57	152	13
Inst50T1.1.G24	55	152	13
Inst60T1.2.G24	81	216	13
Inst70T1.3.G24	82	226	13

Table 6 Solution outputs

Instance	*Solution time best machine order (seconds)*	*Solution time all possible machine orders (seconds)*	*Total tardiness (hours)*	*Number products late*	*Total deviation (hours)*	*Average machine utilisation (%)*	*Number of lots*	*Average number of lots/component*
Inst20T1.4.G21	0.000	0.187	5.6	1	6.3	45.6	19	1.06
Inst30T1.5.G21	0.047	0.047	148.5	10	796.8	92.7	101	1.68
Inst40T1.3.G21	0.032	0.032	0.0	0	2779.9	99.4	95	1.64
Inst50T1.1.G21	0.047	0.047	48.0	1	3049.8	87.5	116	1.57
Inst60T1.2.G21	0.031	0.031	0.0	0	2989.2	97.9	113	1.61
Inst70T1.3.G21	0.046	0.171	411.6	14	1698.3	59.3	130	1.44
Inst20T1.4.G27	0.156	2.293	0.0	0	4609.1	87.9	253	2.84
Inst30T1.5.G27	0.218	27.690	0.7	2	6841.4	95.5	348	2.72
Inst40T1.3.G27	0.655	73.351	0.0	0	8168.4	95.6	370	2.61
Inst50T1.1.G27	0.157	13.307	0.0	0	5594.3	99.5	260	2.17
Inst60T1.2.G27	0.452	814.398	0.0	0	4389.2	99.7	481	2.44
Inst70T1.3.G27	0.358	9.531	222.3	12	5370.2	97.5	393	2.16
Inst20T1.4.G24	0.202	221.598	0.9	2	5501.3	95.5	269	2.86
Inst30T1.5.G24	0.187	11.591	137.3	9	4314.7	97.5	313	3.48
Inst40T1.3.G24	0.592	190.164	26.4	5	5248.9	91.1	457	3.01
Inst50T1.1.G24	0.577	59.249	1.5	6	3396.8	65.3	424	2.79
Inst60T1.2.G24	0.952	1510.850	0.0	0	10081.7	90.2	603	2.79
Inst70T1.3.G24	0.764	263.842	1.0	4	13740.5	88.4	661	2.92

We wrote a computer program for the network flow heuristic in visual C++, and solved the network flow model with Cplex 11.0 (ILOG, 2007). The code produces a week production plan represented in a Gantt chart, with the main performance measures and information related to the machines (the number of machines used for each product, the number of machines used for each component and the number of components produced on each machine).

In Table 6 we present the main performance measures. All the instances tested were solved within half an hour and most of them were solved in few seconds. In column 2 the solution time to solve the problem with the best machine order is presented, while in column 3 the total solution time is presented, including the time to test all the possible machine orders. The values of total tardiness and of total deviation are presented in columns 4 and 6. The machine utilisation of a given machine M_i is given by: $\frac{\textit{Total occupied time of } M_i}{\textit{Time Horizon - Release date } M_i} \times 100$. In column 7 the average machine utilisation of each instance is presented. Finally, in column 8 and 9 we present the number of lots of components obtained for each instance and the average number of lots per component.

In some instances, the total tardiness is large. This may occur, because the due dates are generated randomly, and there may be a huge order with a due date that cannot be fulfilled even if all the resources were assigned to it. The total deviation seems to increase with the size of the instances (measured in terms of the number of components and of the number of machines), especially when the average machine utilisation is high. Those deviations are usually high for the set of instances tested, denoting a potential field of improvement. There seems to be a positive correlation between the number of lots of a solution and the total deviation of that solution, i.e., the instances with a larger number of lots have a larger total deviation (see for example instances Inst60T1.2.G24 and Inst70T1.3.G24). However this relationship does not hold for all the instances tested (see for example instances Inst60T1.2.G27 and instance Inst50T1.1.G24). Finally, as the instance size increases the average number of lots per component also increases.

6 Conclusions

In this paper we presented a MIP model for an integrated lot splitting and scheduling problem existent in the knitting section of a textile factory. The problem has the following characteristics: sets of identical parallel machines, demands and due dates associated with end products, unit production times associated with components, a compatibility matrix between machines and components, release dates of machines and a weekly planning horizon. Besides the minimisation of the total tardiness, we addressed the objective of minimising the deviations of the completion times of the lots of components of the same product. This latter objective is not usually treated, but it is clearly relevant in the practical application described.

For obtaining solutions in a reasonable time, we also developed a heuristic approach based on a network flow model. In the context of the practical problem, the developed methods and computational tools are of major interest since they potentially allow reducing the amount of work-in-process. The computational tests show that feasible solutions for high machine utilisations can be obtained quickly.

Acknowledgements

This work was supported by the Portuguese Science and Technology Foundation through the doctoral Grant SFRH/BD/38582/2007 for Carina Pimentel. The authors thank the anonymous referees for their constructive comments, which led to a clearer presentation of the material.

References

Ahuja, R., Magnanti, T. and Orlin, J. (1993) *Network Flows: Theory, Algorithms and Applications*, New Jersey, Prentice Hall.

Allahverdi, A., Gupta, J. and Aldowaisan, T. (1999) 'A review of scheduling research involving setup considerations', *Omega The International Journal of Management Science*, Vol. 27, pp.219–239.

Allahverdi, A., Ng, C., Cheng, T. and Kovalyov, M. (2008) 'A survey of scheduling problems with setup times or costs', *European Journal of Operational Research*, Vol. 187, pp.985–1032.

Floudas, C. and Lin, X. (2004) 'Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review', *Computers and Chemical Engineering*, Vol. 28, pp.2109–2129.

ILOG (2007) *CPLEX 11.0 User's Manual*.

Rocha, P., Ravetti, M., Mateus, G. and Pardalos, P. (2008) 'Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times', *Computers and Operations Research*, Vol. 35, No. 4, pp.1250–1264.

Sheen, G. and Liao, L. (2007) 'Scheduling machine-dependent jobs to minimize lateness on machines with identical speed under availability constraints', *Computers and Operations Research*, Vol. 34, pp.2266–2278.

Tahar, D., Yalaoui, F., Chu, C. and Amodeo, L. (2006) 'A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times', *International Journal of Production Economics*, Vol. 99, pp.63–73.

Xing, W. and Zhang, J. (2000) 'Parallel machine scheduling with splitting jobs', *Discrete Applied Mathematics*, Vol. 103, pp.259–269.

Yalaoui, F. and Chu, C. (2003) 'An efficient heuristic approach for parallel machine scheduling with job splitting and sequencedependent setup times', *IIE Transactions*, Vol. 35, pp.183–190.

Zhu, X. and Wilhelm, W. (2006) 'Scheduling and lot sizing with sequence-dependent setup: a literature review', *IIE Transactions*, Vol. 38, No. 11, pp.987–1007.